ADVCOMP 2014

The Eighth International Conference on Advanced Engineering Computing and Applications in Sciences

ISBN: 978-1-61208-354-4

August 24 - 28, 2014

Rome, Italy

ADVCOMP 2014 Editors

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain

Sigeru Omatu, Osaka Institute of Technology, Japan

ADVCOMP 2014

Forward

The Eighth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2014) held on August 24 - 28, 2014 - Rome, Italy, was a multi-track event covering a large spectrum of topics related to advanced engineering computing and applications in sciences.

With the advent of high performance computing environments, virtualization, distributed and parallel computing, as well as the increasing memory, storage and computational power, processing particularly complex scientific applications and voluminous data is more affordable. With the current computing software, hardware and distributed platforms effective use of advanced computing techniques is more achievable.

The goal of ADVCOMP 2014 was to bring together researchers from the academia and practitioners from the industry in order to address fundamentals of advanced scientific computing and specific mechanisms and algorithms for particular sciences. The conference provided a forum where researchers were able to present recent research results and new research problems and directions related to them. The conference sought contributions presenting novel research in all aspects of new scientific methods for computing and hybrid methods for computing optimization, as well as advanced algorithms and computational procedures, software and hardware solutions dealing with specific domains of science.

We take here the opportunity to warmly thank all the members of the ADVCOMP 2014 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to ADVCOMP 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ADVCOMP 2014 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the ADVCOMP 2014 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in advanced scientific computing.

We hope Rome provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ADVCOMP 2014 Chairs:

ADVCOMP Advisory Chairs
Chih-Cheng Hung, Southern Polytechnic State University, USA
Juha Röning, Oulu University, Finland
Sigeru Omatu, Osaka Institute of Technology, Japan
Erich Schweighofer, University of Vienna, Austria
Paul Humphreys, University of Ulster, UK
Danny Krizanc, Wesleyan University, USA
Ivan Rodero, Rutgers University - Piscataway, USA
Ali Shawkat, CQ University of Australia - North Rockhampton, Australia
George Spanoudakis, City University London, UK
Vladimir Vlassov, KTH Royal Institute of Technology, Sweden
Jerry Trahan, Louisiana State University, USA
Dean Vucinic, Vrije Universiteit Brussel (VUB), Belgium
Rudolf Berrendorf, Bonn-Rhein-Sieg University, Germany
Wenbing Zhao, Cleveland State University, USA
Camelia Muñoz-Caro, Universidad de Castilla-La Mancha, Spain
Laurent Réveillère, Bordeaux Institute of Technology, France
Ewa Grabska, Jagiellonian University - Krakow, Poland

ADVCOMP Industry/Research Chairs
Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Helmut Reiser, Leibniz Supercomputing Centre (LRZ)-Garching, Germany
H. Metin Aktulga, Lawrence Berkeley National Lab, USA
Sameh Elnikety, Microsoft Research, USA
Umar Farooq, Amazon.com - Seattle, USA
Dmitry Fedosov, Forschungszentrum Juelich GmbH, Germany
Alice Koniges, Lawrence Berkeley Laboratory/NERSC, USA
Markus Kunde, German Aerospace Center & Helmholtz Association - Cologne, Germany
Peter Müller, IBM Zurich Research Laboratory- Rüschlikon, Switzerland
Simon Tsang, Applied Communication Sciences - Piscataway, USA
Anna Schwanengel, Siemens AG, Germany
Christoph Fuenfzig, Fraunhofer ITWM, Germany

ADVCOMP Publicity Chairs
Marie Lluberes, University of Puerto Rico at Mayagüez, USA
Sascha Opletal, University of Stuttgart, Germany
Álvaro Navas, Universidad Politecnica de Madrid, Spain
Iwona Ryszka, Jagiellonian University - Krakow, Poland

ADVCOMP 2014

Committee

ADVCOMP Advisory Chairs

Chih-Cheng Hung, Southern Polytechnic State University, USA
Juha Röning, Oulu University, Finland
Sigeru Omatu, Osaka Institute of Technology, Japan
Erich Schweighofer, University of Vienna, Austria
Paul Humphreys, University of Ulster, UK
Danny Krizanc, Wesleyan University, USA
Ivan Rodero, Rutgers University - Piscataway, USA
Ali Shawkat, CQ University of Australia - North Rockhampton, Australia
George Spanoudakis, City University London, UK
Vladimir Vlassov, KTH Royal Institute of Technology, Sweden
Jerry Trahan, Louisiana State University, USA
Dean Vucinic, Vrije Universiteit Brussel (VUB), Belgium
Rudolf Berrendorf, Bonn-Rhein-Sieg University, Germany
Wenbing Zhao, Cleveland State University, USA
Camelia Muñoz-Caro, Universidad de Castilla-La Mancha, Spain
Laurent Réveillère, Bordeaux Institute of Technology, France
Ewa Grabska, Jagiellonian University - Krakow, Poland

ADVCOMP Industry/Research Chairs

Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Helmut Reiser, Leibniz Supercomputing Centre (LRZ)-Garching, Germany
H. Metin Aktulga, Lawrence Berkeley National Lab, USA
Sameh Elnikety, Microsoft Research, USA
Umar Farooq, Amazon.com - Seattle, USA
Dmitry Fedosov, Forschungszentrum Juelich GmbH, Germany
Alice Koniges, Lawrence Berkeley Laboratory/NERSC, USA
Markus Kunde, German Aerospace Center & Helmholtz Association - Cologne, Germany
Peter Müller, IBM Zurich Research Laboratory- Rüschlikon, Switzerland
Simon Tsang, Applied Communication Sciences - Piscataway, USA
Anna Schwanengel, Siemens AG, Germany
Christoph Fuenfzig, Fraunhofer ITWM, Germany

ADVCOMP Publicity Chairs

Marie Lluberes, University of Puerto Rico at Mayagüez, USA
Sascha Opletal, University of Stuttgart, Germany
Álvaro Navas, Universidad Politecnica de Madrid, Spain
Iwona Ryszka, Jagiellonian University - Krakow, Poland

ADVCOMP 2014 Technical Program Committee

Witold Abramowicz, University of Economics - Poznań, Poland
H. Metin Aktulga, Lawrence Berkeley National Lab, USA
Sónia Maria Almeida da Luz, Polytechnic Institute of Leiria, Portugal / University of Extremadura, Spain
Alina Andreica, Babes-Bolyai University, Romania
Sulieman Bani-Ahmad, Al-Balqa Applied University, Jordan
Roberto Beraldi, "La Sapienza" University of Rome, Italy
Simona Bernardi, Centro Universitario de la Defensa / Academia General Militar - Zaragoza, Spain
Mario Marcelo Berón, National University of San Luis, Argentina
Rudolf Berrendorf, Bonn-Rhein-Sieg University, Germany
Ateet Bhalla, Oriental Institute of Science and Technology, India
Muhammad Naufal bin Mansor, University Malaysia Perlis, Malaysia
Pierre Borne, Ecole Centrale de Lille - Villeneuve d'Ascq, France
Kenneth P. Camilleri, University of Malta - Msida, Malta
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Yeh-Ching Chung, National Tsing Hua University, Taiwan
Marisa da Silva Maximiano, Escola Superior de Tecnologia e Gestão - Instituto Politécnico de Leiria, Portugal
Vieri del Bianco, Università dell'Insubria, Italy
Javier Diaz, Rutgers University, USA
Xing Cai, Simula Research Laboratory, Norway
Yves Caniou, Université de Lyon, France / University of Tokyo, Japan
Juan Carlos Dueñas López, Universidad Politécnica de Madrid, Spain
Cy Chan, Lawrence Berkeley National Laboratory, USA
Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Sameh Elnikety, Microsoft Research, USA
Javier Fabra, University of Zaragoza, Spain
Simon G. Fabri, University of Malta - Msida, Malta
Umar Farooq, Amazon.com - Seattle, USA
Mehdi Farshbaf-Sahih-Sorkhabi, Azad University - Tehran / Fanavaran co., Tehran, Iran
Dmitry Fedosov, Forschungszentrum Juelich GmbH, Germany
Mohammad-Reza Feizi-Derakhshi, University of Tabriz, Iran
Dan Feldman, MIT, USA
Bin Fu, University of Texas - Pan American, USA
Cheng Fu, Shanghai Advanced Research Institute, Chinese Academy of Sciences, China
Akemi Galvez Tomida, University of Cantabria, Spain
Rodrigo García Carmona, Universidad Politécnica de Madrid, Spain
Felix Jesus Garcia Clemente, University of Murcia, Spain
Leonardo Garrido, Tecnológico de Monterrey, Mexico
Wolfgang Gentzsch, HPC Consultant, Germany
Paul Gibson, Telecom & Management SudParis, France
Filippo Gioachin, Hewlett-Packard Laboratories, Singapore
Luis Gomes, Universidade Nova de Lisboa, Portugal
Teofilo Gonzalez, University of California - Santa Barbara, USA
Santiago Gonzalez de la Hoz, IFIC - Universitat de Valencia, Spain
Ewa Grabska, Jagiellonian University - Krakow, Poland
Bernard Grabot, ENIT, France

Adrian Muscat, University of Malta, Malta
Álvaro Navas, Universidad Politecnica de Madrid, Spain
Toan Nguyen, INRIA, France
Sigeru Omatu, Osaka Institute of Technology, Japan
Sascha Opletal, University of Stuttgart, Germany
Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
Mathias Pacher, Leibniz Universität Hannover, Germany
Kwangjin Park, Wonkwang University, Korea
Zornitza Petrova, Technical University of Sofia, Bulgaria
Meikel Poess, Oracle, USA
Radu-Emil Precup, "Politehnica" University of Timisoara, Romania
Luciana Rech, Universidade Federal de Santa Catarina, Brazil
Helmut Reiser, LRZ, Germany
Laurent Réveillère, Bordeaux Institute of Technology, France
Dolores Rexachs, Universidad Autónoma de Barcelona (UAB), Spain
Ivan Rodero, Rutgers University - Piscataway, USA
Alexey S. Rodionov, Institute of Computational Mathematics and Mathematical Geophysics, Russia
Juha Röning, Oulu University, Finland
Iwona Ryszka, Jagiellonian University - Krakow, Poland
Maytham Safar, Focus Consultancy, Kuwait
Subhash Saini, NASA, USA
Jose Francisco Salt Cairols, Universitat de Valencia-CSIC, Spain
Kenneth Scerri, University of Malta, Malta
Rainer Schmidt, Austrian Institute of Technology, Austria
Bruno Schulze, National Laboratory for Scientific Computing - LNCC -Petropolis - RJ, Brasil
Erich Schweighofer, Vienna University, Austria
Kewei Sha, Oklahoma City University, USA
Ali Shawkat, CQ University of Australia - North Rockhampton, Australia
Francesco Silvestri, University of Padova, Italy
George Spanoudakis, City University London, UK
Hari Subramoni, Ohio State University, USA
Saïd Tazi, INSA - Toulouse, France
Parimala Thulasiraman, University of Manitoba, Canada
Jerry Trahan, Louisiana State University, U.S.A.
Simon Tsang, Applied Communications Sciences, USA
José Valente de Oliveira, Universidade do Algarve, Portugal
Doru Vatau, University "Politehnica" of Timisoara, Romania
Vladimir Vlassov, KTH Royal Institute of Technology, Sweden
Dean Vučinić, Vrije Universiteit Brussel (VUB), Belgium
Zhonglei Wang, Intel, Germany
Zhi Wang, Florida State University, USA
Mudasser F. Wyne, National University, USA
Yinglong Xia, IBM, USA
Tse-Chen Yeh, Academia Sinica, China
Marek Zaremba, Université du Québec en Outaouais, Canada
Wenbing Zhao, Cleveland State University, U.S.A
Alcínia Zita Sampaio, Technical University of Lisbon, IST/ICIST, Portugal

Copyright Information

Table of Contents

The Greedy Approach to Dictionary-Based Static Text Compression on a Distributed System

Sergio De Agostino
Computer Science Department
Sapienza University
Rome, Italy
Email: deagostino@di.uniroma1.it

***Abstract*—The greedy approach to dictionary-based static text compression can be executed by a finite state machine. When it is applied in parallel to different blocks of data independently, there is no lack of robustness even on standard large scale distributed systems with input files of arbitrary size. Beyond standard large scale, a negative effect on the compression effectiveness is caused by the very small size of the data blocks. A robust approach for extreme distributed systems is presented in this paper, where this problem is fixed by overlapping adjacent blocks and preprocessing the neighborhoods of the boundaries.**

Keywords-lossless compression; string factorization; parallel computing; distributed system; scalability; robustness

I. Introduction

Static data compression implies the knowledge of the input type. With text, dictionary based techniques are particularly efficient and employ string factorization. The dictionary comprises typical factors plus the alphabet characters in order to guarantee feasible factorizations for every string. Factors in the input string are substituted by pointers to dictionary copies and such pointers could be either variable or fixed length codewords.

The optimal factorization is the one providing the best compression, that is, the one minimizing the sum of the codeword lengths. Efficient sequential algorithms for computing optimal solutions were provided by means of dymamic programming techniques [1] or by reducing the problem to the one of finding a shortest path in a directed acyclic graph [2]. From the point of view of sequential computing, such algorithms have the limitation of using an off-line approach. However, decompression is still on-line and a very fast and simple real time decoder outputs the original string with no loss of information. Therefore, optimal solutions are practically acceptable for read-only memory files where compression is executed only once. Differently, simpler versions of dictionary based static techniques were proposed which achieve nearly optimal compression in practice.

An important simplification is to use a fixed length code for the pointers, so that the optimal decodable compression for this coding scheme is obtained by minimizing the number of factors. Such variable to fixed length approach is robust since the dictionary factors are typical patterns of the input specifically considered. The problem of minimizing the number of factors gains a relevant computational advantage by assuming that the dictionary is *prefix* (*suffix*), that is, all the prefixes (suffixes) of a dictionary element are dictionary elements [3]-[5]. The left to right greedy approach is optimal only with suffix dictionaries. An optimal factorization with prefix dictionaries can be computed online by using a semi-greedy procedure [4], [5]. On the other hand, prefix dictionaries are easier to build by standard adaptive heuristics [6], [7]. These heuristics are based on an ”incremental” string factorization procedure [8], [9]. The most popular for prefix dictionaries is the one presented in [10]. However, the prefix and suffix properties force the dictionary to include many useless elements which increase the pointer size and slightly reduce the compression effectiveness. Moreover, the greedy approach to dictionary-based static text compression is optimal, in practice, for any kind of dictionary even if the theoretical worst case analysis shows that the multiplicative approximation factor with respect to optimal compression achieves the maximum length of a dictionary element. A more natural dictionary with no prefix and no suffix property is the one built by the heuristic in [11] or by means of separator characters as, for example, space, new line and punctuation characters for strings of a natural language. Finally, given an arbitrary dictionary, greedy static dictionary-based compression can be executed by a finite state machine.

Theoretical work was done, mostly in the nineties, to design efficient parallel algorithms on a random access parallel machine (PRAM) for dictionary-based static text compression [12]-[20]. Although the PRAM model is out of fashion today, shared memory parallel machines offer a good computational model for a first approach to parallelization. When we address the practical goal of designing distributed algorithms we have to consider two types of complexity, the interprocessor communication and the input-output mechanism. While the input/output issue is inherent to any parallel algorithm and has standard solutions, the communication cost of the computational phase after the distribution of the data among the processors and before the output of the final result is obviously algorithm-dependent. So, we need to limit the interprocessor communication and involve more local computation to design a practical algorithm. The simplest model for this phase is, of course, a simple array of processors with no interconnections and, therefore, no communication cost. Parallel decompression is, obviously, possible on this model [15]. With parallel compression, the main issue is the one concerning scalability and robustness. Standardly, the scale of a system is considered large when the number of nodes has the order of magnitude of a thousand. Modern distributed systems may nowadays consist of hundreds of thousands of nodes, pushing scalability well beyond traditional scenarios (extreme distributed systems).

In [21], an approximation scheme of optimal compression with static prefix dictionaries was presented for massively parallel architectures, using no interprocessor communication during the computational phase since it is applied in parallel to different blocks of data independently. The scheme is algorithmically related to the semi-greedy approach previously mentioned and implementable on extreme distributed systems because adjacent blocks overlap and the neighborhoods of the boundaries are preprocessed. However, with standard large scale the overlapping of the blocks, the preprocessing of the boundaries and the prefix property of the dictionary are not necessary to achieve nearly optimal compression. Starting from this observation, we present in this paper two implementations of the greedy approach to static text compression with an arbitrary dictionary on a large scale and an extreme distributed system, respectively.

In Section II, we describe the different approaches to dictionary-based static text compression. The previous work on parallel approximations of optimal compression with prefix dictionaries is given in Section III. Section IV shows the two implementations of the greedy approach for arbitrary dictionaries. Conclusions and future work are given in Section V.

II. Dictionary-Based Static Text Compression

In this section, we describe the main dictionary-based static compression techniques and make a greedy versus optimal analysis. Then, we provide a finite state machine implementation of the greedy approach with an arbitrary dictionary.

A. Optimal Solutions

As mentioned in the introduction, the dictionary comprises typical factors (including the alphabet characters) associated with fixed or variable length codewords. The optimal factorization is the one minimizing the sum of the codeword lengths and sequential algorithms for computing optimal solutions were provided by means of dynamic programming techniques [1] or by reducing the problem to the one of finding a shortest path in a directed acyclic graph [2]. With suffix dictionaries we obtain optimality by means of a simple left to right greedy approach, that is, advancing with the on-line reading of the input string by selecting the longest matching factor with a dictionary element. Such procedure can be computed in real time by storing the dictionary in a *trie* data structure (a trie is a tree where the root represents the empty string and the edges are labeled by the alphabet characters). If the dictionary is prefix and the codewords length is fixed, there is an optimal semi-greedy factorization which is computed by the procedure of Fig. 1 [4], [5]. At each step, we select a factor such that the longest match in the next position with a dictionary element ends to the rightest. Since the dictionary is prefix, the factorization is optimal. The algorithm can even be implemented in real time. The real time implementation employs an augmented trie data structure, obtained by modifying the original one [5].

```
$j$:=0; $i$:=0
repeat forever
    for $k = j + 1$ to $i + 1$ compute
        $h(k)$: $x_k ... x_{h(k)}$ is the longest match in the $k^{th}$ position
    let $k'$ be such that $h(k')$ is maximum
    $x_j ... x_{k'-1}$ is a factor of the parsing; $j := k'$; $i := h(k')$
```

Figure 1. The semi-greedy factorization procedure.

The semi-greedy factorization can be generalized to any dictionary by considering only those positions, among the ones covered by the current factor, next to a prefix that is a dictionary element [4]. We will see in the next subsection that the generalized semi-greedy factorization procedure is not optimal while the greedy one is not optimal even when the dictionary is prefix.

B. Greedy versus Optimal Factorizations

The maximum length of a dictionary element is an obvious upper bound to the multiplicative approximation factor of any string factorization procedure with respect to the optimal solution. We show that this upper bound is tight for the greedy and semi-greedy procedures when the dictionary is arbitrary. Such tightness is kept by the greedy procedure even if the dictionary is prefix. Let $baba^n$ be the input string and let $\{a, b, bab, ba^n\}$ be the dictionary. Then, the optimal factorization is b, a, ba^n while $bab, a, a, ..., a, ...a$ is the factorization obtained whether the greedy or the semi-greedy procedure is applied. On the other hand, with the prefix dictionary $\{a, b, ba, bab, ba^k : 2 \leq k \leq n\}$, the optimal factorization ba, ba^n is computed by the semi-greedy approach while the greedy factorization remains the same. These examples, obviously, prove our statement on the tightness of the upper bound.

C. The Finite State Machine Implementation

We show the finite state machine implementation producing the on-line greedy factorization of a string with an arbitrary dictionary. The most general formulation for a finite state machine M is to define it as a sixtuple $(A, B, Q, \delta, q_0, F)$ with an input alphabet A, an output alphabet B, a set of states Q, a transition function $\delta : QxA- > QxB^*$, an initial state q_0 and a set of accepting states $F \subseteq Q$. The trie storing the dictionary is a subgraph of the finite state machine diagram. It is well-known that each dictionary element is represented as a path from the root to a node of the trie where edges are labeled with an alphabet character (the root representing the empty string). The edges are directed from the parent to the child and the set of nodes represent the set of states of the machine. The output alphabet is binary and the factorization is represented by a binary string having the same length of the input string. The bits of the output string equal to 1 are those corresponding to the positions where the factors start. Since every string can be factorized, every state is accepting. The root represents the initial state. We need only to complete the function δ, by adding the the missing edges of the diagram. The empty string is associated as output to the edges in the trie. For each node, the outcoming edges represent a subset of the input alphabet. Let f be the string (or dictionary element) corresponding to the node v in the trie and a an alphabet character not represented by an edge outcoming from v. Let $fa = f_1 \cdots f_k$ be the on-line greedy factorization of fa and i the smallest index such that $f_{i+1} \cdots f_k$ is represented by a node w in the trie. Then, we add to the trie a directed edge from v to w with label a. The output associated with the edge is the binary string representing the sequence of factors $f_1 \cdots f_i$. By adding such edges, the machine is entirely defined. Redefining the machine to produce the compressed form of the string is straightforward.

III. Previous Work

Given an arbitrary dictionary, for every integer k greater than 1 there is an O(km) time, O(n/km) processors distributed algorithm factorizing an input string S with a cost which approximates the cost of the optimal factorization within the multiplicative factor $(k+m-1)/k$, where n and m are the lengths of the input string and the longest factor respectively [12]. However, with prefix dictionaries a better approximation scheme was presented in [21], producing a factorization of S with a cost approximating the cost of the optimal factorization within the multiplicative factor $(k+1)/k$ in O(km) time with O(n/km) processors. This second approach was designed for massively parallel architecture and is suitable for extreme distributed systems, when the scale is beyond standard large values. On the other hand, the first approach applies to standard small, medium and large scale systems. Both approaches provide approximation schemes for the corresponding factorization problems since the multiplictive approximation factors converge to 1 when km converge to n. Indeed, in both cases compression is applied in parallel to different blocks of data independently. Beyond standard large scale, adjacent blocks overlap and the neighborhoods of the boundaries are preprocessed.

To decode the compressed files on a distributed system, it is enough to use a special mark occurring in the sequence of pointers each time the coding of a block ends. The input phase distributes the subsequences of pointers coding each block among the processors. Since a copy of the dictionary is stored in every processor, the decoding of the blocks is straightforward.

In the following two subsections, we describe the two approaches. Then, how to speed up the preprocessing phase of the second approach is described in the last subsection. In the next section, we present new results by arguing that we can relax on the requirement of computing a theoretical approximation of optimal compression since, in practice, the greedy approach is optimal on data blocks sufficiently long. On the other hand, when the blocks are too short since the scale of the distributed system is beyond standard values, the overlapping of the adjacent blocks and the preprocessing of the neighborhoods of the boundaries are sufficient to garantee the robustness of the greedy approach.

A. Standard Scale Distributed Systems

Given an input string of length n, we simply apply in parallel optimal compression to blocks of length km, with k integer greater than one and m maximum length of a factor as stated at the beginning of this section. Every processor stores a copy of the dictionary. For arbitrary dictionary, we execute the dynamic programming procedure computing the optimal factorization of a string in linear time [1] (the procedure in [2] is pseudo-linear for fixed-length coding and, even, super linear for variable length). Obviously, this works for prefix and suffix dictionaries as well and, in any case, we know the semi-greedy and greedy approach are implementable in linear time. It follows that the algorithm requires O(km) time with n/km processors and the multiplicative approximation factor is $(k+m-1))/k$ with respect to any factorization. Indeed, when the boundary cuts a factor the suffix starting the block and its substrings might not be in the dictionary. Therefore, the multiplicative approximation factor follows from the fact that $m-1$ is the maximum length for a proper suffix as shown in Fig. 2 (sequence of plus signs in parentheses). If the dictionary is suffix, the multiplicative approximation factor is $(k+1))/k$ since each suffix of a factor is a factor.

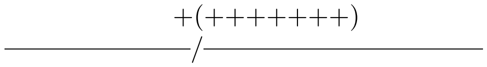


Figure 2. The making of the surplus factors.

The approximation scheme is suitable only for standard scale systems unless the file size is very large. In effect, the block size must be the order of kilobytes to guarantee robustness. Beyond standard large scale, overlapping of adjacent blocks and a preprocessing of the boundaries is required as we will see in the next subsection.

B. Beyond Standard Large Scale

With prefix dictionaries a better approximation scheme was presented in [21]. During the input phase blocks of length $m(k+2)$, except for the first one and the last one which are $m(k+1)$ long, are broadcasted to the processors. Each block overlaps on m characters with the adjacent block to the left and to the right, respectively (obviously, the first one overlaps only to the right and the last one only to the left).

We call a *boundary match* a factor covering positions in the first and second half of the $2m$ characters shared by two adjacent blocks. The processors execute the following algorithm to compress each block:

- For each block, every corresponding processor but the one associated with the last block computes the boundary match between its block and the next one ending furthest to the right, if any;

- each processor computes the optimal factorization from the beginning of its block to the beginning of the boundary match on the right boundary of its block (or the end of its block if there is no boundary match).

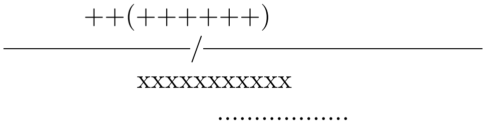


Figure 3. The making of a surplus factor.

Stopping the factorization of each block at the beginning of the right boundary match might cause the making of a surplus factor, which determines the multiplicative approximation factor $(k+1)/k$ with respect to any other factorization. Indeed, as it is shown in Fig. 3, the factor in front of the right boundary match (sequence of x's) might be extended to be a boundary match itself (sequence of plus signs) and to cover the first position of the factor after the boundary (dotted line). Then, the approximation scheme produces a factorization of S with a cost approximating the cost of the optimal factorization within the multiplicative factor $(k+1)/k$ in O(km) time with O(n/km) processors.

In [21], it is shown experimentally that for $k = 10$ the compression ratio achieved by such factorizarion is about the same as the sequential one and, consequently, the approach is suitable for extreme distributed systems, as we will explain in the next section.

C. Speeding up the Preprocessing

The parallel running time of the preprocessing phase computing the boundary matches is O(m^2) by brute force. To lower the complexity to O(m), an augmented trie data structure is needed. For each node v of the trie, let f be the dictionary element corresponding to v and a an alphabet character not represented by an edge outcoming from v. Then, we add an edge from v to w with label a, where w represents the longest proper suffix of fa in the dictionary. Each processor has a copy of this augmented trie data structure and first preprocess the $2m$ characters overlapped by the adjacent block on the left boundary and, secondly, the ones on the right boundary. In each of these two sub-phases, the processors advance with the reading of the $2m$ characters from left to right, starting from the first one while visiting the trie starting from the root and using the corresponding edges. A temporary variable t_2 stores the position of the current character during the preprocessing while another temporary variable t_1 is, initially, equal to t_2. When an added edge of the augmented structure is visited, the value $t = t_2 - d + 1$ is computed where d is the depth of the node reached by such edge. If t is a position in the first half of the $2m$ characters, then t_1 is updated by changing its value to t. Else, the procedure stops and t_2 is decreased by 1. If t_2 is a position in the second half of the $2m$ characters then t_1 and t_2 are the first and last position of a boundary match, else there is no boundary match.

IV. The Greedy Approach

In practice, greedy factorization is nearly optimal. As a first approach, we simply apply in parallel left to right greedy compression to blocks of length km. With standard scale systems, the block size must be the order of kilobytes to guarantee robustness. Each of the O(n/km) processors could apply the finite state machine implementation of subsection II.C to its block.

Beyond standard large scale, overlapping of adjacent blocks and a preprocessing of the boundaries are required as for the optimal case. Again, during the input phase overlapping blocks of length $m(k+2)$ are broadcasted to the processors as in the previous section. On the other hand, the definition of boundary match is extended to those factors, which are suffixes of the first half of the $2m$ characters shared by two adjacent blocks. The following procedure, even if it is not an approximation scheme from a theoretical point of view, performs similarly (observe that, in this case, we compute the longest boundary match rather than the one ending furthest to the right):

- For each block, every corresponding processor but the one associated with the last block computes the longest boundary match between its block and the next one;

- each processor computes the greedy factorization from the end of the boundary match on the left boundary of its block to the beginning of the boundary match on the right boundary.

To lower the parallel running time of the preprocessing phase to $\mathrm{O}(m)$, the same augmented trie data structure, described in the previous section, is needed but, in this case, the boundary matches are the longest ones rather than the ones ending furthest to the right. Then, besides the temporary variables t_1 and t_2, employed by the preprocessing phase described in the previous section, two more variables τ_1 and τ_2 are required and, initially, equal to t_1 and t_2. Each time t_1 must be updated by such preprocessing phase, before it the value $t_2 - t_1 + 1$ is compared with $\tau_2 - \tau_1$. If it is greater or τ_2 is smaller than the last position of the first half of the $2m$ characters, τ_1 and τ_2 are set equal to t_1 and $t_2 - 1$. Then, t_1 is updated. At the end of the procedure, τ_1 and τ_2 are the first and last positions of the longest boundary match. We wish to point out that there is always a boundary match that is computed, since the final value of τ_2 always corresponds to a position equal either to one in the second half of the $2m$ characters or to the last position of the first half. Again, after preprocessing each of the $\mathrm{O}(n/km)$ processors could apply the finite state machine implementation of subsection II.C to its block.

The approach is nearly optimal for k = 10, as the approximation scheme of previous section. The compression ratio achieved by such factorizarion is about the same as the sequential one. Considering that typically the average match length is 10, one processor can compress down to 100 bytes independently. This is why the approximation scheme was presented for massively parallel architecture and the approach, presented in this section, is suitable for extreme distributed systems, when the scale is beyond standard large values. Indeed, with a file size of several megabytes or more, the system scale has a greater order of magnitude than the standard large scale parameter. We wish to point out that the computation of the boundary matches is very relevant for the compression effectiveness when an extreme distributed system is employed since the sub-block length becomes much less than 1K. With standard large scale systems the block length is several kilobytes with just a few megabytes to compress and the approach using boundary matches is too conservative.

V. Conclusion

We presented parallel implementations of the greedy approach to dictionary-based static text compression suitable for standard and non-standard large scale distributed systems. In order to push scalability beyond what is traditionally considered a large scale system, a more involved approach distributes overlapping blocks to compute boundary matches. These boundary matches are relevant to maintain the compression effectiveness on a so-called extreme distributed system. If we have a standard small, medium or large scale system available, the approach with no boundary matches can be used. The absence of a communication cost during the computation guarantees a linear speed-up. Moreover, the finite state machine implementation speeds up the execution of the distributed algorithm in a relevant way when the data blocks are large, that is, when the size of the input file is large and the size of the distributed system is relatively small. As future work, experiments on parallel running times should be done to see how the preprocessing phase effects on the linear speed-up when the system is scaled up beyond the standard size and how relevant the employment of the finite state machine implementation is when the data blocks are very small.

References

[1] R. A. Wagner, ”Common Phrases and Minimum Text Storage,” Communications of the ACM, vol. 16, 1973, pp. 148-152.

[2] E. J. Shoegraf and H. S. Heaps, ”A Comparison of Algorithms for Data Base Compression by Use of Fragments as Language Elements,” Information Storage and Retrieval, vol. 10, 1974, pp. 309-319.

[3] M. Cohn and R. Khazan, ”Parsing with Suffix and Prefix Dictionaries,” Proceedings IEEE Data Compression Conference, 1996, pp. 180-189.

[4] M. Crochemore and W. Rytter, Jewels of Stringology, World Scientific, 2003.

[5] A Hartman and M. Rodeh, ”Optimal Parsing of Strings,” Combinatorial Algorithms on Words (eds. Apostolico, A., Galil, Z.), Springer, 1985, pp. 155-167.

[6] T. C. Bell, J. G. Cleary and I. H. Witten, Text Compression, Prentice Hall, 1990.

[7] J. A. Storer, Data Compression: Methods and Theory, Computer Science Press, 1988.

[8] A. Lempel and J. Ziv, ”On the Complexity of Finite Sequences,” IEEE Transactions on Information Theory, vol. 22, 1976, pp. 75-81.

[9] J. Ziv and A. Lempel, ”Compression of Individual Sequences via Variable-Rate Coding,” IEEE Transactions on Information Theory, vol. 24, 1978, pp. 530-536.

[10] T. A. Welch, ”A Technique for High-Performance Data Compression,” IEEE Computer, vol. 17, 1984, pp. 8-19.

[11] V. S. Miller and M. N. Wegman, ”Variations on Theme by Ziv - Lempel,” Combinatorial Algorithms on Words (eds. Apostolico, A., Galil, Z.), Springer, 1985, pp. 131-140

[12] L. Cinque, S. De Agostino and L. Lombardi, ”Scalability and Communication in Parallel Low-Complexity Lossless Compression,” Mathematics in Computer Science, vol. 3, 2010, pp. 391-406.

[13] S. De Agostino, Sub-Linear Algorithms and Complexity Issues for Lossless Data Compression, Master’s Thesis, Brandeis University, 1994.

[14] S. De Agostino, Parallelism and Data Compression via Textual Substitution, Ph. D. Dissertation, Sapienza University of Rome, 1995.

[15] S. De Agostino, ”Parallelism and Dictionary-Based Data Compression,” Information Sciences, vol. 135, 2001, pp. 43-56.

[16] S. De Agostino S. and J. A. Storer, ”Parallel Algorithms for Optimal Compression Using Dictionaries with the Prefix Property,” Proceedings IEEE Data Compression Conference, 1992, pp. 52-61.

[17] D. S. Hirschberg and L. M. Stauffer, ”Parsing Algorithms for Dictionary Compression on the PRAM,” Proceedings IEEE Data Compression Conference, 1994, pp. 136-145.

[18] D. S. Hirschberg and L. M. Stauffer, ”Dictionary Compression on the PRAM,” Parallel Processing Letters, vol. 7, 1997, pp. 297-308.

[19] H. Nagumo, M. Lu and K. Watson, ”Parallel Algorithms for the Static Dictionary Compression,” Proceedings IEEE Data Compression Conference, 1995, pp. 162-171.

[20] L. M. Stauffer and D. S. Hirschberg, ”PRAM Algorithms for Static Dictionary Compression,” Proceedings International Symposium on Parallel Processing, 1994, pp. 344-348.

[21] D. Belinskaya, S. De Agostino and J. A. Storer, ”Near Optimal Compression with respect to a Static Dictionary on a Practical Massively Parallel Architecture,” Proceedings IEEE Data Compression Conference, 1995, pp. 172-181.

Efficient Splitting Characteristic Method for Solving Multi-component Aerosol Spatial Transports in Atmospheric Environment

Dong Liang and Kai Fu

Department of Mathematics and Statistics, York University
Toronto, Ontario, M3J 1P3, Canada
School of Mathematics, Shandong University
Jinan, Shandong, 250100, China
Email: dliang@yorku.ca and kfu@yorku.ca

Wenqia Wang

School of Mathematics
Shandong University
Jinan, Shandong, 250100, China
Email: wangwq@sdu.edu.cn

Abstract—**In this work, we develop a splitting characteristic method for solving multi-component atmospheric aerosol spatial dynamics, which can efficiently evaluate aerosol spatial transports by using large time step sizes. The method can compute the multi-component aerosol distributions in high-dimensional domains with large ranges of concentrations and for different aerosol types. Numerical tests show the computational efficiency of the proposed method. An actual simulation focusing on the sulfate pollution is taken in the domain with a varying wind field. We also simulate multi-component aerosol transports in a large area in the southeast of America. The developed algorithm can be applied for the large scale predictions of multi-component aerosols in multi-regions and multi-levels in atmospheric environment.**

***Keywords*–Atmospheric aerosol transport; Multi-component; Splitting; Characteristic method; Efficiency; High accuracy.**

I. Introduction

Global climate change and warming in atmosphere have been widely recognized. As one of most important constituents, aerosols have a direct radiative forcing by scattering and absorbing solar and infrared radiation in atmosphere, while they have an indirect radiative forcing associated with the changes in cloud properties by decreasing the precipitation efficiency of warm clouds. In these processes, the physical states including the gas and aerosol phases (i.e., gas, liquid and solid) and the composition of aerosols including the aerosol-associated water mass, and the multi-components of aerosols are of great significance. Numerical modeling has been playing a key role in the study of aerosol processes and aerosol concentration distributions in the atmospheric environment prediction and the air quality control.

Aerosol transport model in atmosphere is a complex multi-component system that involves several physical and chemical processes, such as emission, transport, dispersion, aerosol dynamics and aerosol chemistry processes. The studied area usually covers a large region. Odman and Russell [12] studied the URM model and the UAM-AIM model in the aerosol simulation in the southern California. Grell et al. [5] developed the WRF/Chem model and simulated the aerosol distributions in the eastern United States and contiguous areas. Further applications have been done to some areas in the Europe [11]. However, in these computations, very small time steps have to be used in order to ensure the numerical stability of the numerical schemes, which brings a huge cost of computation and some limitation of applications. Therefore, it has been an important task to develop efficient numerical algorithm for multi-component aerosol transports.

In this paper, we present our new development of the efficient splitting method for solving aerosol transports in atmosphere. We consider general spatial aerosol transport problems in atmosphere and develop the splitting characteristic method for modeling multi-component aerosol transports. For the spatial transport systems, we propose the characteristic finite difference method to solve the transport process by combining with the operator splitting technique to deal with processes of emission, aerosol dynamics and chemical process. The methods of characteristics to treat convection terms were studied for high dimensional convection-diffusion problems in porous media [1][2][4][9]. In this study, we take the important advantages of both the characteristic method and the operator splitting technique to solve the spatial aerosol transport dynamical system, which can be solved in parallel computation. The developed method can efficiently compute the multi-component aerosol transport dynamics in high-dimensional domains with a large range of aerosol concentrations and for different types of aerosols. Numerical tests show the computational efficiency of the proposed method for both the spatially homogeneous aerosol dynamic problems and the spatial transport aerosol dynamic problems. An actual simulation focusing on the sulfate pollution is then taken in the domain with a varying wind field. Finally, a simulation of multi-component aerosol transports in a large area in the southeast of America is taken, which shows clearly that city areas usually have high $PM_{2.5}$ concentration and marine aerosols have affection in marine and coast areas. The developed algorithm can be applied for the large scale predictions of multi-component aerosols in multi-regions and multi-levels in environment.

The paper is arranged as follows. The mathematical model of multi-component aerosol dynamics is presented in Section II. The splitting numerical method is proposed for the multi-component aerosol transport equations in Section III. Numerical simulations are given in Section IV. Some conclusions are addressed in Section V.

II. Multi-component Aerosol Dynamic Models

The multi-component aerosol spatial transport models are:

$$\begin{aligned}\frac{\partial c_l}{\partial t} &= -\mathbf{U}\cdot\nabla c_l + \nabla\cdot(K\nabla c_l) + \mathcal{L}_{Aerosol}(\vec{c}) \\ &\quad + E_{l_{Emission}}(\vec{x},v,t), \quad l=1,2,\cdots,s, \end{aligned} \tag{1}$$

$$c_l(\vec{x},v,t) = c_l^{IN}(\vec{x},v,t),\ \vec{x}\in\Gamma_{IN}, \tag{2}$$

$$K\nabla c_l\cdot\vec{\nu} = 0,\ \vec{x}\in\Gamma_{OUT}, \tag{3}$$

$$K\nabla c_l\cdot\vec{\nu} = 0,\ \vec{x}\in\Gamma_{TOP}, \tag{4}$$

$$K\nabla c_l\cdot\vec{\nu} = E_{l,g},\ \vec{x}\in\Gamma_{GR}, \tag{5}$$

$$c_l(\vec{x},v,0) = c_l^0(\vec{x},v), \tag{6}$$

where $c_l(\vec{x},v,t)$ is the mass concentration of aerosol species l at position $\vec{x}=(x,y,z)$ in space at time t, and at particle volume v; $\vec{c}=(c_1,c_2,\cdots,c_s)$; c_l^0 and c_l^{IN} are initial and inflow boundary values. Ω is a three dimensional rectangle computational domain, and the boundary of the domain is partitioned into $\partial\Omega=\Gamma_{IN}\cup\Gamma_{OUT}\cup\Gamma_{GR}\cup\Gamma_{TOP}$, where Γ_{IN} denotes the inflow lateral boundary, Γ_{OUT} denotes the outflow lateral boundary, Γ_{GR} is the ground level portion of the boundary, and Γ_{TOP} is the top level portion of the boundary. $\vec{\nu}$ is the unit outward normal to the boundary $\partial\Omega$. The volume interval is $[V_{\min},V_{\max}]$. At $v=V_{\min}$, the concentrations of aerosols are zero. $E_{l,g}$ is the rate of ground level emission of the specie.

III. The Splitting Numerical Scheme

Let Δt be the splitting time step size, and $t^n=n\Delta t$. In a time interval $(t^n,t^{n+1}]$, the splitting numerical scheme for the multi-component aerosol spatial transport problems is proposed as:

Step 1. From c_l^n, the value at $t=t^n$, solve the aerosol emission process over $(t^n,t^{n+1}]$

$$\frac{\partial c_l}{\partial t} = E_{l_{Emission}}(\vec{x},v,t), \tag{7}$$

$$c_l(\vec{x},v,t^n) = c_l^n(\vec{x},v)\ \vec{x}\in\Omega. \tag{8}$$

The emission rates of multi-component aerosol particles are different due to differences in the emission characteristics of emitted particles from a variety of natural and anthropogenic sources. We can get the solution $c_{l,emission}^{n+1}$ of problem (7)(8) at $t=t^{n+1}$.

Step 2. Solve the multi-component aerosol dynamic process in $(t^n,t^{n+1}]$ with $c_{l,emission}^{n+1}$ being the initial value at $t=t^n$.

$$\frac{\partial c_l}{\partial t} = \mathcal{L}_{Aerosol}(c_1,c_2,\cdots,c_s;\vec{x},v,t) \tag{9}$$

$$c_l(\vec{x},v,t^n) = c_{l,emission}^{n+1}(\vec{x},v),\ l=1,2,\cdots,s. \tag{10}$$

The aerosol process includes the aerosol dynamic processes (condensation, coagulation), and the aerosol chemical process. Numerical methods, such as modal methods, sectional methods, and wavelet methods ([10]), can be used to solve the aerosol dynamic process. The aerosol chemical process is described by aerosol thermodynamic equilibrium equations, which can be solved ISORROPIA II ([7]) and the moving-cut HDMR method ([3] [8]). Denote the solution by $c_{l,aero}^{n+1}$.

Step 3. From $c_{l,aero}^{n+1}$ as the value at $t=t^n$, the spatial transport process is solved in $(t^n,t^{n+1}]$.

$$\frac{\partial c_l}{\partial t} = -\mathbf{U}\cdot\nabla c_l + \nabla\cdot(K\nabla c_l), \tag{11}$$

$$c_l(\vec{x},v) = c_l^{IN}(\vec{x},v,t^n),\vec{x}\in\Gamma_{IN}, \tag{12}$$

$$K\frac{\partial c_l(\vec{x},v)}{\partial\nu} = 0,\vec{x}\in\Gamma_{OUT}, \tag{13}$$

$$K\frac{\partial c_l(\vec{x},v)}{\partial\nu} = 0,\vec{x}\in\Gamma_{TOP}, \tag{14}$$

$$K\frac{\partial c_l(\vec{x},v)}{\partial\nu} = E_{l,g},\ \vec{x}\in\Gamma_{GR}, \tag{15}$$

$$c_l(\vec{x},v,t^n) = c_{l,aero}^{n+1}(\vec{x},v). \tag{16}$$

We propose a characteristic method to solve systems (11)-(16) in this step. Finally, the solution c_l^{n+1} is obtained at $t=t^{n+1}$.

IV. Numerical simulation

We now give numerical simulations of the spatially homogeneous aerosol dynamics and the multi-component atmospheric aerosol spatial transport dynamics.

Example 1.

We consider the spatially homogeneous three-component aerosol dynamics of aerosol water, black carbon and sulfate components. The initial values are tri-modal log-normal distributions. The spatially homogeneous three-component aerosol dynamic equations are

$$\begin{aligned}\frac{\partial c_i(m,t)}{\partial t} &= \mathcal{L}_{Aerosol}(\vec{c}) := \eta_i c(m,t) - \frac{\partial(m\eta c_i)}{\partial m} \\ &+ \int_{M_{\min}}^{m-M_{\min}} \beta(m,m-m')c_i(m,t)\frac{c(m-m',t)}{m-m'}dm' \\ &- c_i(m,t)\int_{M_{\min}}^{M_{\max}}\beta(m,m')\frac{c(m',t)}{m'}dm',\end{aligned} \tag{17}$$

$$c_i(M_{min},t)=0,\quad t\in[0,T], \tag{18}$$

$$c_i(m,0)=c_i^0(m),\quad m\in\Omega,\quad i=1,2,3. \tag{19}$$

where $t>0$ is the time, and $T>0$ is the time period; the finite mass interval $I=[M_{\min},M_{\max}]$ where $M_{\min}>0$ is the minimal mass and $M_{\max}>0$ is a finite maximal mass. $\beta(m,m')$ is the coagulation kernel function. $\eta_i(m,t)$ is the growth rate of species i, $\eta(m,t)=\sum_{i=1}^3\eta_i(m,t)$. $c_i(m,t)$ is the mass concentration of aerosol species i and $c(m,t)=\sum_{i=1}^3 c_i(m,t)$. Eq. (17) forms a system of nonlinear integral-differential equations on time and particle mass m.

Let

$$\begin{aligned}&F_i\big(m,t,c_i(m,t),c(m,t)\big) \\ &= \int_{M_{\min}}^{m-M_{\min}}\beta(m',m-m')c_i(m',t)\frac{c(m-m',t)}{m-m'}dm' \\ &\quad - c_i(m,t)\int_{M_{\min}}^{M_{\max}}\beta(m,m')\frac{c(m',t)}{m'}dm'.\end{aligned} \tag{20}$$

We propose the time second-order characteristic method for the three-component aerosol dynamic equations (17) as

$$\begin{aligned}&\frac{c_{i,h}^{k+1}(m) - c_{i,h}^{k}(\bar{m})}{\Delta t} + \eta \frac{c_{i,h}^{k+1}(m) + c_{i,h}^{k}(\bar{m})}{2} \\ &-\eta_i \frac{c_{h}^{k+1}(m) + c_{h}^{k}(\bar{m})}{2} = \frac{3}{2} F_i\big(\bar{m}, c_{i,h}(\bar{m}, t^k), c_h(\bar{m}, t^k)\big) \\ &-\frac{1}{2} F_i\big(\bar{\bar{m}}, c_{i,h}(\bar{\bar{m}}, t^{k-1}), c_h(\bar{\bar{m}}, t^{k-1})\big), \\ &\qquad i = 1, 2, 3, \end{aligned} \tag{21}$$

where $\bar{m}^k$ and $\bar{\bar{m}}^{k-1}$ are the intersection points of the characteristic curve at time level $t = t^k$ and $t = t^{k-1}$ respectively.

TABLE I. COMPARISON OF ERRORS BY OUR METHOD AND S-C-FEM FOR THE THREE-COMPONENT AEROSOL CONDENSATION PROBLEM WITH A TRI-MODAL INITIAL DISTRIBUTION.

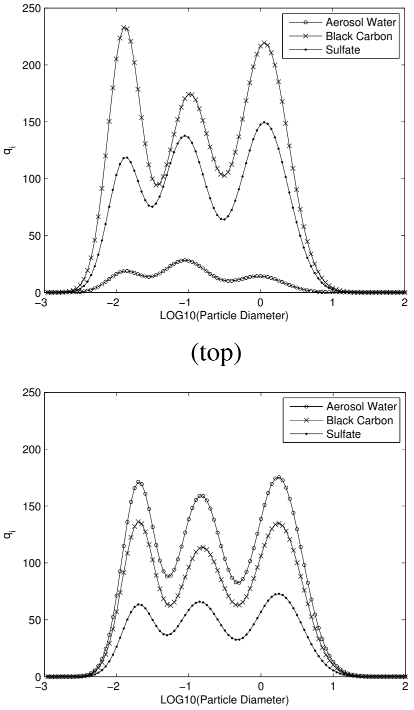
	Δt (hour)	$\frac{T}{8}$	$\frac{T}{16}$	$\frac{T}{32}$	$\frac{T}{48}$
Our method	E_∞	5.1358e-3	1.3000e-3	3.3209e-4	1.4579e-4
	Ratio	-	1.9821	1.9688	2.0304
	E_2	3.2027e-3	7.9209e-4	1.9431e-4	8.4280e-5
	Ratio	-	2.0155	2.0273	2.0601
S-C-FEM	E_∞	3.6335e-1	1.7538e-1	7.8061e-2	4.5141e-2
	Ratio	-	1.0509	1.1678	1.3508
	E_2	2.2616e-1	1.0915e-1	4.8613e-2	2.8123e-2
	Ratio	-	1.0510	1.1669	1.3498

We solve the general problem on the $\Omega = [5.236 \times 10^{-22}, 5.236 \times 10^{-7}]$g and time interval $[0, T] = [0, 5]$ hours. Table I presents the comparison of the errors and ratios in time step of the predicted results by our method and the standard characteristics FEM scheme (S-C-FEM), for the three-component problems with a same three-modal distribution for each species. It is obvious that our method is of second-order accuracy in time step but the standard characteristic finite element method (S-C-FEM) is only of first-order accuracy in time step.

The predicted mass concentration distributions of the of aerosol water, black carbon and sulfate components are shown in Fig. 1 for problem with different initial species distributions. We can see that the peaks of the distributions of aerosol water, black carbon and sulfate components change a lot during the simulation. For example, at time $T = 0$, the mass distribution of aerosol water has the highest peak value of 28.307 in the accumulation mode, while at time $T = 5$, the highest peak value is 175.28 located in the coarse mode. Due to the highest condensation rate among the three species, the concentration of aerosol water is the smallest at time $T = 0$ hour, but becomes the largest at time $T = 5$ hours. For the distribution of the total mass, we can see that the peak values keep almost unchange.

Example 2.

We then do a simulation to the spatial multicomponent aerosol transport by the developed method, where the WRF ([13]) is used to provide the information of wind components, temperature, pressure, water vapor, clouds, and rainfalls, etc. The studied domain is a 2400 km×1800 km area centered at 79.25°W longitude and 43.40°N latitude with the horizontal grid dimension 40 (west-east)×30 (south-north) with the spacing of 60 km, the vertical interval of the domain consist of 27 layers up to approximately 20.1 km. The layers of the aerosol transport model are aligned with the layers in the meteorological WRF model, with a vertical mesh interval from 60 m near the surface to 1.5 km at the domain top. The emission inventory used in the simulation is the EPA U.S. National Emissions Inventory (NEI-05) [14]. The simulation period presented here consists of the last 72 hours of a 108-hour simulation.



(top)

(bottom)

Figure 1. Initial distributions of the mass concentrations of aerosol water, black carbon and sulfate components (top) and the numerical solutions at time $T = 5$ hours (bottom) for the multi-component condensation problem. The condensation rates of the aerosol water (aw), black carbon (bc) and sulfate (sul) are $\alpha_{aw} = 1.6 \times 10^{-8}$ hour $^{-1}$, $\alpha_{bc} = 7 \times 10^{-9}$ hour $^{-1}$, $\alpha_{sul} = 3 \times 10^{-9}$ hour $^{-1}$, $M_{\min} = 5.236 \times 10^{-22}$g, $M_{\max} = 5.236 \times 10^{-7}$g.

Fig. 2 presents the hourly predicted concentrations of total $PM_{2.5}$ total mass and the sum of $PM_{2.5}$ sulfate, ammonium, nitrate, sodium and chloride concentrations in New York, the rural and marine area, which show that these five species are the mainly constituents of $PM_{2.5}$ total mass, and the sum of their concentrations determines the variation trends of total concentration of $PM_{2.5}$ total mass.

The variation of the nitrate concentration and temperature of NY are presented in Fig. 3. Thus, it is obvious that low temperatures are more favorable to the generation of the nitrate particulate phase. This result agrees well with existing theories and experiment.

Example 3.

Finally, we simulate multi-component aerosol spatial transports in a large regional area in the southeast of the America. The studied domain is a 40×40 mesh grid, which centered at 82.5°W longitude and 36°N latitude and the spacing is 20km. We take a 228 hours simulation and analyze numerically the last 120 hours simulation results, which will limit the influence of the default initial condition of chemical species

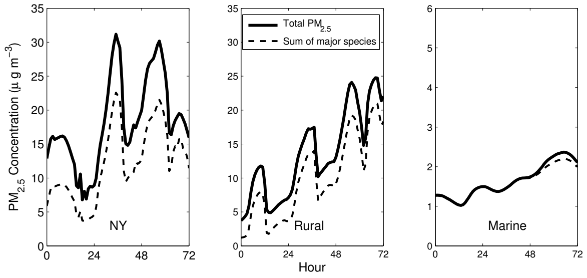


Figure 2. Predicted concentrations of total $PM_{2.5}$ mass concentration and the sum of major species, $PM_{2.5}$ sulfate, ammonium, nitrate, sodium and chloride concentrations in New York, the rural and marine area.

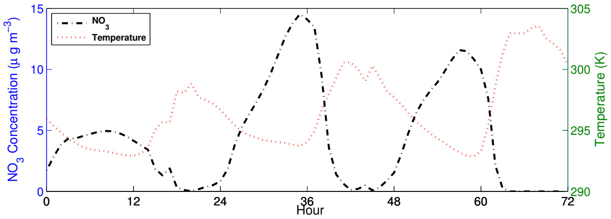
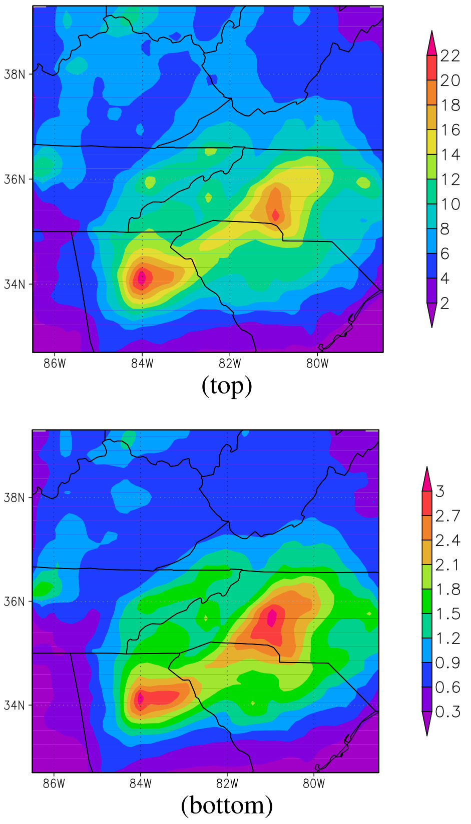


Figure 3. Comparisons of hourly predicted concentrations of $PM_{2.5}$ nitrate and temperature in New York.



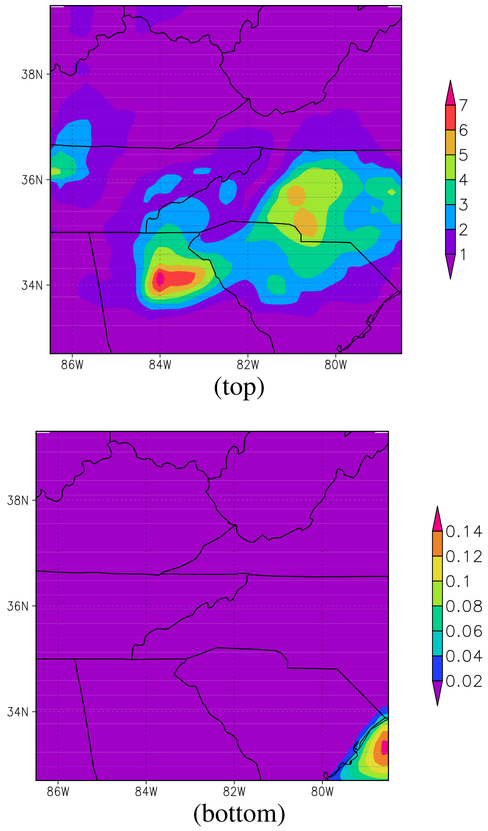
(top)

(bottom)

Figure 4. The predicted ground-level concentration distributions of $PM_{2.5}$ total mass (top) and ammonium (bottom) (μg m^{-3}).

concentration. For transport process, the time step size is used as 1800s for the characteristic finite difference method, and the time step of the aerosol process step is used as 120s in the system.

Fig. 4 and Fig. 5 show the averaged ground-level concentrations of $PM_{2.5}$ total mass, nitrate, ammonium, chloride (μg m^{-3}). We can see that, nitrate accounts for a large portion of the $PM_{2.5}$ total mass in the predicted domain, and $PM_{2.5}$ nitrate, ammonium generally have high concentrations near cities. For instance, the area near Atlanta (84.39°W, 33.67°N) has the highest predicted concentrations of $PM_{2.5}$ nitrate and ammonium, and total mass with over 7 μg m^{-3}, 3 μg m^{-3}, and 22 μg m^{-3}, respectively. Marine aerosol of chloride only exists in the marine and coast area,which is 0.03 to 0.14 μg m^{-3} for chloride from the coast to the open sea.



(top)

(bottom)

Figure 5. The predicted ground-level concentration distributions of nitrate (top) and chloride (bottom) (μg m^{-3}).

V. Conclusion

In this paper, we developed a characteristic method, combining with the splitting technique, for multi-component aerosol spatial transports which involve several physical and chemical processes of advection, dispersion, emission, deposition and aerosol dynamic processes. The developed algorithm is robust, efficient, and highly accurate for the spatially homogeneous aerosol dynamics and the spatial multi-component aerosol transports, in which large time steps can be used in simulation. A simulation of multi-component aerosol transports over an area in the northeast America was further done by the algorithm. The 72-hour simulation results at the New York, a rural area and a marine area show that the $PM_{2.5}$ has the highest concentration among the three areas, while the marine area has the lowest. The major species of aerosols in the New York and rural areas are sulfate, nitrate and ammonia, while chloride and sodium make the most portion of the marine aerosol. The results also exhibited that the nitrate concentration varies inversely with the temperature. The comparisons of simulation results with and without dry deposition provided the fact that dry deposition makes great contribution to the aerosol decrease, and has more influence to larger particles. Finally, a real life multi-component aerosol transport simulation was carried out over a large area in the southeast of America, which showed that, city areas usually have high concentration of $PM_{2.5}$, and marine aerosols only have affection of marine and coast areas. The developed algorithm can be applied to efficiently simulate multi-component atmospheric aerosol spatial transports in large domains.

Acknowledgment

The work was supported partially by Natural Sciences and Engineering Research Council of Canada and by Natural Science Foundation of China under grant 11271232.

References

[1] T. Arbogast and Ch.-S. Huang, “A fully conservative Eulerian-Lagrangian method for a convection-diffusion problem in a solenoidal field”, J. Comput. Phys., vol. 229, 2010, pp. 3415-3427.

[2] M.A. Celia, T.F. Russell, I. Herrera, and R.E. Ewing, “An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation”, Adv. Water Resour., vol. 13, 1990, pp. 187-206.

[3] Y. Cheng, D. Liang, W. Wang, S. Gong, and M. Xue, “An Efficient Approach of Aerosol Thermodynamic Equilibrium Predictions by the HDMR Method”, Atmos. Environ., vol. 44, 2010, pp. 1321-1330.

[4] J. Douglas Jr. and T.F. Russell, “Numerical methods for convection dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures”, SIAM Journal on Numerical Analysis, vol. 19, 1982, pp. 871-885.

[5] G.A. Grell, et al., “Fully coupled ‘online’ chemistry within the WRF model”, Atmos. Environ., vol. 39, 2005, pp. 6957-6975.

[6] M.E. Gustafsson and L.G. Frantzrn, “Dry deposition and concentration of marine aerosols in a coastal area, SW Sweden”, Atmospheric Environment, vol. 30, 1996, pp. 977-989.

[7] C. Fountoukis and A. Nenes, “ISORROPIA II: a computationally efficient thermodynamic equilibrium model for K^{+}-Ca^{2+}-Mg^{2+}-NH_4^{+}-Na^{+}-SO_4^{2-}-NO_3^{-}-Cl^{-}-H_2O aerosols”, Atmos. Chem. Phys., vol. 7, 2007, pp. 4639-4659, doi:10.5194/acp-7-4639-2007.

[8] K. Fu, D. Liang, W. Wang, Y. Cheng, and S. Gong, “Multi-component atmospheric aerosols prediction by a multi-functional MC-HDMR approach”, Atmospheric Research, vol. 113, 2012, pp. 43-56.

[9] D. Liang, C. Du and H. Wang, “A fractional step ELLAM approach to high-dimensional convection-diffusion problems with forward particle tracking”, Journal of Computational Physics, vol. 221, 2007, pp. 198-225.

[10] D. Liang, Q. Guo, and S. Gong, “A new splitting wavelet method for solving the general aerosol dynamics equations”, Journal of Aerosol Science, vol. 39, 2008, pp. 467-487.

[11] U. Nopmongcol, et al., “Modeling Europe with CAMx for the Air Quality Model Evaluation International Initiative (AQMEII)”, Atmospheric Environment, vol. 53, 2012, pp. 177-185.

[12] M.T. Odman and A.G. Russell, “A multiscale finite element pollutant transport scheme for urban and regional modeling”, Atmospheric Environment, vol. 25A, 1991, pp. 2385-2394.

[13] W.C. Skamarock, A description of the Advanced Research WRF version 3, NCAR Tech. Note NCAR/TN-475+STR, 2008.

[14] US Environmental Protection Agency, 2010, Technical Support Document: Preparation of Emissions Inventories for the Version 4, 2005-based Platform, Office of Air Quality Planning and Standards, Air Quality Assessment Division.

Reusable Modeling of Diagnosis Functions for Embedded Systems

Shingo Nakano, Tatsuya Shibuta, Masatoshi Arai, Noriko Matsumoto, Norihiko Yoshida
Graduate School of Science and Engineering
Saitama University
Saitama, Japan
Emails: {nakano, shibuta, arai, noriko, yoshida}@ss.ics.saitama-u.ac.jp

***Abstract*—This paper presents a technique to embed diagnosis functions in model-based design of embedded systems, allowing designers to do this at early design stages, before separation of hardware and software (HW/SW) implementations and derivation of several variations. First, we develop some simple monitoring functions suitable for both HW/SW based on JTAG (Joint Test Action Group). Then, in order to identify faulty components in a complex embedded systems where a fault in a component can affect others, we employ a method proposed by Kutsuna et al., which is based on "model-based diagnosis" studied in the field of Artificial Intelligence. This paper uses MATLAB/Simulink as a modeling framework, and employs aspect-oriented approach for diagnosis description to promote reuse of diagnosis function models. This method enables to locate a fault source even if the fault propagates multiple modules.**

***Keywords*—Embedded System Diagnoses; Aspect-Oriented Systems**

I. Introduction

Recently, embedded systems are getting large, complex, distributed, and are composed of many components both in HW/SW. When a system fails, specifying the source is difficult because it often involves a number of components. Therefore, by implementing a function of detecting failure and specifying the source in the system, one can reduce cost and increase running rate of the system by shortening the average time to repair.

There exist various techniques based on studies of diagnosis of embedded systems, such as abstract model-based diagnosis [1], modelling and verification [2], and health assessment [3]. Additionally, we can effectively develop embedded systems by adding diagnosis functions at design modeling stages in accordance with model-based design, since we can verify the model of the system, which includes these functions, and generate source code from these models [4].

However, these techniques [1][2][3][4] have been studied individually, so it is still not clear how they will be applied in actual development of embedded systems.

This paper proposes a technique to implement diagnosis functions that detect failed components, and embed them at modeling stage. We aim to include diagnosis functionality into embedded systems in a way that allows implementing HW easily and with less resources. In this research, we use MATLAB/Simulink [5] for modeling.

First, we describe a function getting Input and Output (I/O) of component in the modeling stage of implementing the HW/SW (referring to a method proposed by Irizuki et al. [6] for monitoring the I/O) in order to get I/O needed for the diagnosis. Then, we model and implement a diagnosis method proposed by Kutsuna et al. [1] to locate the fault source.

Embedding of the diagnosis functions occurs at the modeling stage using aspect-oriented programming. By using aspects, we can easily add modularized functions, and it is possible to handle variations as well as make design more effective.

The structure of the paper is as follows. In Section II, we propose how to obtain the I/O of components and create a model using MATLAB/Simulink. In Section III, we summarize the diagnosis methods proposed in [1] and explain our method. In Section IV, we describe how to apply diagnosis functions to the target using aspects, since it enables function parts to be reused. In Section V, we simulate proposed diagnosis functions using a simple model to verify feasibility of this research. Finally, in Section VI, we describe the conclusion and outline future work of this study.

II. Obtaining Input and Output of a component

A. Idea

Our idea is that diagnosis requires Input and Output of a component in an embedded system. In this research, we use JTAG (Joint Test Action Group) [7], which is the standard test access port and boundary scan architecture of IC chips to obtain Input and Output.

B. JTAG

Irizuki et al. propose a method which monitors Inputs and Outputs of an embedded system using JTAG to prevent malfunction [6]. In JTAG, it is possible to I/O cells corresponding to the respective I/O pins of the IC chip from outside. The controller for JTAG test is standardized as TAP (Test Access Port) Controller and has at least four serial interfaces: TCK (Test Clock), TMS (Test Mode Select), TDI (Test Data In) and TDO (Test Data Out).

Fig. 1 shows the structure of the IC chip according to JTAG. Cells are placed between I/O pins of the IC and the internal logic, and store the values of the I/O. Since cells are implemented as cascading shift registers, it is possible to input from TDI and output to TDO.

C. Abstract JTAG

JTAG is implemented in HW and does not involve complex computations, therefore it is also suitable for embedded systems with limited resources and functionality. However, because of that, it is only possible to monitor Inputs and

Outputs in HW. In this research, we propose getting Inputs and Outputs needed for diagnosis at the modeling stage before separation of HW/SW, since diagnosis is HW/SW-independent. For this purpose, we make JTAG more abstract in the following way.

- Inputs and Outputs go directly into an IC without using cells.
- There is no need for instructions and instruction register.
- The value of TMS is used for control, instead of TAP controller.
- We obtain the whole value of the Input and Output rather than obtaining it bit by bit.
- The value of TDI is shifted only after the shift in values of the Input and Output.

D. Getting Inputs and Outputs with abstract JTAG

Abstract JTAG obtains Inputs and Outputs switching the following two states of the value of TMS.

TMS=0: Inputs and Outputs of diagnosis target are stored in cells.

TMS=1: Stored values are shifted.

Then, we get Inputs and Outputs using above states as follows.

Step 1 Input and Output values are stored in cells (TMS=0).
Step 2 Saved values are shifted to TDO (TMS=1).
Step 3 Repeat Step 2 times the number of Inputs and Outputs.

E. Modeling in MATLAB/Simulink

In this research, we make JTAG more abstract and model it in MATLAB/Simulink, which is widely used in embedded system design.

If HW implementation is done according to this model, it becomes identical or similar to JTAG. On the other hand, if SW implementation is done according to this model, there is no need for dynamic memory assignment, and control is performed with fixed amount of memory (determined by the number of Inputs and Outputs) using assignment operations and control statements.

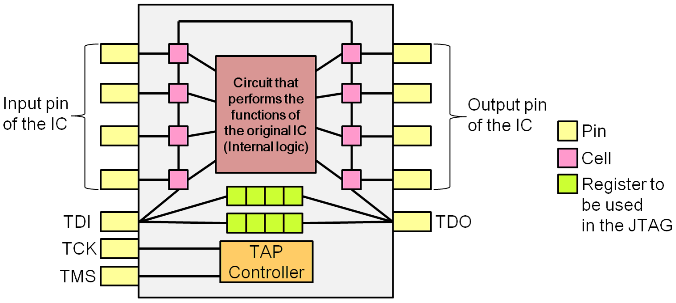

Input pin of the IC
Circuit that performs the functions of the original IC (Internal logic)
Output pin of the IC
TDI
TCK
TMS
TAP Controller
TDO
Pin
Cell
Register to be used in the JTAG

Fig. 1. IC chip construction based on JTAG.

III. Locating faults by model-based diagnosis

A. Idea

If one component fails and outputs abnormal value in embedded systems, from the outside it may look like multiple components fail, since components that receive the output also output abnormal values. This problem is known as fault propagation problem, which makes it difficult to identify faulty component in large-scale systems.

In this research, we use abstract model-based diagnosis [1] to identify the faulty component.

B. Abstract model-based diagnosis

Model-based diagnosis is a framework for system diagnosis that defines the behavior of each component and determines whether components are normal or not using logical relations derived from the structure of the system, and observations of data flows through the system [8][9]. For this purpose, it uses statements SD (System Description), OBS (Observations) and DIAG (Diagnosis).

SD Statement indicating the logical relations derived from the structure of the system.
OBS Statement which represents observations of data flows through the system.
DIAG Statement which represents whether each component is normal or not.

Model-based diagnosis is usually necessary to describe the behavior of components. However, writing down exact behavior of the software is difficult. To solve this problem, abstract model-based diagnosis has been proposed [1]. There, the logical relations derived from the configuration of the system are acquired by using the formula ”outputs are normal if components and inputs are normal”, so it becomes unnecessary to write down the behavior of the components.

As an example, let us show the steps of the abstract model-based diagnosis using an abstract model shown in Fig. 2 which is used as example in [1]. First, SD is defined as follows.

$$\begin{aligned}\mathbf{SD} \equiv \{ok(C_1) \wedge ok(a) \wedge ok(b) \rightarrow ok(c) \wedge ok(d)\} \\ \wedge\{ok(C_2) \wedge ok(c) \rightarrow ok(e)\} \\ \wedge\{ok(C_3) \wedge ok(d) \rightarrow ok(f)\} \qquad (1)\end{aligned}$$

The first line of the equation (1) shows that outputs c, d are normal if C_1 is normal and inputs a, b are normal.

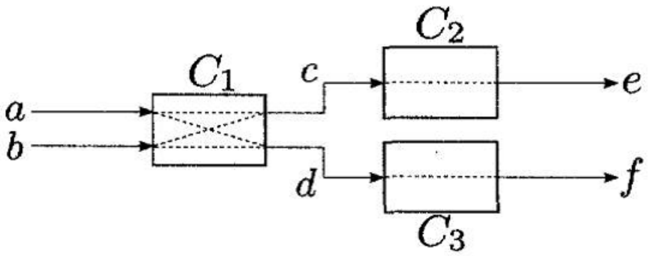


Fig. 2. Example of an abstract model.

In abstract model-based diagnosis, OBS is defined as a result of checking whether each data is normal or not according to some criteria. For example, if c, e are abnormal and other data are normal in Fig. 2, OBS is defined as follows.

$$\begin{aligned} \mathbf{OBS} \equiv & \, ok(a) \wedge ok(b) \wedge \neg ok(c) \\ & \wedge ok(d) \wedge \neg ok(e) \wedge ok(f) \end{aligned} \tag{2}$$

As for the method for determining whether data is normal or not, the following two are considered.

- Methods based on designers knowledge, such as specifying range or period within which the data must be generated.
- Methods that are not based on knowledge, such as using statistical learning or data mining on accumulated data, or building separate model for the interior of the component and using formal methods like model checking.

DIAG shows whether each component is normal or not, and in case component C_1, C_3 are abnormal and component C_2 is normal, DIAG is written as follows.

$$\mathbf{DIAG} \equiv \neg ok(C_1) \wedge ok(C_2) \wedge \neg ok(C_3) \tag{3}$$

$\neg ok(C)$ means that the component C is abnormal. Therefore we will write DIAG as a list of abnormal components in parentheses. For example, DIAG of the equation (3) is represented as $\{C_1, C_3\}$.

Model-based diagnosis seeks appropriate DIAG from SD and OBS. That is, it seeks DIAG based on the condition ”given what DIAG, conflict does not occur between SD and OBS”, which is represented by the following equation.

$$\mathbf{SD} \wedge \mathbf{OBS} \wedge \mathbf{DIAG} = \mathbf{True} \tag{4}$$

When SD is given by the equation (1) and OBS is given by the equation (2), possible values of DIAG are $\{C_1\}$,$\{C_1, C_2\}$,$\{C_1, C_3\}$, $\{C_1, C_2, C_3\}$. Among all the possible DIAGs, minimal diagnosis is defined as the one which contains minimal number of components that does not make (4) contradicting. In the example above, $\{C_1\}$ is the minimal diagnosis. Minimal diagnosis may also contain two or more faulty components, which means that it is possible to identify multiple faults.

As shown in the example above, in abstract model-based diagnosis, sometimes multiple DIAGs are obtained for given OBS and SD. In this case, we consider the probability of multiple abnormal components at the same time to be small, and abstract model-based diagnosis will output the DIAG which is minimal number of abnormal components as the diagnosis result. Thus, in example above, abstract model-based diagnosis outputs a result showing that C_1 is abnormal.

C. *Modeling in MATLAB/Simulink*

We propose the way to model diagnosis systems using abstract model-based diagnosis (in particular, dealing with OBS) similar to modeling the acquisition of Inputs and Outputs in Section II, so that diagnosis is implemented independent from HW or SW.

1) Placement of the diagnosis system: For example, we place the diagnosis system for target model in Fig. 2 as shown in Fig. 3. Here, Diagnosis Component (DC) stands for the component that receives Inputs and Outputs of each component from abstract JTAG (introduced in Section II) to determine normality, and DCC (Diagnosis Component Center) stands for the component that outputs OBS based on results from DCs.

2) Criteria of data normality: As mentioned in Section III-B, in the abstract model-based diagnosis, there are two types of methods for determining data normality. In this study, we use the method based on designers knowledge with the following two criteria.

1. Combination of Inputs and Outputs is determined uniquely.
2. Inputs and Outputs must fall within a certain range.

There is no established format describing information for deciding whether Inputs and Outputs of each component are normal or not in [1], so in this research, we use a matrix called normal data matrix.

The normal data matrix is defined, as shown in Fig. 4. We write the numbers of corresponding components to column C, each input condition to columns of inputs (1,$\cdots$, m), each output condition to columns of outputs (1,$\cdots$, n), and last two columns contain position of the boundary between the Input and Output data and the number of criteria.

DC determines whether Inputs and Outputs are normal or not by corresponding criteria as following.

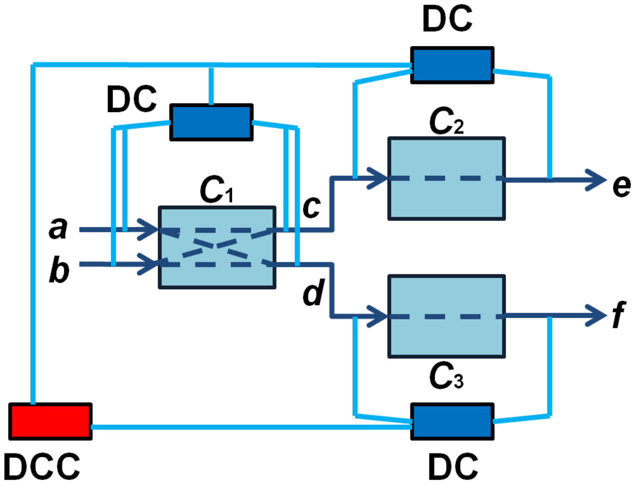


Fig. 3. Placement of diagnosis system.

ISBN: 978-1-61208-354-4

$$\begin{array}{ccccccccc} C & \mathit{input}\ 1 & \cdots & \mathit{input}\ m & \mathit{output}\ 1 & \cdots & \mathit{output}\ n & \mathit{boundary\ between\ input\ and\ output} & \mathit{criteria} \\ x & a_1 & \cdots & a_m & b_1 & \cdots & b_n & m+1 & y \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x & c_1 & \cdots & c_m & d_1 & \cdots & d_n & m+1 & y \end{array}$$

Fig. 4. Normal data matrix notation.

Criteria 1:

Step 1 Compose normal data matrix that contains combination of the normal I/O of the diagnosis target in each line.
Step 2 Given the observation values of I/O and values of I/O values in normal data matrix, we choose the ones that contain more identical values.
Step 3 Given the observation values and selected combination of I/O, we consider the same I/O as normal and different ones as abnormal.

Criteria 2:

Step 1 Compose normal data matrix that contains the smallest values of I/O of diagnosis target in a first line, and the largest values in a second line.
Step 2 Define I/O that are within the range as normal and others as abnormal.

3) Example of normal data matrix: As an example, here we describe normal data matrix using the model from Fig. 3.

First, we assume that normality for Inputs and Outputs of each component is determined as follows using two criteria mentioned above.

C_1: a is normal if $-1 \leq a \leq 1$. (Criteria 2)
C_1: b is normal if $-2 \leq b \leq 2$. (Criteria 2)
C_1: c is normal if $0 \leq c \leq 1$. (Criteria 2)
C_1: d is normal if $0 \leq d \leq 1$. (Criteria 2)
C_2: c,e are normal if value (c,e) is any of the following: (0,0), (0,1), (1,0), (1,1). (Criteria 1)
C_3: d,f are normal if value (d,f) is any of the following: (0,−1), (0,1), (1,1), (1,−1). (Criteria 1)

Normal data matrices of C_1, C_2, C_3 are described as follows.

Normal data matrix of C_1:

$$\begin{pmatrix} 1 & -1 & -2 & 0 & 0 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 3 & 2 \end{pmatrix}$$

Normal data matrix of C_2:

$$\begin{pmatrix} 2 & 0 & 0 & 2 & 1 \\ 2 & 0 & 1 & 2 & 1 \\ 2 & 1 & 0 & 2 & 1 \\ 2 & 1 & 1 & 2 & 1 \end{pmatrix}$$

Normal data matrix of C_3:

$$\begin{pmatrix} 3 & 0 & -1 & 2 & 1 \\ 3 & 0 & 1 & 2 & 1 \\ 3 & 1 & -1 & 2 & 1 \\ 3 & 1 & 1 & 2 & 1 \end{pmatrix}$$

4) Rapid increase in the size of normal data matrix: There is a problem that normal data matrix size increases rapidly with the number of I/O of a component and the number of possible value combinations according to criteria 1. To address this problem, we propose either creating a program describing the combination of the normal I/O or using criteria 2. In order to deal with situations where such approach is not possible, it would be necessary to think of other criteria of data normality.

5) Procedures of DC and DCC: Procedures of DC and DCC are described using normal data matrix as follows.

Procedure of DC:

Step 1 Obtain observed value of the I/O of the diagnosis target from abstract JTAG.
Step 2 Get normal data matrix and choose normality criteria.
Step 3 According to normality criteria, determine normality.
Step 4 Report each Input and Output as normal or not to DCC.

Procedure of DCC:

Step 1 Get the information reported from the all DCs.
Step 2 Output OBS from reported information.

D. Modeling in MATLAB/Simulink

User-defined function block enables modeling diagnosis functions in MATLAB/Simulink. Since users can write the process of the block as a program, diagnosis system is modeled by describing the process of DC and DCC like that.

IV. Modularize and reuse functions using aspects

A. Idea

In this study, we propose embedding the diagnosis functions by using aspects and model transformation, so that we are able to reuse them. Because aspect can be applied after the completion of the target model, changing the model and adding functions can be done easily. Also, it allows variations, such as implementation with just diagnosis functions removed.

In this study, we use model transformation with aspects not as specific means for embedding the diagnosis functions, but in a way that enables to use it in the general model.

B. Model transformation with aspects

Aspect is a technique that is designed to extract and modularize process that is common in many models or programs. Such process is called Crosscutting Concern [10], and while it is difficult to modularize it only with modularization, aspects perform well in this case.

Join point model, which is a representative model of the aspect, consists of join point, pointcut and advice. Join point is a ”point in code” to which aspect can be applied, and a set of more than one join points is called pointcut. Advice is a set of instructions that indicate what should be handled by aspect code. Aspect can be applied to the target program at compile time or at a run time. Action that adds the functionality defined by advice to a join point specified by a pointcut is called weave, and the utility which performs weave is called weaver.

Models created with MATLAB/Simulink are stored as a code on a layer-structured document. Therefore, in this study, we perform model transformation by changing the contents of the code with aspects. That is, we indicate certain location in code as a pointcut, and change internal structure of model by changing the contents of the code with advice (Fig. 5). For embedding the diagnosis functions modeled by MATLAB/Simulink, we apply aspect that adds blocks, lines and branches of line needed for diagnosis. However, since weaver that can be used in aspect applying is currently incomplete, we apply aspects by hand.

Since the amount of components and the amount and names of each component’s I/O are different depending on diagnosis targets, it is difficult to design common embedding diagnosis functionality. Therefore, we divide diagnosis functionality according to target and common parts, and modularize common parts together. This way we can design efficiently only by changing the point of embedding common part of diagnosis functionality.

In this study we use MATLAB/Simulink for describing model in XML, since we describe aspect using XML format also. The outline of description is shown in Fig. 6. Here, the entire aspect is contained in ⟨aspect⟩ tag, description of pointcut is contained in ⟨pointcut⟩ tag, and advice is contained in ⟨advice⟩ tag. When adding block and line, join point is represented by target system, and for the branches of line, join point is represented by a target line. The action in advice is selected according to type. In this study, the type can be block-add, line-add and branch-add.

V. Experiments

A. Details of the experiment

We verify feasibility of proposed method using simple test model. As test model, we use Wave.mdl [11] shown in Fig. 7.

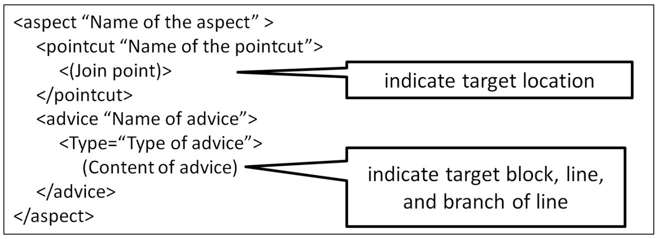
Here, Twice_Component has input of sine wave **a** with amplitude of 1. We regard this component as C_1. C_1 outputs sine wave **b** with amplitude twice as large as **a**. Add_Component has inputs of sine wave **c** with amplitude of 1 and sine wave **b** which is the output of C_1. We regard this component as C_2. C_2 outputs **d** which is the sum of sine waves **b** and **c**. We set time step for performing Input and Output to 0.1, and perform simulation from 0.0 till 6.0.

We embed diagnosis functions to Wave.mdl, and we identify the component that is causing error by obtaining I/O of C_1 and C_2 and detecting error for each component.

We made Wave_DCC_JTAG.mdl shown in Fig. 8, which models the embedding diagnosis functionality. Diagnosis modules may look complicated in the figure, but actually they have a simple structure and does not require much resources to implement.

Criteria for deciding whether Inputs and Outputs of each component are normal or not are as follows.

C_1: a is normal if $-1 \le \mathbf{a} \le 1$. (Criteria 2)
C_1: b is normal if $-2 \le \mathbf{b} \le 2$. (Criteria 2)
C_2: b is normal if $-2 \le \mathbf{b} \le 2$. (Criteria 2)
C_2: c is normal if $-1 \le \mathbf{c} \le 1$. (Criteria 2)
C_2: d is normal if $-3 \le \mathbf{d} \le 3$. (Criteria 2)



```
<aspect “Name of the aspect” >
   <pointcut “Name of the pointcut”>
      <(Join point)>
   </pointcut>
   <advice “Name of advice”>
      <Type=“Type of advice”>
         (Content of advice)
   </advice>
</aspect>
```

Fig. 6. Outline of aspect description.

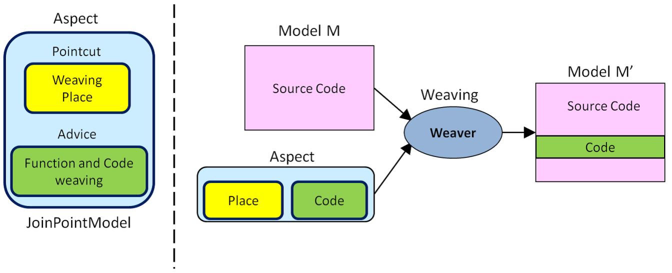


Fig. 5. Model transformation with aspects.

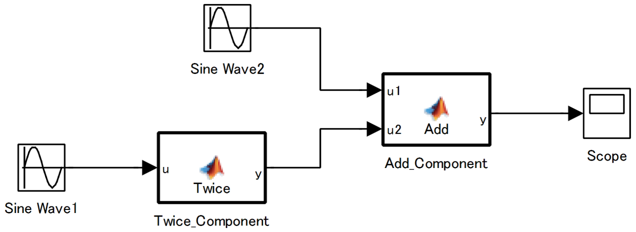


Fig. 7. Test model: Wave.mdl.

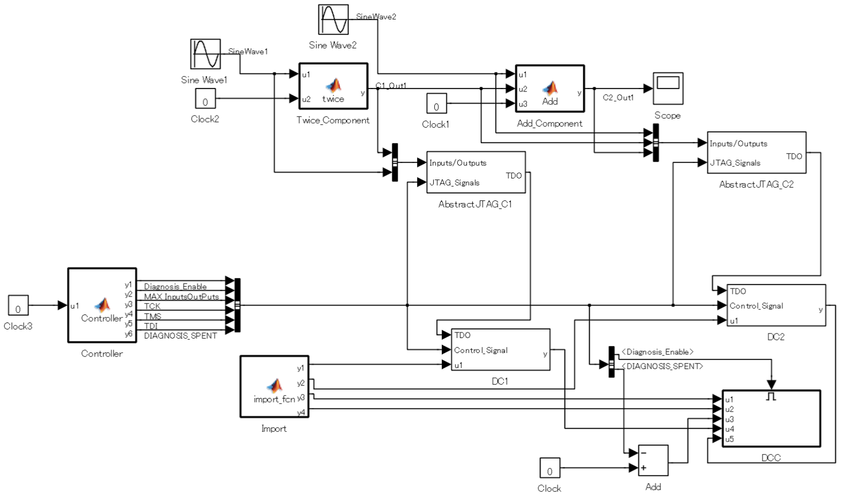


Fig. 8. Model: Wave_DCC_JTAG.mdl.

TABLE I. THE DETAILS OF CAUSING ERRORS

Time	Position	Error details
1.0	C_1	Multiplies input **a** by 5.
3.0	C_2	Multiplies input **b** by -1 before addition.
$4.0 \sim 5.0$	C_2	Multiplies input **c** by 3.
5.0	C_1	Multiplies input **a** by -4.

TABLE II. DIAGNOSIS RESULT

Time	Input and Output	Component
0.0	All normal	All normal
1.0	Output **b** in C_1, input **b** and output **d** in C_2 are abnormal	C_1 is abnormal
2.0	All normal	All normal
3.0	All normal	All normal
4.0	Output **d** in C_2 is abnormal	C_2 is abnormal
5.0	Output **b** in C_1, input **b** and output **d** in C_2 are abnormal	C_1 is abnormal

The diagnosis is performed at time steps 0.0, 1.0, 2.0, 3.0, 4.0 and 5.0. In addition, we raise errors shown in Table I and verify whether diagnosis function can detect them.

B. *Experiment results*

Experiment results are shown in Table II. The results at 0.0, 1.0, 2.0 and 4.0 are all successful. However, the results at 3.0 and 5.0 show abnormalities. First, at 3.0, output **d** is supposed to become abnormal because we introduced an error to C_2, but the error was not detected, since it did not exceed the range of normal values that were used in the criteria. When using criteria that check if Input and Output fall within a certain range, we can see that detection is not possible if the error does not make Input and Output exceed the range. In addition, at 5.0, we caused an error in both the C_1 and C_2, but only C_1 was identified as abnormal component. The reason for this is that model-based diagnosis outputs minimal diagnosis. That is, if several components become abnormal at the same time, only the component that is the starting point would be identified as abnormal.

VI. CONCLUSION AND FUTURE WORK

In this paper, we described abstract JTAG which monitors I/O of component, and abstract model-based diagnosis which identifies faulty component at modeling stage. We also proposed model transformation using aspects for modularizing and reuse of those functions. This method enables to locate a fault source even if the fault propagates multiple modules.

In the future, we will confirm the utility of proposed method by actually generating HW and SW from MATLAB/Simulink model and evaluating overhead and increase in code size that is caused by embedding diagnosis functions. In addition, since we currently calculate DIAG manually, we need to automate it as well.

REFERENCES

[1] T. Kutsuna, S. Sato, and N. Chyujo, “Fault Location in Collaborative Systems Using Abstract Model Based Diagnosis”, Workshop on embedded technology and network (ETNET2009), 2009, pp. 43–48.

[2] Z. Simeu-Abazi, M. Di Mascolo, and M. Knotek, “Fault diagnosis for discrete event systems: Modelling and verification”, Reliability Engineering & System Safety, vol. 95, no. 4, 2010, pp. 369-378.

[3] M. Dievart, P. Charbonnaud, and X. Desforges, “An embedded distributed tool for transportation systems health assessment”, Embedded Real Time Software and Systems (ERTS2 2010), 2010, pp. 1–10.

[4] P. F. Smith, S. M. Prabhu, and J. Friedman, “Best Practices for Establishing a Model-Based Design Culture”, SAE 2007 World Congress, 2007, pp. 1–7.

[5] MATLAB/Simulink, http://www.mathworks.com/products/simulink/ [accessed: 2014-07-08].

[6] Y. Irizuki, M. Ohara, and K. Sakamaki, “An Online Self-Monitoring Approach for Embedded Systems Using JTAG Interface”, The journal of Reliability Engineering Association of Japan, vol. 32, no. 3, 2010, pp. 185–190.

[7] IEEE Standard Test Access Port and Boundary-Scan Architecture-Description, 1990.

[8] J. De Kleer and J. Kurien, “Fundamentals of model-based diagnosis”, Proceedings of IFAC Safeprocess 3, 2004, pp. 25–36.

[9] J. De Kleer and B. C. Williams, “Diagnosing multiple faults”, Artificial Intelligence, 32 (1), 1987, pp. 97–130.

[10] R. Laddad, “AspectJ in Action”, Manning, 2003.

[11] mdl, http://www.mathworks.com/help/simulink/ug/saving-a-model.html [accessed: 2014-07-08].

Kinematically Exact Beam Finite Elements Based on Quaternion Algebra

Eva Zupan
Slovenian National Building
and Civil Engineering Institute
Ljubljana, Slovenia
Email: eva.zupan@zag.si

Miran Saje and Dejan Zupan
University of Ljubljana
Faculty of Civil and Geodetic Engineering
Ljubljana, Slovenia
Email: dejan.zupan@fgg.uni-lj.si

***Abstract*—Rotations in three-dimensional Euclidean space can be represented by the use of quaternions numerically efficient and robust. In the present approach, rotational quaternions are used as the primary quantities to describe the rotational degrees of freedom for static and dynamic analysis of geometrically non-linear beam-like structures. The classical concept of parametrization of the rotation matrix by the rotational vector is thus completely abandoned. The consistent governing equations of the beam in terms of the quaternion algebra are presented and several quaternion-based numerical implementations are discussed.**

Keywords–beam theory; non-linear geometry; quaternion algebra; finite-element formulation, statics; dynamics

I. Introduction

Beams are important load-carrying members of various engineering structures. A common characteristic of these structural elements is that one dimension is considerably longer than the other two, which allows us to employ relatively simple mathematical models to describe their geometry. Nevertheless, modern demands for such structures impose the need to predict accurately and efficiently large displacements and rotations and finite strains that can occur during the deformation. Therefore, the governing equations of the problem in general become non-linear and too demanding to be solved analytically.

Efficient numerical implementation of three-dimensional non-linear beam elements is still a challenge for researches. Most of the problems in non-linear beam formulations reported in literature stem from the properties of three-dimensional rotations that are directly or indirectly incorporated into the methods. The non-linearity of three-dimensional rotations requires a special treatment in parametrization, discretization, interpolation and iterative update. It is crucial for the overall efficiency of the finite element formulation that in all these procedures a sufficient attention is paid to the properties of rotations.

Among various existing non-linear beam theories we here study the 'geometrically exact beam theory' also denoted as 'Cosserat theory of rods' as introduced by Antman [1], Reissner [2], and Simo [3]. We will especially focus on the treatment of rotations in numerical implementation. There is a number of possible ways of choosing the suitable representation or parametrization of rotations. Widely used are the three-component parametrizations as we directly avoid any algebraic constraints. Among such parametrizations a 'rotational vector' [4] seems to be very popular. Simo [3], Bottasso and Borri [5], Jelenić and Crisfield [6] and many others employed the rotational vector as the members of the primary unknowns. In contrast to these models, Betsch and Steinmann [7] used the director triad with constraints. By the use of directors several disadvantages of the rotational vector, such as singularity and strain non-objectivity, are avoided with the additional cost of six constraint equations that need to be considered at each node. An interesting alternative is the algebra of quaternions that was only recently recognized as a suitable tool in three-dimensional beam formulations [8]–[12].

Quaternions are elements of the four-dimensional Euclidean space. By introducing an additional operation called 'quaternion multiplication' we are able to represent rotations with quaternions. Only one additional degree of freedom in such parametrization is sufficient to avoid singularities while it introduces only one algebraic constraint that needs to be satisfied. The equations of the beam need to be properly transformed and all the steps in the numerical procedure need to be taken in accord with the new configuration space. In this work, we therefore present the derivation of the dynamic governing equations of the three-dimensional beam in terms of quaternions using an energy-consistent approach and discuss the numerical implementation in terms of quaternion algebra for statics and dynamics.

This paper is structured in the following manner. Section 2 introduces quaternion algebra and its properties. In Section 3, we describe the mathematical model of the three-dimensional beam. Kinematic equations are reviewed in Section 4. Section 5 introduces the continuous governing equations in terms of quaternion algebra, while, in Sections 6 and 7, the finite-element implementations for static and dynamic problems are described, respectively, and some numerical examples are given. The paper ends with concluding remarks.

II. Quaternions

The set of quaternions $I\!H$ is a four-dimensional Euclidean linear space. Its elements are often presented as the sum of a scalar and a vector, i.e., $\widehat{x} = s + \overrightarrow{v} = \left(s, \overrightarrow{v}\right)$, $s \in I\!R$, $\overrightarrow{v} \in I\!R^3$. Addition and scalar multiplication are inherited from $I\!R^4$. We additionally introduce the *quaternion multiplication*

$$\widehat{x} \circ \widehat{y} = \left(s\,c - \overrightarrow{v} \cdot \overrightarrow{w}\right) + \left(c\overrightarrow{v} + s\overrightarrow{w} + \overrightarrow{v} \times \overrightarrow{w}\right), \qquad (1)$$

where $\widehat{y} = c + \overrightarrow{w} \in I\!H$. Here $(\cdot)$ denotes the scalar product and $(\times)$ denotes the cross-vector product in $I\!R^3$.

Any quaternion q, with unit norm ($|\widehat{q}| = 1$) can be expressed in polar form as

$$\widehat{q} = \cos\frac{\vartheta}{2} + \sin\frac{\vartheta}{2}\,\vec{n}, \qquad \left|\vec{n}\right| = 1, \tag{2}$$

where ϑ is the angle of rotation and $\vec{n}$ is the unit vector on the axis of rotation. That is why we also call them *rotational quaternions*. Let $\vec{b} \in \mathbb{R}^3$ denote a three-dimensional vector obtained by rotating vector $\vec{a} \in \mathbb{R}^3$ by an angle ϑ about an axis, defined by unit vector $\vec{n}$. Then a relationship between the two vectors can be expressed as

$$\vec{b} = \widehat{q} \circ \vec{a} \circ \widehat{q}^{*}, \tag{3}$$

where $\widehat{q}^{*} = \cos\frac{\vartheta}{2} - \sin\frac{\vartheta}{2}\,\vec{n}$ is the conjugated quaternion.

III. Model of a three-dimensional beam

A three-dimensional beam, Fig. 1, is described by the family of position vectors $\vec{r}\,(x,t)$, $x \in [0, L]$, of the line of centroids and local orthonormal bases $\left\{\vec{G}_1\,(x,t)\,, \vec{G}_2\,(x,t)\,, \vec{G}_3\,(x,t)\right\}$ describing the inclination of cross-sections, which are assumed to preserve their shape and area during the deformation.

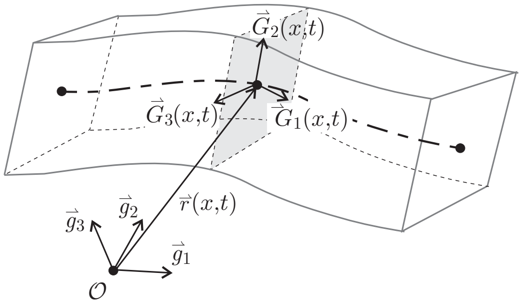


Figure 1. A three-dimensional beam.

After introducing a global orthonormal basis $\left\{\vec{g}_1, \vec{g}_2, \vec{g}_3\right\}$ each local basis can be defined by the rotation of the global one. In terms of the quaternion algebra, the relation between the moving and the fixed basis can be written as

$$\vec{G}_i\,(x,t) = \widehat{q}\,(x,t) \circ \vec{g}_i \circ \widehat{q}^{*}\,(x,t)\,, \qquad i = 1,2,3. \tag{4}$$

For computational purposes, vectors will be expressed with respect to either of the two bases and the component description will be denoted by bold face symbols. For quaternions a trivial extension of these two bases into quaternion space together with the fourth base vector - the identity element $\widehat{1} = 1 + \vec{0}$ will be used. An arbitrary quaternion, $\widehat{x} = s + \vec{v}$, can thus be expressed as

$$\widehat{x} = s\,\widehat{1} + v_1\vec{g}_1 + v_2\vec{g}_2 + v_3\vec{g}_3 = S\,\widehat{1} + V_1\vec{G}_1 + V_2\vec{G}_2 + V_3\vec{G}_3$$

and the components are represented by one-column matrices $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{X}}$, respectively. The relationship between the two representations of any quaternion is given by

$$\widehat{\mathbf{X}} = \widehat{\mathbf{q}}^{*} \circ \widehat{\mathbf{x}} \circ \widehat{\mathbf{q}}, \qquad \widehat{\mathbf{x}} = \widehat{\mathbf{q}} \circ \widehat{\mathbf{X}} \circ \widehat{\mathbf{q}}^{*}. \tag{5}$$

We will identify vectors and quaternions with zero scalar part since any vector in $\mathbb{R}^3$ can be treated as an element of the four-dimensional Euclidean space:

$$\mathbf{v} \equiv \widehat{\mathbf{v}} = \left[\begin{array}{cccc} 0 & v_1 & v_2 & v_3 \end{array}\right]^T.$$

The hat over the symbol will be omitted when the first component equals zero since the appropriate size of one-column representation will always be evident from the context.

IV. Kinematic equations

In Reissner-Simo beam theory [2] – [3], the resultant strain measures at the centroid of each cross-section are directly introduced and expressed with kinematic variables by the first order differential equations. For describing the rate of change of the position vector we introduce the translational strain

$$\boldsymbol{\gamma} = \mathbf{r}' - \boldsymbol{\gamma}_0, \qquad \boldsymbol{\Gamma} = \widehat{\mathbf{q}}^{*} \circ \widehat{\mathbf{r}}' \circ \widehat{\mathbf{q}} + \boldsymbol{\Gamma}_0, \tag{6}$$

where $\boldsymbol{\gamma}_0$ and $\boldsymbol{\Gamma}_0$ are variational constants, determined from the initial configuration of the beam. The differentiation of equation (4) with respect to parameter x results in *rotational strain* vector. In terms of quaternions, the rotational strain, also called the curvature, is determined by

$$\boldsymbol{\kappa} = 2\widehat{\mathbf{q}}' \circ \widehat{\mathbf{q}}^{*}, \qquad \boldsymbol{K} = 2\widehat{\mathbf{q}}^{*} \circ \widehat{\mathbf{q}}'. \tag{7}$$

Analogously we have

$$\mathbf{v} = \dot{\mathbf{r}}, \qquad \mathbf{V} = \widehat{\mathbf{q}}^{*} \circ \dot{\mathbf{r}} \circ \widehat{\mathbf{q}}. \tag{8}$$

describing the velocity in both descriptions. The time-dependent analogy to curvature is the *angular velocity* vector:

$$\boldsymbol{\omega} = 2\dot{\widehat{\mathbf{q}}} \circ \widehat{\mathbf{q}}^{*}, \qquad \boldsymbol{\Omega} = 2\widehat{\mathbf{q}}^{*} \circ \dot{\widehat{\mathbf{q}}}. \tag{9}$$

Again, $\boldsymbol{\omega}$ denotes the angular velocity with respect to the fixed basis while $\boldsymbol{\Omega}$ is its local basis representation.

The weak or linearized form of kinematic equations (6) and (7) relating the variations of strains, displacements and rotational quaternions will also be needed. They can be derived by direct linearization of the strong form of kinematic equations, which leads to

$$\delta_{\mathrm{rel}}\boldsymbol{\Gamma} = (\widehat{\mathbf{q}}^{*} \circ \delta\widehat{\mathbf{r}}' \circ \widehat{\mathbf{q}}) + \; 2\widehat{\mathbf{q}}^{*} \circ (\widehat{\mathbf{r}}' \times (\delta\widehat{\mathbf{q}} \circ \widehat{\mathbf{q}}^{*})) \circ \widehat{\mathbf{q}} \tag{10}$$

$$\delta_{\mathrm{rel}}\boldsymbol{K} = 2\widehat{\mathbf{q}}^{*} \circ (\delta\widehat{\mathbf{q}} \circ \widehat{\mathbf{q}}^{*})' \circ \widehat{\mathbf{q}}. \tag{11}$$

Note that by (10)–(11) the relative or objective variations of strains are defined. The objective variations are variations of components with respect to the local basis where the changes of are not taken into account. It is also interesting to observe that the same equations can be obtained from the virtual work principle as presented in [13].

A result similar to (7) is obtained after we express the variation of the rotational vector $\boldsymbol{\vartheta} = \vartheta\mathbf{n}$ with the variation of rotational quaternion. After a short derivation we obtain

$$\delta\boldsymbol{\vartheta} = 2\delta\widehat{\mathbf{q}} \circ \widehat{\mathbf{q}}^{*}. \tag{12}$$

The above result is crucial for the replacement of rotational vectors with quaternions in any variational principle.

V. THE CONTINUOUS GOVERNING EQUATIONS

Our goal is to express the governing equations in terms of quaternions as the only rotational degrees of freedom. Quaternions will serve as a suitable replacement for both rotational vector and matrix. We will start from the weak form of the dynamic equilibrium of a three-dimensional beam:

$$\begin{aligned}
&\int_0^L \left[\mathbf{N}(x,t)\cdot\delta_{\mathrm{rel}}\boldsymbol{\Gamma}(x,t)+\mathbf{M}(x,t)\cdot\delta_{\mathrm{rel}}\boldsymbol{K}(x,t)\right]dx\\
&=\int_0^L\left[\tilde{\mathbf{n}}(x,t)-\rho A\ddot{\mathbf{r}}(x,t)\right]\cdot\delta\mathbf{r}(x,t)\;dx\\
&+\int_0^L\Big[\tilde{\mathbf{m}}(x,t)-\mathbf{R}(x,t)\,\mathbf{J}_\rho\dot{\boldsymbol{\Omega}}(x,t)\\
&\qquad-\boldsymbol{\omega}(x,t)\times\mathbf{R}(x,t)\,\mathbf{J}_\rho\boldsymbol{\Omega}(x,t)\Big]\cdot\delta\boldsymbol{\vartheta}(x,t)\;dx\\
&+\boldsymbol{f}^L(t)\cdot\delta\mathbf{r}(L,t)+\boldsymbol{h}^L(t)\cdot\delta\boldsymbol{\vartheta}(L,t)\\
&-\boldsymbol{f}^0(t)\cdot\delta\mathbf{r}(0,t)+\boldsymbol{h}^0(t)\cdot\delta\boldsymbol{\vartheta}(0,t)\,,
\end{aligned}\tag{13}$$

where the term on the left hand side denotes the virtual work of internal forces and the terms of the right hand side denote the virtual work of external (applied) and inertial forces and moments. $\mathbf{N}$ and $\mathbf{M}$ are the stress-resultant force and moment vectors of the cross-section; $\tilde{\mathbf{n}}$ and $\tilde{\mathbf{m}}$ are external distributed force and moment vectors per unit of the initial length; ρ denotes mass per unit of the initial volume; $\mathbf{R}$ is the rotation matrix; A is the area of the cross-section; $\mathbf{J}_\rho$ is the centroidal mass-inertia matrix of the cross-section; $\boldsymbol{f}^0$, $\boldsymbol{h}^0$, $\boldsymbol{f}^L$ and $\boldsymbol{h}^L$ are the external point forces and moments at the two boundaries, $x=0$ and $x=L$. For simplicity and clearness of the notation, the dependency on x and t will be omitted.

We will replace any rotation matrix acting on a vector by the quaternion-based rotation, see (3). Further, we will replace the variation of the three-parameter rotational vector by the variation of the four-parameter rotational quaternion using (12) and rearrange the scalar product in the following manner:

$$\mathbf{w}\cdot\delta\boldsymbol{\vartheta}=2\mathbf{w}\cdot\delta\widehat{\mathbf{q}}\circ\widehat{\mathbf{q}}^*=2\left(\mathbf{w}\circ\widehat{\mathbf{q}}\right)\cdot\delta\widehat{\mathbf{q}}\tag{14}$$

for any vector $\mathbf{w}$.

The four components of the rotational quaternion are mutually dependent due to the unit norm condition. The unit norm constraint is enforced in the model by the method of Lagrangian multipliers. The constraint ($\widehat{\mathbf{q}}\cdot\widehat{\mathbf{q}}=1$) is first multiplied by an arbitrary unknown scalar function $\lambda(x,t)$, independent on the primary unknowns, varied and integrated along the length of the beam:

$$\int_0^L 2\lambda\widehat{\mathbf{q}}\cdot\delta\widehat{\mathbf{q}}\,dx+\int_0^L\left(\widehat{\mathbf{q}}\cdot\widehat{\mathbf{q}}-1\right)\delta\lambda\,dx.\tag{15}$$

Equation (13) is rewritten in terms of quaternion algebra and (15) is added to finally get

$$\begin{aligned}
&\int_0^L\left[\mathbf{N}\cdot\delta_{\mathrm{rel}}\boldsymbol{\Gamma}+\mathbf{M}\cdot\delta_{\mathrm{rel}}\boldsymbol{K}\right]dx\\
&=\int_0^L\left[\tilde{\mathbf{n}}-\rho A_r\ddot{\mathbf{r}}\right]\cdot\delta\mathbf{r}\,dx\\
&+2\int_0^L\Big[\{\tilde{\mathbf{m}}-\widehat{\mathbf{q}}\circ\left(\mathbf{J}_\rho\dot{\boldsymbol{\Omega}}\right)\circ\widehat{\mathbf{q}}^*\\
&\qquad-\boldsymbol{\omega}\times\left(\widehat{\mathbf{q}}\circ\left(\mathbf{J}_\rho\boldsymbol{\Omega}\right)\circ\widehat{\mathbf{q}}^*\right)\}\circ\widehat{\mathbf{q}}\Big]\cdot\delta\widehat{\mathbf{q}}\,dx\\
&-\int_0^L 2\lambda\widehat{\mathbf{q}}\cdot\delta\widehat{\mathbf{q}}\,dx-\int_0^L\left(\widehat{\mathbf{q}}\cdot\widehat{\mathbf{q}}-1\right)\delta\lambda\,dx+\boldsymbol{f}^L\cdot\delta\mathbf{r}^L\\
&+2\left(\boldsymbol{h}^L\circ\widehat{\mathbf{q}}\right)\cdot\delta\widehat{\mathbf{q}}^L-\boldsymbol{f}^0(t)\cdot\delta\mathbf{r}^0-2\left(\boldsymbol{h}^0\circ\widehat{\mathbf{q}}\right)\cdot\delta\widehat{\mathbf{q}}^0.
\end{aligned}\tag{16}$$

A. The quaternion-based equations of dynamic equilibrium

Equation (16) represents the variational principle in which the variations $\delta_{\mathrm{rel}}\boldsymbol{\Gamma}$, $\delta_{\mathrm{rel}}\boldsymbol{K}$, $\delta\mathbf{r}$ and $\delta\widehat{\mathbf{q}}$ are not independent functions. The weak kinematic constraints (10)–(11) are therefore inserted into (16) to employ the fundamental lemma of the calculus of variations, which yields the continuous balance equations of a three-dimensional beam in quaternion notation:

$$\mathbf{n}'+\tilde{\mathbf{n}}-\rho A_r\ddot{\mathbf{r}}=\mathbf{0}\tag{17}$$

$$\begin{aligned}
&\Big[\mathbf{m}'+\mathbf{r}'\times\mathbf{n}+\tilde{\mathbf{m}}-\widehat{\mathbf{q}}\circ\left(\mathbf{J}_\rho\dot{\boldsymbol{\Omega}}\right)\circ\widehat{\mathbf{q}}^*\\
&\qquad-\boldsymbol{\omega}\times\left(\widehat{\mathbf{q}}\circ\left(\mathbf{J}_\rho\boldsymbol{\Omega}\right)\circ\widehat{\mathbf{q}}^*\right)-\lambda\widehat{\mathbf{1}}\Big]\circ\widehat{\mathbf{q}}=\widehat{\mathbf{0}}
\end{aligned}\tag{18}$$

$$\widehat{\mathbf{q}}\cdot\widehat{\mathbf{q}}-1=0\tag{19}$$

together with the boundary conditions:

$$\mathbf{n}^0-\mathbf{f}^0=\mathbf{0}\tag{20}$$

$$\left(\mathbf{m}^0-\mathbf{h}^0\right)\circ\widehat{\mathbf{q}}^0=\widehat{\mathbf{0}}\tag{21}$$

$$\mathbf{n}^L-\mathbf{f}^L=\mathbf{0}\tag{22}$$

$$\left(\mathbf{m}^L-\mathbf{h}^L\right)\circ\widehat{\mathbf{q}}^L=\widehat{\mathbf{0}}.\tag{23}$$

Here, $\mathbf{n}$ and $\mathbf{m}$ represent stress-resultant force and moment vectors of the cross-section with respect to the fixed basis, i.e.

$$\boldsymbol{n}=\widehat{\mathbf{q}}\circ\mathbf{N}\circ\widehat{\mathbf{q}}^*,\qquad\boldsymbol{m}=\widehat{\mathbf{q}}\circ\mathbf{M}\circ\widehat{\mathbf{q}}^*.\tag{24}$$

Equations (17)–(19) represent a system of eight governing equations for eight unknown functions – three components of displacement vector, four components of rotational quaternion and the Lagrangian multiplier. Equation (17) is identical to the standard linear momentum balance equation as it does not depend on rotation. In contrast, the balance equation (18) differs from the standard angular momentum balance equation. Using the notation $\mathcal{M}$ for the standard form of balance equation

$$\mathcal{M}=\mathbf{m}'+\mathbf{r}'\times\mathbf{n}+\tilde{\mathbf{m}}-\mathbf{R}\mathbf{J}_\rho\dot{\boldsymbol{\Omega}}-\boldsymbol{\omega}\times\mathbf{R}\mathbf{J}_\rho\boldsymbol{\Omega}\tag{25}$$

and replacing rotation matrix with rotational quaternion (3) in (18), yields

$$\left[\mathcal{M}-\lambda\widehat{\mathbf{1}}\right]\circ\widehat{\mathbf{q}}=\widehat{\mathbf{0}}.\tag{26}$$

Equation (26) represents the extension of the angular momentum balance equation to quaternion algebra as it follows from generalized d'Alembert principle. After (26) is multiplied on the right by $\widehat{\mathbf{q}}^*$ and the unity of quaternion $\widehat{\mathbf{q}}$ is considered, we get

$$\widehat{\mathcal{M}} - \lambda\widehat{\mathbf{1}} = \widehat{\mathbf{0}}$$

or, equivalently,

$$\lambda = 0 \tag{27}$$

$$\mathcal{M} = \mathbf{0}, \tag{28}$$

since $\mathcal{M}$ is a pure quaternion and λ is a scalar. Thus, Lagrangian multiplier λ vanishes for this problem and it could be eliminated from the set of unknown quantities iff the unity of quaternions is satisfied. Note that the unit-norm constraint is not neccessarily preserved after the discretization. When (19) is exactly preserved by the solution procedure the standard moment equilibrium equation can be used. Both approaches will be presented in numerical formulations.

VI. Numerical implementations for statics

In static analysis, inertial terms in governing equations vanish, which leads to a system of ordinary algebraic equations. In finite element approach, these differential equations are replaced by a set of non-linear algebraic equations and therefore the algebraic constraint for quaternions fits well into the numerical solution method. The consistent quaternion-based approach introduced eight equations (17)–(19) for eight primary unknowns $\mathbf{r}$, $\widehat{\mathbf{q}}$ and λ. The increments of the primary unknowns can be interpolated in a standard manner:

$$\Delta\mathbf{r}\left(x\right) = \sum_i P_i\left(x\right)\Delta\mathbf{r}^i, \qquad \Delta\mathbf{r}^i = \Delta\mathbf{r}\left(x_i\right) \tag{29}$$

$$\Delta\widehat{\mathbf{q}}\left(x\right) = \sum_i P_i\left(x\right)\Delta\widehat{\mathbf{q}}^i, \qquad \Delta\widehat{\mathbf{q}}^i = \Delta\widehat{\mathbf{q}}\left(x_i\right) \tag{30}$$

$$\Delta\lambda\left(x\right) = \sum_i P_i\left(x\right)\Delta\lambda^i, \qquad \Delta\lambda^i = \Delta\lambda\left(x_i\right), \tag{31}$$

where $x_i \in [0, L]$, $i = 0, 1, 2, ..., N+1$, with $x_0 = 0$ and $x_{N+1} = L$, are the *discretization points* and $P_i(x)$ are the interpolation functions. It is evident that such approach introduces additional degrees of freedom, but we should stress that the elements of the tangent-stiff matrix are computationally relatively inexpensive to evaluate. We could expect that the constraint (19) would enforce the unity of quaternions and that a standard additive update can be used. Unfortunately, the incremental quaternions are not unit, which leads to severe numerical problems observed at iteration procedure when solving discrete non-linear equations. The normalization of incremental quaternions does not result in considerably better convergence properties, but they are extremely improved after employing the kinematically consistent update. To obtain the updated rotational quaternion in a consistent manner, the following formula directly derived from (12) is used

$$D\widehat{\mathbf{q}}^i = \cos\left(\left|\Delta\widehat{\mathbf{q}}^i \circ \widehat{\mathbf{q}}^{*i}\right|\right) + \frac{\sin\left(\left|\Delta\widehat{\mathbf{q}}^i \circ \widehat{\mathbf{q}}^{*i}\right|\right)}{\left|\Delta\widehat{\mathbf{q}}^i \circ \widehat{\mathbf{q}}^{*i}\right|}\Delta\widehat{\mathbf{q}}^i \circ \widehat{\mathbf{q}}^{*i}. \tag{32}$$

The updated rotational quaternion is then obtained by multiplying two unit quaternions:

$$\widehat{\mathbf{q}}^{i[n+1]} = D\widehat{\mathbf{q}}^i \circ \widehat{\mathbf{q}}^{i[n]}. \tag{33}$$

The update procedure preserves the unity of rotational quaternions at interpolation points. Between the interpolation points the unity is enforced in the resultant sense as we demand $\int_0^L P_i\left(\widehat{\mathbf{q}} \cdot \widehat{\mathbf{q}} - 1\right) dx = 0.$

For comparison reasons a similar model was proposed but without introducing the additional Lagrangian multiplier λ and by omitting the constraint (19). Such approach reduces the number of degrees of freedom, but requires greater care since the unit norm constraint of rotational quaternions is preserved only pointwise at the interpolation nodes following (32)–(33). To increase the accuracy of numerical integration we employed the same update procedure also for the quaternions at the integration nodes, which were additionally stored during iteration.

Both approaches were proven to be computationally efficient and they both give very accurate results. It was observed, however, that the second approach might face some convergence problems, especially when a very high order of interpolation is used. The first approach does not suffer from such problems wich results in better efficiency and robustness.

A. *Bending of 45° arch*

We will present the results for the classical test problem by Bathe and Bolourchi [14], see Figure 2.

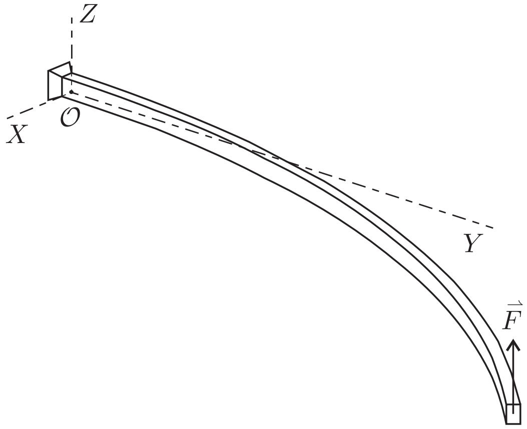


Figure 2. 45° arch.

The circular arch with the radius 100 is located in the horizontal plane and clamped at one end. The cross-section is taken to be a unit square. The arch is subjected to a vertical load $F = 600$ at the free end. The elastic and the shear moduli of material are $E = 10^7$ and $G = E/2$.

TABLE I. Free-end position of the 45° bend cantilever.

formulation	r_X	r_Y	r_Z
present, consistent, $N = 1$	15.29	47.42	53.47
present, consistent, $N = 3$	15.29	47.42	53.47
present, consistent, $N = 7$	15.29	47.42	53.47
present, consistent, $N = 15$	15.29	47.42	53.47
present, reduced, $N = 1$	15.91	46.98	53.94
present, reduced, $N = 3$	15.79	46.92	53.42
present, reduced, $N = 7$	15.74	47.15	53.43
present, reduced, $N = 15$	15.74	47.15	53.43
[14]	15.9	47.2	53.4
[3]	15.79	47.23	53.37

number of elements=8, N=number of internal points.

In Table I, we compare the results for the position vector of the free end of the cantilever. Eight straight elements of various order were used to obtain the results of the present formulation. The present results agree well with the results of other authors. Slightly better behaviour of consistent formulation with additional degree of freedom can be observed.

VII. Numerical implementations for dynamics

In dynamics analysis, we base our model on the reduced set of equations (17) and (28). To avoid obtaining the system of the algebraic-differential equations we enforce the constraint (19) as a part of numerical algorithm. Primary unknowns are therefore only $\mathbf{r}(x,t)$ and $\widehat{\mathbf{q}}(x,t)$. They are replaced by a set of one-parameter functions $\mathbf{r}^i(t) = \mathbf{r}(x_i,t)$ and $\widehat{\mathbf{q}}^i(t) = \widehat{\mathbf{q}}(x_i,t)$ at $N+2$ discretization points from the interval $[0,L]$. Similarly as for static case it is suitable to introduce an interpolation of these functions with respect to parameter x:

$$\mathbf{r}(x,t) = \sum_i P_i(x)\,\mathbf{r}^i(t) \qquad \widehat{\mathbf{q}}(x,t) = \sum_i P_i(x)\,\widehat{\mathbf{q}}^i(t)\,. \tag{34}$$

Discretization with respect to x results in a system of $7(N+2)$ ordinary second-order scalar differential equations for dynamic analysis for $7(N+2)$ unknown scalar functions $\mathbf{r}^i(t)$ and $\widehat{\mathbf{q}}^i(t)$.

Several procedures are possible for the time discretization. We have successfully employed the methods of the Runge-Kutta family. Note that the standard methods for solving systems of ordinary differential equations do not automatically conserve the unit norm of the rotational quaternion. In order to obtain the kinematically admissible results, the rotational quaternions are normalized after each time step has been completed.

An interesting alternative is to derive time integrators specially designed for rotational quaternions. Based on the standard Newmark scheme on the additive configuration spaces:

$$\mathbf{r}^{[n+1]} = \mathbf{r}^{[n]} + \Delta t\,\mathbf{v}^{[n]} + \Delta t^2\left[\left(\frac{1}{2}-\beta\right)\dot{\mathbf{v}}^{[n]} + \beta\,\dot{\mathbf{v}}^{[n+1]}\right]$$

$$\mathbf{v}^{[n+1]} = \mathbf{v}^{[n]} + \Delta t\left[(1-\gamma)\,\dot{\mathbf{v}}^{[n]} + \gamma\,\dot{\mathbf{v}}^{[n+1]}\right]$$

and the kinematically consistent update of quaternions (32)–(33) the following scheme is obtained:

$$\begin{aligned}\Delta\widehat{\mathbf{q}}^{[n]} &= \widehat{\mathbf{q}}^{[n]} \circ \frac{1}{2}\Bigg\{\Delta t\,\boldsymbol{\Omega}^{[n]} \\ &\qquad +\Delta t^2\left[\left(\frac{1}{2}-\beta\right)\dot{\boldsymbol{\Omega}}^{[n]} + \beta\,\dot{\boldsymbol{\Omega}}^{[n+1]}\right]\Bigg\} \\ \widehat{\mathbf{q}}^{[n+1]} &= \Bigg[\cos\left|\Delta\widehat{\mathbf{q}}^{[n]}\circ\widehat{\mathbf{q}}^{[n]*}\right| \\ &\qquad + \frac{\sin\left|\Delta\widehat{\mathbf{q}}^{[n]}\circ\widehat{\mathbf{q}}^{[n]*}\right|}{\left|\Delta\widehat{\mathbf{q}}^{[n]}\circ\widehat{\mathbf{q}}^{[n]*}\right|}\Delta\widehat{\mathbf{q}}^{[n]}\circ\widehat{\mathbf{q}}^{[n]*}\Bigg]\circ\widehat{\mathbf{q}}^{[n]} \\ \boldsymbol{\Omega}^{[n+1]} &= \boldsymbol{\Omega}^{[n]} + \Delta t\left[(1-\gamma)\,\dot{\boldsymbol{\Omega}}^{[n]} + \gamma\,\dot{\boldsymbol{\Omega}}^{[n+1]}\right].\end{aligned}$$

For both schemes $\beta \in \left[0,\frac{1}{2}\right]$ and $\gamma \in [0,1]$. The upper indices $[n]$ and $[n+1]$ denote the quantities at the previous and at the current time, t_n and t_{n+1}, respectively, while $\Delta t = t_{n+1} - t_n$ is the time increment.

A. Right angle cantilever

This example is taken from [15]. The geometry and loading data are presented in Fig. 3. The remaining data reads:

$$\begin{aligned} &A_1 = A_2 = A_3 = A && EA = GA = 10^6 \\ &J_1 = J_2 = J_3 = J && EJ = GJ = 10^3 \\ &A\rho = 1 && \mathbf{J}_\rho = \mathrm{diag}\,[\ 20 \quad 10 \quad 10\ ]\,. \end{aligned}$$

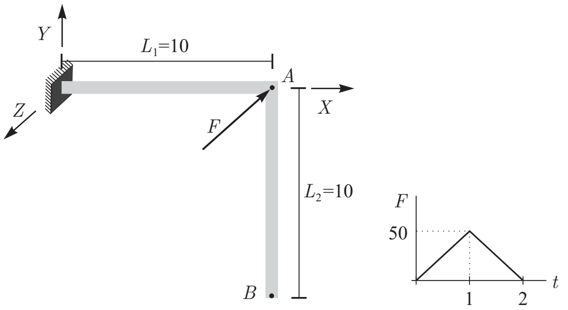


Figure 3. The right-angle cantilever beam subjected to out-of-plane loading.

The present results of the Newmark scheme-based method were obtained using the mesh consisting of 2×10 elements with three internal collocation points per element and the 6-point Gaussian integration rule.

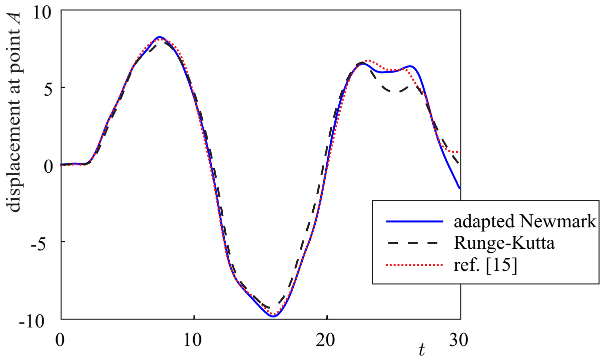


Figure 4. The right-angle cantilever: The right-angle comparison of displacements at point A.

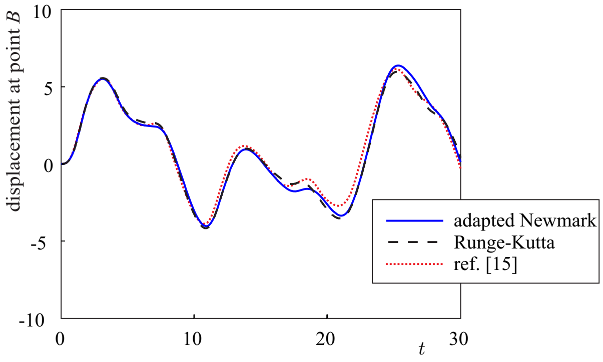


Figure 5. The right-angle cantilever: The right-angle comparison of displacements at point B.

On the whole time interval, $[0, 30]$, we used the same time step as Simo and Vu-Quoc [15]: $\Delta t = 0.25$. The results based on Runge-Kutta method were obtained for the mesh of 2×12 elements with two internal collocation points per element and the 4-point Gaussian integration rule. In contrast to Newmark scheme, the time step is not fixed and varies with respect to the prescribed local error tolerance, taken to be $\varepsilon = 10^{-5}$.

After the loading is removed at $t = 2$, the cantilever continues to vibrate freely, undergoing bending, torsion, axial deformations and large rotations. As observed from Fig. 4 and Fig. 5, the displacements are in a good agreement with [15], particularly in the time interval $[0, 15]$. With time differences between different time integrators become more evident.

VIII. Conclusion

Novel rotational-quaternion based approaches in finite-element methods for analysis of spatial frame structures under static and dynamic loads has been presented. The quaternion algebra was employed for the derivation of continuous equations of dynamic equilibrium with a special consideration of the unit norm constraint of rotational quaternion. Several approaches in discretization of the system of governing equation in terms of quaternions and several solution approaches are possible. The most suitable are the ones that are fully in accord with the nature of three-dimensional rotations and their quaternion representation. When properly treated the quaternions were found to be a computationally efficient, and robust tool for describing incremental and iterative rotations in non-linear beams.

References

[1] S. S. Antman, “Kirchhoffs problem for nonlinearly elastic rods,” Q. Appl. Math., vol. 32, no. 3, 1974, pp. 221–240.

[2] E. Reissner, “On finite deformations of space-curved beams,” Z. Angew. Math. Phys., vol. 32, no. 6, 1981, pp. 734–744.

[3] J. C. Simo, “A finite strain beam formulation - the three-dimensional dynamic problem. Part I.” Comput. Meth. Appl. Mech. Eng., vol. 49, no. 1, 1985, pp. 55–70.

[4] J. Argyris, “An excursion into large rotations,” Comput. Meth. Appl. Mech. Eng., vol. 32, no. 1-3, 1982, pp. 85–155.

[5] M. Borri and C. L. Bottasso, “An intrinsic beam model-based on a helicoidal approximation .1. formulation,” Int. J. Numer. Methods Eng., vol. 37, no. 13, 1994, pp. 2267–2289.

[6] G. Jelenic and M. A. Crisfield, “Geometrically exact 3D beam theory: implementation of a strain-invariant finite element for statics and dynamics,” Comput. Meth. Appl. Mech. Eng., vol. 171, no. 1-2, 1999, pp. 141–171.

[7] P. Betsch and P. Steinmann, “Frame-indifferent beam finite elements based upon the geometrically exact beam theory,” Int. J. Numer. Methods Eng., vol. 54, no. 12, 2002, pp. 1775–1788.

[8] H. Lang, J. Linn, and M. Arnold, “Multi-body dynamics simulation of geometrically exact Cosserat rods,” Multibody Syst. Dyn., vol. 25, no. 3, 2011, pp. 285–312.

[9] H. Lang and M. Arnold, “Numerical aspects in the dynamic simulation of geometrically exact rods,” Appl. Numer. Math., vol. 62, no. 10, SI, 2012, pp. 1411–1427.

[10] E. Zupan, M. Saje, and D. Zupan, “The quaternion-based three-dimensional beam theory,” Comput. Meth. Appl. Mech. Eng., vol. 198, no. 49-52, 2009, pp. 3944–3956.

[11] ——, “Quaternion-based dynamics of geometrically nonlinear spatial beams using the Runge-Kutta method,” Finite Elem. Anal. Des., vol. 54, 2012, pp. 48–60.

[12] ——, “Dynamics of spatial beams in quaternion description based on the Newmark integration scheme,” Comput. Mech., vol. 51, no. 1, 2013, pp. 47–64.

[13] ——, “On a virtual work consistent three-dimensional ReissnerSimo beam formulation using the quaternion algebra,” Acta Mechanica, vol. in press, 2013.

[14] K. J. Bathe and S. Bolourchi, “Large displacement analysis of 3-dimensional beam structures,” Int. J. Numer. Methods Eng., vol. 14, no. 7, 1979, pp. 961–986.

[15] J. C. Simo and L. Vu-Quoc, “On the dynamics in space of rods undergoing large motions - a geometrically exact approach,” Comput. Meth. Appl. Mech. Eng., vol. 66, no. 2, 1988, pp. 125–161.

Interpretable Knowledge Acquisition for Predicting Bioluminescent Proteins Using an Evolutionary Fuzzy Classifier Method

Hui-Ling Huang[1,2,3], Hua-Chin Lee[1,2,3], Phasit Charoenkwan[1,2], Wen-Lin Huang[4], Li-Sun Shu[5] and Shinn-Ying Ho[1,2,3]*

[1]Institute of Bioinformatics and Systems Biology, National Chiao Tung University (NCTU), Taiwan
[2]Department of Biological Science and Technology, NCTU, Taiwan
[3]Center for Bioinformatics Research, NCTU, Taiwan
[4] Department of Management Information System, Asia Pacific Institute of Creativity, Taiwan
[5]Department of Multimedia and Game Design, Overseas Chinese University, Taichung, Taiwan
*Corresponding Email: hlhuang@mail.nctu.edu.tw, syho@mail.nctu.edu.tw

***Abstract*—New applications of using bioluminescent proteins (BLPs) are constantly increasing in a variety of research fields such as protein engineering of using single-cell bioluminescent organisms to determine how animals move through water. In this study, we propose a knowledge acquisition method for characterizing BLPs and understanding their functions using a compact set of fuzzy rules. The rule set was obtained by designing an if-then fuzzy-rule-based bioluminescent protein classifier (named iFBPC) with physicochemical properties as input features. In designing iFBPC, feature selection, membership function design, and fuzzy rule base generation are all simultaneously optimized using an intelligent genetic algorithm (IGA). We used the same benchmark dataset for comparisons used in existing SVM-based prediction methods BLProt and PBLP using 100 and 15 features of physicochemical properties, respectively. The classifier iFBPC has two fuzzy rules (one for BLP and the other for non-BLP) and four physicochemical properties with test accuracy of 74.82% where BLProt and PBLP have accuracies of 80.06% and 81.79%, respectively. The four physicochemical properties are structures, protein linkers, nucleation, and membrane proteins in the AAindex database. The analysis of characterizing BLPs was conducted based on knowledge of the fuzzy rule base.**

Keywords-bioluminescent proteins; feature selection; fuzzy rules; genetic algorithm; knowledge acquisition; physicochemical properties.

I. INTRODUCTION

Bioluminescence of organisms occurs in diverse forms of morphology, with various mechanisms of light emission. Then the cited state of the emitter will emit light with a very short lifetime. After releasing the energy in the form of a photon, the reaction time just keep a few nanoseconds. At quite another situation, fluorophore is another substance, which can generate light. It can acquire its excitation energy in bypassing excitation of primary emitter. For example, the green flurocense protein (GFP) usually use the covalently bond of fluorophore. In Aequorea GFP [1], the post-translational reaction of cyclization, dehydration and oxidation of Ser65-Tyr66-Gly67 [2] because the emitting light and hence it can be easily expressed in eukaryotic and prokaryotic orgasms without losing its emitting function. No coelenterate-specific enzymes are needed to join the reaction. Understanding physicochemical properties of the bioluminescent proteins (BLPs) may help improve the applications of BLPs.

The experimental methods [3][4] to identify the BLPs could be often time-consuming expensive and have very limited scopes due to some restrictions for many enzymatic reactions. Recently, researchers have interests in computational methods, which have been developed to predict BLPs.

Kandaswamy et al. [5] first proposed a predictive method, as known as BLProt, based on support vector machine (SVM) and physicochemical properties to predict BLPs. The three different filter approaches, ReliefF, infogain, and mRMR were utilized to identify the most informative features. In 2011, Huang et al. [6] proposed a novel method using the physicochemical properties (PBLP). In that work PBLP, an efficient algorithm inheritable bi-objective genetic algorithm (IBCGA) [7] was used to select significant features, which could discriminate the two classes of BLPs. Recently, Zhao et al. [8] developed a new computational method to predict BLPs using a model based on position specific scoring matrix and auto covariance (PSSM-AC). Their results showed that accuracy of PSSM-AC model was higher than BLProt and PBLP. The existing methods [5][6][8] can predict BLPs but suffer from obtaining human-interpretable knowledge from sequences.

In our previous work, PBLP [6] investigates the optimal design of predictors for predicting from amino acid sequence using both informative features and an appropriate classifier. Furthermore, we obtained a set of relevant physicochemical properties can advance prediction performance. The proposed PBLP identified m=15 features of properties for predicting BLPs with an independent test accuracy of 81.79%. Since the set of 15 physicochemical properties performs well, we would apply it to acquire the rule-based knowledge for predicting and analyzing BLPs.

In this paper, we design an interpretable fuzzy rule classifier based on the 15 physicochemical properties as features [6]. The proposed classifier with an accurate and compact fuzzy rule base using a scatter partition of feature space for BLPs is named iFBPC. Because BLPs from database [6] have the property of natural clustering, fuzzy classifiers using a scatter partition of feature spaces often have a smaller number of rules than those using grid partitions. The design of iFBPC has three objectives to be simultaneously optimized: maximal classification accuracy, minimal number of rules, and minimal number of used physicochemical properties. In designing iFBPC, the flexible membership function, fuzzy rule, and physicochemical properties selection are simultaneously optimized. Huang et al. [9] applied an intelligent genetic algorithm (IGA) [10] to efficiently solve the design problem with a large number of tuning parameters.

The iFBPC built with 2 rules and 4 physicochemical properties have fine training accuracy of 73.67% and test accuracy of 74.82%. These results are suggested that iFBPC provides the interpretable and confidant rules that can will identify the BLPs. The results show that the membrane protein properties are most important to BLPs and the amino acids prone to locate at terminal of the alpha-helix are not preferred in BLPs. This is might be caused from the working environment of BLPs and these results would also give the biologists the considerations about the protein engineering examinations for altering the BLP stability.

The rest of this paper is organized as follows. Section II describes the materials and methods used. Section III describes the results and performance, and Section IV addresses the conclusions of this paper. Finally, the acknowledgement closes the article.

II. Materials and Methods

We propose a fuzzy rule-based knowledge acquisition system based on interpretable if-than fuzzy classifiers (iFBPC). The design of iFBPC is provided with an accurate and compact fuzzy rule base using a scatter partition of feature space for bioluminescent protein data analysis. The framework is presented in Fig. 1.

A. Dataset

The bioluminescent proteins (BLPs) were extracted from Kandaswamy et al. [5]. More details about this data set can be found by [5][6][8]. After all, a total 441 BLPs are kept as positive dataset. The statistic of the training and test sets is shown in Table I. 300 BLPs are random selected from the 441 positive dataset and served as training dataset. The others are served as testing dataset. 300 non-BLPs are also randomly picked from seed proteins of Pfam protein families. These proteins, served as negative dataset, are unrelated to BLPs. The negative testing dataset is composed of the 141 non-BLPs Pfam protein families and are different from training non-BLPs. Finally, the testing dataset is composed of 141 BLPs and 141 non-BLPs.

TABLE I - THE STATISTIC OF THE TRAINING/TEST SETS.

Dataset	Number of BLPs	Number of non-BLPs
Training	300	300
Test	141	141

B. Feature set

Considering the BLPs data set, the set of m=15 informative properties (PCPs) identified by PBLP performs best where the best solution with accuracy of 84.11% is used [6]. The PBLP is a systematic approach to automatically identify a set of physicochemical and biochemical properties in the AAindex database to design SVM-based classifiers for predicting and analyzing BLPs. The set of m=15 PCPs is identified by PBLP, we would apply it to acquire the rule-based knowledge for predicting and analyzing BLPs data set. The set of 15 PCPs is described in Table II.

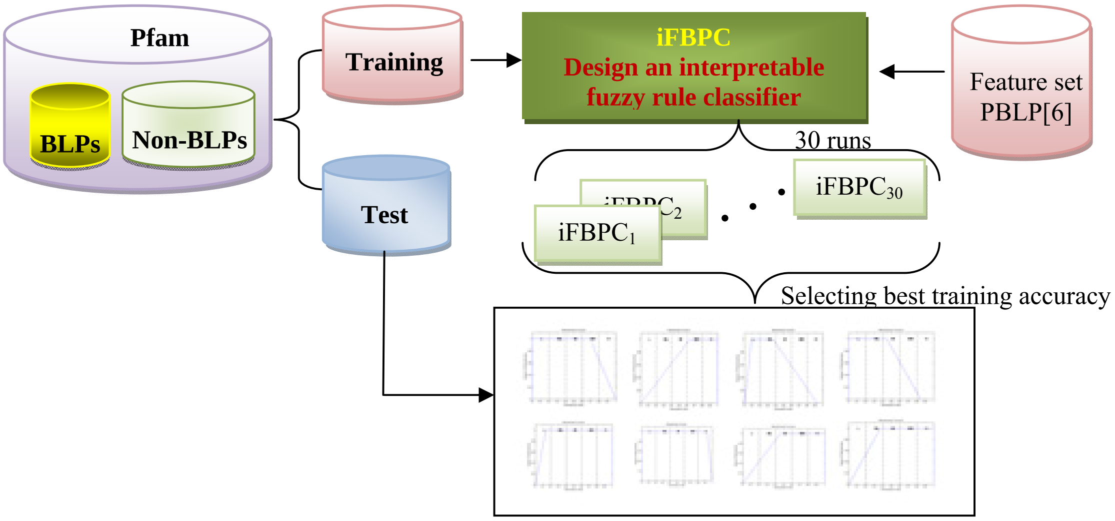


Figure 1. The framework of if-than fuzzy rule-based classifier for bioluminescent proteins (iFBPC).

TABLE II - THE PBLP INDENTED A SET OF *M*=15 PHYSICOCHEMICAL PROPERTIES ON BLPs.

Feature ID	AAindex ID	Description
8	BHAR880101	Positional flexibilities of amino acid residues in globular proteins
13	BROC820102	The isolation of peptides by high-performance liquid chromatography using predicted elution positions
18	BUNA790103	1H-nmr parameters of the common amino acid residues measured in aqueous solutions of the linear tetrapeptides H-Gly-Gly-X-L-Ala-OH
95	FINA910104	Physical reasons for secondary structure stability: alpha-helices in short peptides
107	GEIM800111	Amino acid preferences for secondary structure vary with protein class
202	NAKH920101	The amino acid composition is different between the cytoplasmic and extracellular sides in membrane proteins
223	PALJ810101	Protein secondary structure
310	RACS820111	Differential geometry and polymer conformation. 4. Conformational and nucleation properties of individual amino acids
380	VENT840101	Hydrophobicity parameters and the bitter taste of L-amino acids
439	PARS000102	Protein thermal stability: insights from atomic displacement parameters (B values)
473	MITS020101	Amphiphilicity index of polar amino acids as an aid in the characterization of amino acid preference at membrane-water interfaces
475	TSAJ990102	The packing density in proteins: standard radii and volumes
489	PUNT030101	A knowledge-based scale for amino acid membrane propensity
491	GEOR030101	An analysis of protein domain linkers: their classification and role in protein folding

An FGPMF $\mu(x)$ with a single fuzzy set is defined as

C. *Acquisition of the rule-based knowledge*

The performance of iFBPC mainly arises from two aspects. One is to simultaneously optimize all parameters in the design of iFBPC where all the elements of the fuzzy classifier design have been transformed into parameters of a large parameter optimization problem. The other is to use an efficient optimization algorithm IGA, which is a specific variant of the intelligent evolutionary algorithm [10]. The intelligent evolutionary algorithm uses a divide-and-conquer strategy to effectively solve large parameter optimization problems. IGA is shown to be effective in the design of accurate classifiers with a concise fuzzy rule base using an evolutionary scatter partition of feature space [11].

The proposed iFBPC design involves: 1) flexible generic parameterized membership functions (FGPMFs) and a hyperbox-type fuzzy partition of feature space, 2) determining a fuzzy reasoning method and fuzzy if-then rules corresponding to fuzzy regions, and 3) determining a fitness function and a chromosome representation for using IGA to optimize the system's tuning parameters.

$$\mu(x) = \begin{cases} 0 & \text{if } x \le a \text{ or } x \ge d \\ \dfrac{x-a}{b-a} & \text{if } a < x < b \\ \dfrac{d-x}{d-c} & \text{if } c < x < d \\ 1 & \text{if } b \le x \le c \end{cases} \qquad (1)$$

where $x \in [0, 1]$ and $a \leq b \leq c \leq d$. Some illuminations of FGPMF are shown in Fig. 2. The variables *a*, *b*, *c* and *d* determining the shape of a trapezoidal fuzzy set are the parameters to be optimized. This transformative scheme of training patterns and the encoded parameters of the IGA's chromosomes have been described more detail in previous research [9].

D. *Fuzzy if–then rule and Fuzzy reasoning method*

The following fuzzy if–then rule base for *n*-dimensional classification problems are used in the design of iFBPC:

R_j : If x_1 is A_{j1} and . . . and x_n is A_{jn} then class CL_j with CF_j, $j = 1, . . . , N$.

where R_j is a rule label, x_i denotes a variable of physicochemical property, A_{ji} is an antecedent fuzzy set, C is a number of classes, $CL_j \in \{1, . . ., C\}$ denotes a consequent class label, CF_j is a certainty grade of this rule in the unit interval [0, 1], and N is a number of initial fuzzy rules in the training phase. In this study, C=2 (two classes for BLPs and non-BLPs), n=15 (initial number in the feature set to be selected), and N=3C (initial number in the rule set to be selected).

To enhance interpretability of fuzzy rules, linguistic variables in fuzzy rules can be used. Each variable xi has a linguistic set U = {S (small), SM (small medium), M (medium), ML (medium large), L (large)}. Each linguistic value of xi equally represents 1/5 of the domain [0, 1]. Examples of linguistic antecedent fuzzy sets are shown in Fig. 3.

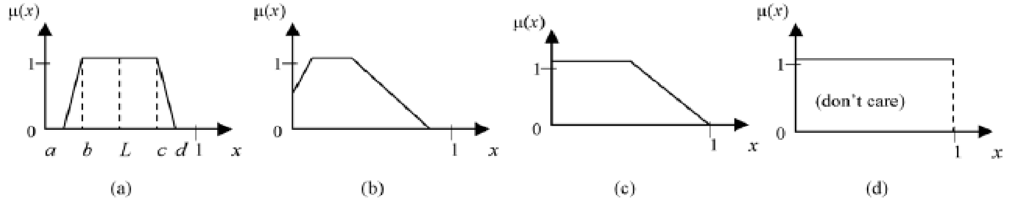


Figure 2. Illuminations of FGPMF: (a) a>0 and d< 1; (b) a<0<b, (c) b≦0; (d) b≦0 and c≧1.

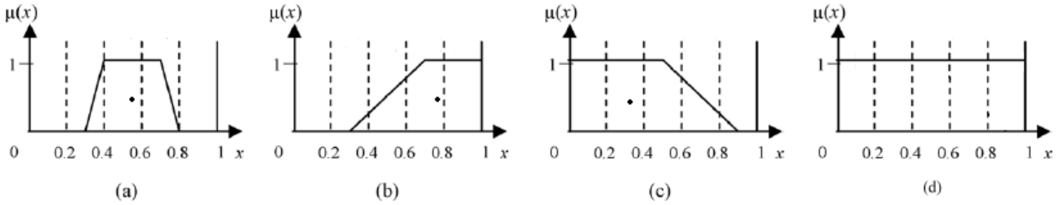


Figure 3. Examples of an antecedent fuzzy set A_{ji} with linguistic values (L: low, ML: medium low, M: medium, MH: medium high, H: high): (a) A_{ji} represents {ML, M, MH}; (b) A_{ji} represents {ML, M, MH, H}, i.e., not Low; (c) A_{ji} represents {L, ML, M, MH, H} or ALL.

In the training phase, all the variables CL_j and CF_j are treated as parametric genes encoded in a chromosome and their values are obtained using IGA. The following fuzzy reasoning method is adopted to determine the class of an input pattern $x_p = (x_{p1}, x_{p2}, . . ., x_{pn})$ based on voting using multiple fuzzy if–then rules:

Step 1: Calculate score $S_{\text{Class}v}$ (v = 1, . . . , C) for each class as follows:

$$S_{\text{Classv}} = \sum_{\substack{R_j \in FC \\ CL_j = Class\ v}} \mu_j(x_p) CF_j, \quad \mu_j(x_p) = \prod_{i=1}^{n} \mu_{ji}(x_{pi}), \quad (2)$$

where FC denotes the fuzzy classifier, and $\mu_{ji}(\cdot)$ represents the membership function of the antecedent fuzzy set A_{ji}.

Step 2: Classify x_p as the class with a maximal value of S_{Classv}.

Notably, x_p is classified into the BLP or non-BLP class for one iFBPC. The final classification of x_p is determined using the proposed classifier iFBPC in the study.

E. IGA to optimize the system's tuning parameters

A GA-chromosome consists of control GA-genes for selecting useful features and significant fuzzy rules, and parametric GA-genes for encoding the membership functions and fuzzy rules. The control GA-gene comprises two types of parameters. One is parameter r_j, j=1,…N, represented by one bit for eliminating unnecessary fuzzy rules. The other is parameter f_i, i=1,…N, represented by one bit or eliminating useless features. The parametric genes determine variables of three types: $V_{ji}^{t} \in [0, 1]$, t=1, …, 5, for determining the antecedent fuzzy set A_{ji}, CL_j for determining the consequent class label of rule R_j, and $CF_j \in [0, 1]$ for determining the certainty grade of rule R_j , where j=1, ..., N and i=1, ..., n. A rule base with N fuzzy rules is represented as an individual. The detailed explanation of the chromosome representation and implementation can be referred to [9]. The design of an efficient fuzzy classifier is formulated as a large parameter optimization problem. Once the solution of IGA is obtained, an accurate classifier with a concise fuzzy rule base can be obtained.

We define the fitness function of IGA for designing iFBPC as follows:

$$\max Fit(FC) = ACC - W_r N_r - W_f N_f \quad (3)$$

where W_r and W_f are positive weights. In this study, the fitness function is used to optimize the three objectives:

1) to maximize the classification accuracy ACC,
2) to minimize the number N_r of fuzzy rules, and
3) to minimize the number N_f of selected features.

The trade-off between prediction accuracy and conciseness of the rule base can be determined by tuning the weights W_r and W_f. For obtaining an easily-interpretable and compact knowledge rule base with concise iFBPC, the small values of N_r and N_f are preferred. Therefore, we used large values of penalty weights W_r = 0.6 and W_f = 0.1. If the high accuracy of an individual iFBPC is the most important objective, small values of penalty weights are preferred. The simulation results show that the weights W_r and W_f are not very sensitive to the accuracy of the obtained solutions using IGA. To further advance the prediction accuracy of predicting BLP is utilized.

III. Results

The parameter settings of IGA [10] are N_{pop} = 20, P_c = 0.7, P_s =1−P_c, P_m = 0.01 and α = 15. Because the search space of the optimal design of iFBPC is proportional to the number N_p of parameters to be optimized, the stopping condition is suggested to use a fixed number 100N_p of fitness evaluations.

A. Prediction performance evaluation

The training samples with 15 properties in the dataset BLPs are represented as 15-dimensional feature vectors. This set of 15 physicochemical properties is identified by PBLPs [12]. Due to the non-deterministic characteristic of genetic algorithms, the average performance of 30 independent iFBPC is given in Table III. The top six of high selected frequency PCPs in the 30 runs are shown in Table IV.

TABLE III. THE AVERAGE VALUES OF 30 INDEPENDENT RUNS OF THE PROPOSED iFBPC.

Training			Test
Accuracy. (%)	Feature no.	Rule no.	Accuracy (%)
73.67	3.67	2	74.82%

TABLE IV. THE TOP SIX OF HIGH SELECTED FREQUENCY PCPs IN THE 30 RUNS.

Freq.	Feature No.	AAindex No.	Category
20	489	PUNT030101	Membrane Protein
17	202	NAKH920101	Membrane Protein
12	491	GEOR030101	Protein linker
10	223	PALJ810101	Structure
10	475	TSAJ990102	Structure
10	502	ZHOH040103	Hydrophobicity

B. Rule-based knowledge

We selected one iFBPC_1 with best training accuracy in the independent 30 runs, to illustrate the rules for bioluminescent proteins mechanism. The iFBPC_1 has training accuracies of 73.67%, the test accuracies of 74.82%, the feature numbers N_f of 4, and the rule numbers N_r of 2, respectively. The selected physicochemical properties are BHAR880101 (Protein linker), GEIM800111 (Structure), PUNT030101 (Membrane Protein) and FINA910104 (Nucleation), shown in Fig 4. The fuzzy rules are linguistically interpretable as follows:

Fuzzy Classifier iFBPC_1:

R1: if BHAR880101 is ALL, GEIM800111 is {medium, large}, PUNT030101 is {small, medium} and FINA910104 is {small, medium}, then BLPs with CF=0.714.

R2: if BHAR880101 is ALL, GEIM800111 is ALL, PUNT030101 is {medium, large} and FINA910104 is {medium, large}, then non- BLPs with CF= 0.267.

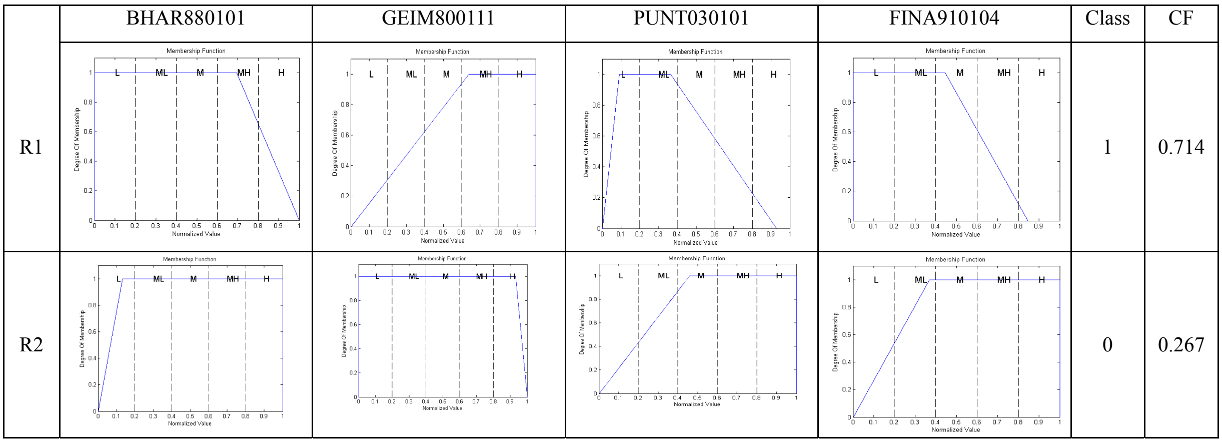


Figure 4. Fuzzy rules of the selected 4 PCPs, the training ACC is 73.67% and testing ACC is 74.82%. Class: 1 for BLPs and 0 for non-BLPs.

C. The physicochemical properties of BLPs

The 15 informative PCPs are further classed into 5 categories that are structures, hydrophobicity, protein linkers, nucleation and membrane proteins. The importance of bioluminescence protein versus protein linker, hydrophobicity, structure is documented in previous work [11][12]. The BLPs works, sometimes they will meet a hydrophobic environment that caused by the luciferin, a quite hydrophobic substance [13]. The rules, which could stabilized the structures of BLPs at hydrophobic environments, of membrane protein folding could be also considered.

From the fuzzy rules, the BHAR880101, the structure of flexibility, and GEIM800111, aperiodic indices for alpha/beta-proteins, should be considered both in BLPs or non-BLPs. PUNT030101, a membrane protein properties, and FINA910104, the helix termination parameter at position, show reversed results in BLP and non-BLPs. In BLPs, the property, PUNT030101, should be small to medium. In original study [14] of this properties, the authors defined that a negative value indicated a high membrane propensity. This can be interpreted that the BLPs would have some properties that membrane proteins also have as mentioned in previous study [6].

It is interesting that the FINA910104 also shows reversed results in BLPs and non-BLPs. This property, the index about the amino acids locate at the C-terminal of alpha-helix, is driven from the nucleus structure study. From previous study, some proteins mainly composed of beta-sheets will transform to alpha-helices in partial organic solvent suggesting that this kind of solvent could keep the alpha-helical structure [15]. Forming the C-Capping will increase in alpha-helicity [16] and would cause the BLPs to form alpha-helix from native structures, which are not alpha-helix in such partial organic solvent. The BLPs would use the

ISBN: 978-1-61208-354-4

strategy that to avoid being composed of the amino acids, which favors to locate at the alpha-helix terminal and often forms the C-Capping structure to escape alpha-helixes. This can keep the original structure of BLPs away from the structural transformation and would maintain the biological function of BLPs.

IV. CONCLUSION

The iFBPC is a high performance sequence-based classifier for identifying the BLPs based on fuzzy-rule classifier. It provides a high confident fuzzy rule that could identify the BLPs well and also provide some useful knowledge. In BLPs, the membrane properties are important because BLPs work at the partial organic solvent, which will change the folding nature of proteins and make the proteins lose their functions. BLPs would use the strategy that avoiding being composed of the amino acids favoring to locate at the terminal of alpha-helices. This strategy could provide the protein engineers a new though for protein engineering.

ACKNOWLEDGMENT

This work was funded by National Science Council of Taiwan under the contract number NSC-102-2221-E-009-169-, and "Center for Bioinformatics Research of Aiming for the Top University Program" of the National Chiao Tung University and Ministry of Education, Taiwan, R.O.C. for the project 102W962. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

REFERENCES

[1] F. H. Johnson, O. Shimomura, Y. Saiga, L. C. Gershman, G. T. Reynolds, and J. R. Waters, “Quantum efficiency of Cypridina luminescence, with a note on that of Aequorea,” J. Cell. Comp. Physiol, vol. 60, no. 1, 1962.

[2] D. C. Prasher, V. K. Eckenrode, W. W. Ward, F. G. Prendergast, and M. J. Cormier, “Primary Structure of the Aequorea-Victoria Green-Fluorescent Protein,” Gene, vol. 111, no. 2, Feb 15, 1992, pp. 229-233.

[3] O. Shimomura, F. H. Johnson, and Y. Saiga, “Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, Aequorea,” J Cell Comp Physiol, vol. 59, Jun, 1962, pp. 223-39.

[4] P. A. Vidi, and V. J. Watts, “Fluorescent and bioluminescent protein-fragment complementation assays in the study of G protein-coupled receptor oligomerization and signaling,” Mol Pharmacol, vol. 75, no. 4, Apr, 2009, pp. 733-9.

[5] K. K. Kandaswamy, G. Pugalenthi, M. K. Hazrati, K. U. Kalies, and T. Martinetz, “BLProt: prediction of bioluminescent proteins based on support vector machine and relieff feature selection,” BMC Bioinformatics, vol. 12, 2011, pp. 345.

[6] H.-L. Huang, Y.-F. Liou, H.-C. Lee, W.-L. Huang, and S.-Y. Ho, “Designing predictors of bioluminescence proteins using an efficient physicochemical property mining method,” IEEE International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2012), 2012.

[7] S. Y. Ho, J. H. Chen, and M. H. Huang, “Inheritable genetic algorithm for biobjective 0/1 combinatorial optimization problems and its applications,” Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics, vol. 34, no. 1, Feb, 2004, pp. 609-620.

[8] X. W. Zhao, J. K. Li, Y. X. Huang, Z. Q. Ma, and M. H. Yin, “Prediction of Bioluminescent Proteins Using Auto Covariance Transformation of Evolutional Profiles,” International Journal of Molecular Sciences, vol. 13, no. 3, Mar, 2012, pp. 3650-3660.

[9] H. L. Huang, F. L. Chang, S. J. Ho, L. S. Shu, W. L. Huang, and S. Y. Ho, “FRKAS: Knowledge Acquisition Using a Fuzzy Rule Base Approach to Insight of DNA-Binding Domains/Proteins,” Protein and Peptide Letters, vol. 20, no. 3, Mar, 2013, pp. 299-308.

[10] S. Y. Ho, L. S. Shu, and J. H. Chen, “Intelligent evolutionary algorithms for large parameter optimization problems,” Ieee Transactions on Evolutionary Computation, vol. 8, no. 6, Dec, 2004, pp. 522-541.

[11] S. A. Moore, and M. N. G. James, “Common Structural Features of the Luxf Protein and the Subunits of Bacterial Luciferase - Evidence for a (Beta-Alpha)(8) Fold in Luciferase,” Protein Science, vol. 3, no. 11, Nov, 1994, pp. 1914-1926.

[12] A. J. Fisher, T. B. Thompson, J. B. Thoden, T. O. Baldwin, and I. Rayment, “The 1.5-A resolution crystal structure of bacterial luciferase in low salt conditions,” J Biol Chem, vol. 271, no. 36, Sep 6, 1996, pp. 21956-68.

[13] G. W. J. Moss, N. P. Franks, and W. R. Lieb, “Modulation of the General Anesthetic Sensitivity of a Protein - a Transition between 2 Forms of Firefly Luciferase,” Proceedings of the National Academy of Sciences of the United States of America, vol. 88, no. 1, Jan, 1991, pp. 134-138.

[14] M. Punta, and A. Maritan, “A knowledge-based scale for amino acid membrane propensity,” Proteins-Structure Function and Genetics, vol. 50, no. 1, Jan 1, 2003, pp. 114-121.

[15] K. Shiraki, K. Nishikawa, and Y. Goto, “Trifluoroethanol-Induced Stabilization of the Alpha-Helical Structure of Beta-Lactoglobulin - Implication for Non-Hierarchical Protein-Folding,” Journal of Molecular Biology, vol. 245, no. 2, Jan 13, 1995, pp. 180-194.

[16] J. P. Schneider, and W. F. DeGrado, “The design of efficient alpha-helical C-capping auxiliaries,” Journal of the American Chemical Society, vol. 120, no. 12, Apr 1, 1998, pp. 2764-2767.

Fuzzy LMI Integral Control of DC Series Motor

Umar Farooq[1], Jason Gu[2], Jun Luo[3] and M. Usman Asad[4]
[1,4]Department of Electrical Engineering University of The Punjab Lahore-54590, Pakistan
[2]Department of Electrical and Computer Engineering Dalhousie University, Halifax, N.S., Canada
[3]School of Mechatronic Engineering & Automation, Shanghai University, China
E-mail: engr.umarfarooq@yahoo.com, jason.gu@dal.ca, luojun@shu.edu.cn, usmanasad01@hotmail.com

Abstract—**Based on the Takagi Sugeno (TS) fuzzy model of DC series motor, an integral controller is designed for controlling its speed. The nonlinear plant is converted into regional fuzzy models for which the control gains are found through a set of linear matrix inequalities (LMIs). The local fuzzy controllers are blended afterwards through parallel distributed compensation scheme to yield the final control law. MATLAB simulation results are presented to show the effectiveness of the designed controller.**

Keywords-DC series motor;TS fuzzy model;LMI integral controller;Parallel distributed compensation;MATLAB Simulink

I. INTRODUCTION

DC series motor is constructed by placing the field circuit in series wit the armature circuit and is therefore a popular choice for wide variety of applications which require high torques at low speeds such as electric traction applications [1]-[3]. The mathematical model of this motor is nonlinear due to a square law relationship between the torque it produces and the current supplied to it. Another source of nonlinearity is the product of current and speed which constitutes the back EMF. The model can be linearized for small range of operation. However, the dynamic operation requires the nonlinear model for designing the motor controller.

A variety of nonlinear control methods have been reported in literature for the speed control of dc series motors [4]-[15]. The use of feedback linearization technique is reported in [4]. Through nonlinear state transformation, a linear control law is designed to regulate the speed of the motor. A nonlinear observer for speed and load torque is also constructed based on the current measurements. The real time results of the proposed approach are also presented. Another nonlinear control method known as back stepping is employed in [8] to control the speed of dc series motor. By introducing the virtual control inputs in a recursive fashion, the control Lyapunov functions are found. An improved version of this method is reported in [9] for achieving better transient performance.

The model free techniques such as fuzzy logic and neural networks have also been used for the control of dc series motors [10]-[13]. A PID-ANN controller in [10] acquires training data from a conventional PID controller and the trained controller drives the dc motor through a chopper. The controller is also implemented in real time using an 80C51 microcontroller. A rule base fuzzy logic controller is described in [11] which employs speed and current controllers for dc series motor. Both the controllers accept error and change in error from the set points as inputs and generate the firing signals for thyristors after evaluating 49 rules in the rule base. The performance of the designed fuzzy scheme shows improvement compared with classical PI control strategy. The power of fuzzy logic controller for speed of dc series motors is presented in [12] where a simple fuzzy logic controller with a single input and single output is shown to perform better than PI controller.

This paper describes model based fuzzy control of dc series motor. Using Takagi-Sugeno fuzzy modeling approach, the nonlinear terms in the mathematical model are translated as fuzzy sets which yield a number of local linear models. A set of local controllers corresponding to these models are obtained after solving a set of LMIs which also guarantee the global stability of the control scheme. MATLAB simulations are performed to show the validity of the designed controller. The contribution of the paper lies in the design of integral fuzzy disturbance rejection controller for DC series motor with an objective to improve the transient performance of the system when load torque is changing frequently. The proposed controller clearly performs better than the pole placement state feedback controller as evident from the simulation results.

We start by developing the TS fuzzy model of DC series motor in Section II. Controller design is presented in Section III followed by results in Section IV. Conclusions are drawn in Section V.

II. TS FUZZY MODEL

The motion model of DC series motor after neglecting the magnetic saturation in field circuit can be given as [9]:

$$L\frac{di}{dt} = u - Ri - Mi\omega \tag{1}$$

$$J\frac{d\omega}{dt} = Mi^2 - T_L \tag{2}$$

$$T_e = Mi^2 \tag{3}$$

$$E = Mi\omega \tag{4}$$

where, 'i'is the armature (or field) current, 'u'is the terminal voltage, 'ω'is the rotation speed of the motor, 'L'is the net armature and field circuit inductance, 'R'is the net armature and field circuit resistance, 'J'is the inertia associated with both the motor and the load, 'M'is the

ISBN: 978-1-61208-354-4

motor constant, ‘ T_L ’is the load torque, ‘ T_e ’is the electromagnetic torque, and ‘ E ’is the back EMF. The motor parameters are adopted from [9] and are listed in Table 1.

Let, the state variables for the system be:

$$x_1 = \omega, x_2 = i, y = x_1 \tag{5}$$

Using (5), the system equations (1)-(2) can be given as:

$$\frac{dx_1}{dt} = \frac{M}{J} x_2{}^2 - \frac{T_L}{J} \tag{6}$$

$$\frac{dx_2}{dt} = -\frac{M}{L} x_2 x_1 - \frac{R}{L} x_2 + \frac{u}{L} \tag{7}$$

We can write (6)-(7) in matrix form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{M}{J} x_2 \\ -\frac{M}{L} x_2 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u + \begin{bmatrix} -\frac{1}{J} \\ 0 \end{bmatrix} T_L \tag{8}$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The general form of (8) is given as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}d$$
$$y = \mathbf{C}\mathbf{x} \tag{9}$$

where, ‘ $\mathbf{A}$ ’is the system matrix, ‘ $\mathbf{B}$ ’is the input vector, ‘ $\mathbf{E}$ ’is the disturbance vector, ‘ $\mathbf{x}$ ’ is the system state vector, ‘ u ’is the control input which is the motor voltage, ‘ d ’is the disturbance input which in this case is taken to be the load torque and ‘ $\mathbf{C}$ ’is the output vector.

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{M}{J} x_2 \\ -\frac{M}{L} x_2 & -\frac{R}{L} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}, \mathbf{E} = \begin{bmatrix} -\frac{1}{J} \\ 0 \end{bmatrix} \tag{10}$$

It can be seen from (10) that system matrix is state dependent. Thus, we will represent it in TS frame work. Let the state variable ‘ x_2 ’ be taken as premise variable (also known as the scheduling variable) that takes on values in the interval $[x_{2L}, x_{2U}]$. Then we can define the two triangular fuzzy sets namely ‘ M_1 ’and ‘ M_2 ’ that will be centered at x_{2L} and x_{2U} respectively, as shown in Fig. 1. Thus, we have two local subsystems given as:

$$\dot{\mathbf{x}}_i = \mathbf{A}_i\mathbf{x} + \mathbf{B}_i u + \mathbf{E}_i d, \ \ i = 1, 2 \tag{11}$$

where,

TABLE I. DC SERIES MOTOR PARAMETERS

Parameters	Values
R	1 Ω
L	0.05 H
M	0.027 H
J	0.5 Kgm^2

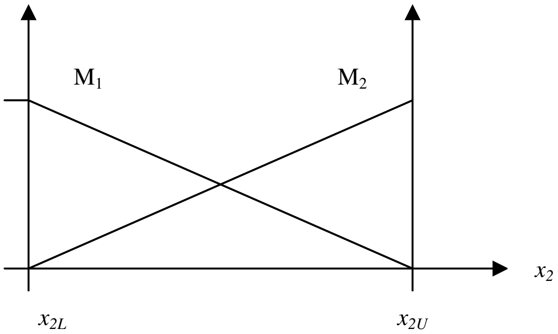


Figure 1. Fuzzy sets for the premise variable

$$\mathbf{A}_1 = \begin{bmatrix} 0 & \frac{M}{J} x_{2L} \\ -\frac{M}{L} x_{2L} & -\frac{R}{L} \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}, \mathbf{E}_1 = \begin{bmatrix} -\frac{1}{J} \\ 0 \end{bmatrix}$$
$$\mathbf{A}_2 = \begin{bmatrix} 0 & \frac{M}{J} x_{2U} \\ -\frac{M}{L} x_{2U} & -\frac{R}{L} \end{bmatrix}, \mathbf{B}_2 = \mathbf{B}_1, \mathbf{E}_2 = \mathbf{E}_1 \tag{12}$$

Based on (11)-(12), the two rule TS-fuzzy model for the nonlinear DC series motor model (8) can be constructed as:

Rule 1: IF $x_2(t)$ is M_1 THEN $\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x} + \mathbf{B}_1 u + \mathbf{E}_1 d$

Rule 2: IF $x_2(t)$ is M_2 THEN $\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x} + \mathbf{B}_2 u + \mathbf{E}_2 d$

The net TS-fuzzy model is then given as:

$$\dot{\mathbf{x}} = \sum_{i=1}^{2} h_i(x_2(t))(\mathbf{A}_i\mathbf{x} + \mathbf{B}_i u + \mathbf{E}_i d) \tag{13}$$

where, $h_i(x_2(t))$ is the normalized firing strength of i^{th} rule.

$$h_i(x_2(t)) = \frac{w_i(x_2(t))}{\sum_{i=1}^{2} w_i(x_2(t))} \tag{14}$$

where, $w_i(x_2(t))$ is the firing strength of i^{th} rule.

$$w_i(x_2(t)) \geq 0, \sum_{i=1}^{2} w_i(x_2(t)) = 1 \tag{15}$$

III. TS FUZZY CONTROLLER

The control objective here is to design a controller ‘ u ’that will be able to maintain the reference speed ‘ ω_{ref} ’ by rejecting the jumping load torque disturbances. The employed control structure is shown in Fig. 2. From the integral controller in Fig. 2, we can write the error dynamics as:

$$\dot{\xi} = \omega_{ref} - y = -\mathbf{C}\mathbf{x} + \omega_{ref} \tag{16}$$

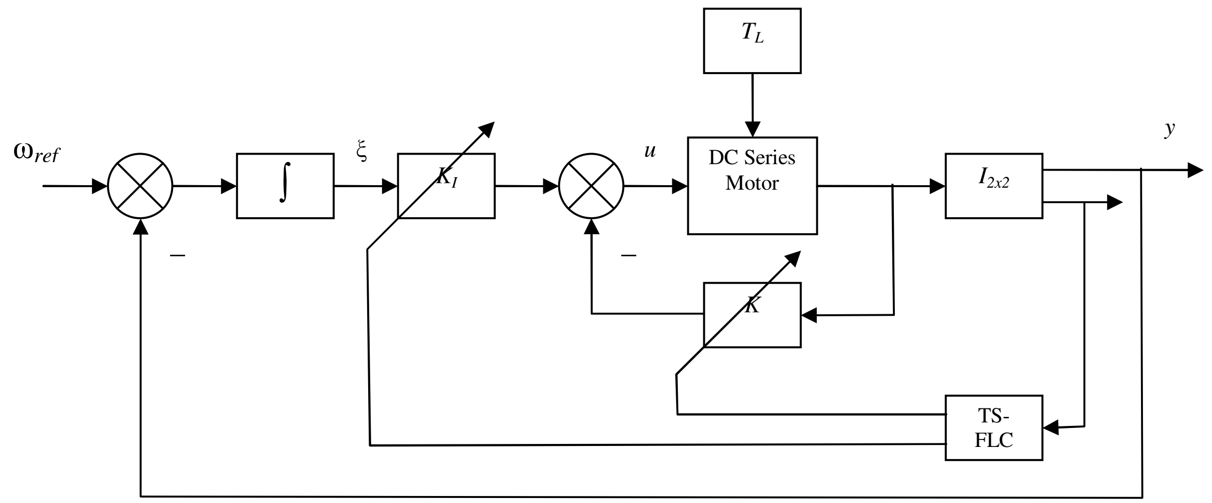


Figure 2. Fuzzy LMI integral controller

Combining (11) and (16), the augmented system dynamics can be given as:

$$\begin{bmatrix} \dot{\mathbf{x}}_i \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \xi \end{bmatrix} + \begin{bmatrix} \mathbf{B}_i \\ 0 \end{bmatrix} u + \begin{bmatrix} \mathbf{E}_i \\ 0 \end{bmatrix} d + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \omega_{ref} \qquad (17)$$

$$y = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \xi \end{bmatrix}$$

The general form of (17) is given as:

$$\dot{\tilde{\mathbf{x}}}_i = \tilde{\mathbf{A}}_i \tilde{\mathbf{x}} + \tilde{\mathbf{B}}_i u + \tilde{\mathbf{E}}_i d + \tilde{\mathbf{F}} \omega_{ref} \qquad (18)$$

The control rules for the augmented TS-fuzzy model (17) can be formulated as:Rule 1: IF $x_2(t)$ is M_1 THEN $u(t) = -\tilde{\mathbf{K}}_1 \tilde{\mathbf{x}}(t)$

Rule 2: IF $x_2(t)$ is M_2THEN $u(t) = -\tilde{\mathbf{K}}_2 \tilde{\mathbf{x}}(t)$

where, $\tilde{\mathbf{K}}_i = \begin{bmatrix} \mathbf{K}^i & -K_I{}^i \end{bmatrix}$ is the control gain for i^{th} rule.

The overall TS-fuzzy control law will be given as:

$$u(t) = -\sum_{i=1}^{2} h_i\left(x_2(t)\right) \tilde{\mathbf{K}}_i \tilde{\mathbf{x}}(t) \qquad (19)$$

In order to find the control gains '$\tilde{\mathbf{K}}_i$' for the local fuzzy models, we impose certain performance constraints [15]. First, the controller should be able to minimize the effect of jumping load torque disturbances on the speed profile of the DC series motor i.e., 'γ'in the following inequality needs to be minimized:

$$\sup_{\|d(t)\|_2 \neq 0} \frac{\|y(t)\|_2}{\|d(t)\|_2} \leq \gamma \qquad (20)$$

Second, the controller should provide good transient performance for disturbance rejection and reference tracking tasks. Thus, the states should decay at a rate 'α' which requires the derivative, $\dot{V}\left(\tilde{\mathbf{x}}(t)\right)$ of quadratic lyapunov function, $V\left(\tilde{\mathbf{x}}(t)\right) = \tilde{\mathbf{x}}^T \mathbf{P} \tilde{\mathbf{x}}$ to satisfy the following inequality:

$$\dot{V}\left(\tilde{\mathbf{x}}(t)\right) < -2\alpha V\left(\tilde{\mathbf{x}}(t)\right) \qquad (21)$$

Third, the control input should be realizable in real time, i.e.:

$$\|u(t)\|_2 \leq \mu \qquad (22)$$

For the design constraints in (20)-(22), the inequalities in (23)-(25) should hold for a symmetric positive definite matrix, $P > 0$.

$$\begin{bmatrix} -\frac{1}{2}\left(\tilde{\mathbf{A}}_{c,ij}{}^T \mathbf{P} + \mathbf{P}\tilde{\mathbf{A}}_{c,ij} + \tilde{\mathbf{A}}_{c,ji}{}^T \mathbf{P} + \mathbf{P}\tilde{\mathbf{A}}_{c,ji}\right) & * & * \\ -\frac{1}{2}\left(\tilde{\mathbf{E}}_i + \tilde{\mathbf{E}}_j\right)^T \mathbf{P} & \gamma^2 \mathbf{I} & \mathbf{0} \\ \frac{1}{2}\left(\tilde{\mathbf{C}}_i + \tilde{\mathbf{C}}_j\right) & \mathbf{0} & \mathbf{I} \end{bmatrix} \geq 0 \qquad (23)$$

$$\tilde{\mathbf{A}}_{c,ii}{}^T \mathbf{P} + \mathbf{P}\tilde{\mathbf{A}}_{c,ii} + 2\alpha\mathbf{P} < 0, \forall i$$

$$\frac{\left(\tilde{\mathbf{A}}_{c,ij} + \tilde{\mathbf{A}}_{c,ji}\right)^T}{2}\mathbf{P} + \mathbf{P}\frac{\left(\tilde{\mathbf{A}}_{c,ij} + \tilde{\mathbf{A}}_{c,ji}\right)}{2} + 2\alpha\mathbf{P} \leq 0, i < j \qquad (24)$$

$$\begin{bmatrix} 1 & \mathbf{x}(0)^T \\ \mathbf{x}(0) & \mathbf{P} \end{bmatrix} \geq 0, \begin{bmatrix} \mathbf{P} & \mathbf{P}\tilde{\mathbf{K}}_i^T \\ \tilde{\mathbf{K}}_i \mathbf{P} & \mu^2 \mathbf{I} \end{bmatrix} \geq 0 \qquad (25)$$

where, '$*$'denotes the transposed entry, and

$$\tilde{\mathbf{A}}_{c,ij} = \tilde{\mathbf{A}}_i - \tilde{\mathbf{B}}_i \tilde{\mathbf{K}}_j \qquad (26)$$

The inequalities in (23)-(25) are not LMIs. However, they can be recast as LMIs through congruence transformation (which preserves the definiteness property) and intermediate matrix variables. By pre- and post-multiplying (23)-(24) with matrices, $\mathbf{X} = diag\left\{\mathbf{P}^{-1}, \mathbf{I}, \mathbf{I}\right\}$, $\mathbf{Y} = \mathbf{P}^{-1}$,respectively and by defining $\mathbf{P} = \mathbf{P}^{-1}$, $\tilde{\mathbf{Q}}_i = \tilde{\mathbf{K}}_i\,\mathbf{P}$,the control gains can be determined as:

$$\tilde{\mathbf{K}}_i = \tilde{\mathbf{Q}}_i\,\mathbf{P}^{-1} \qquad (27)$$

By considering, $x_{2L} = 20$, $x_{2U} = 200$, $\gamma = 0.2$, $\alpha = 5$, $\mu = 230$;the following control gains and symmetric positive definite matrix are obtained using LMI toolbox of MATLAB:

$$\tilde{\mathbf{K}}_1 = \begin{bmatrix} 66.8561 & 4.9888 & -350.7599 \end{bmatrix} \qquad (28)$$

$$\tilde{\mathbf{K}}_2 = \begin{bmatrix} 57.6915 & 6.0023 & -332.9448 \end{bmatrix} \qquad (29)$$

$$\mathbf{P} = \begin{bmatrix} 0.0421 & -0.2594 & 0.0034 \\ -0.2594 & 4.1790 & -0.0006 \\ 0.0034 & -0.0006 & 0.0006 \end{bmatrix} \times 10^3 \qquad (30)$$

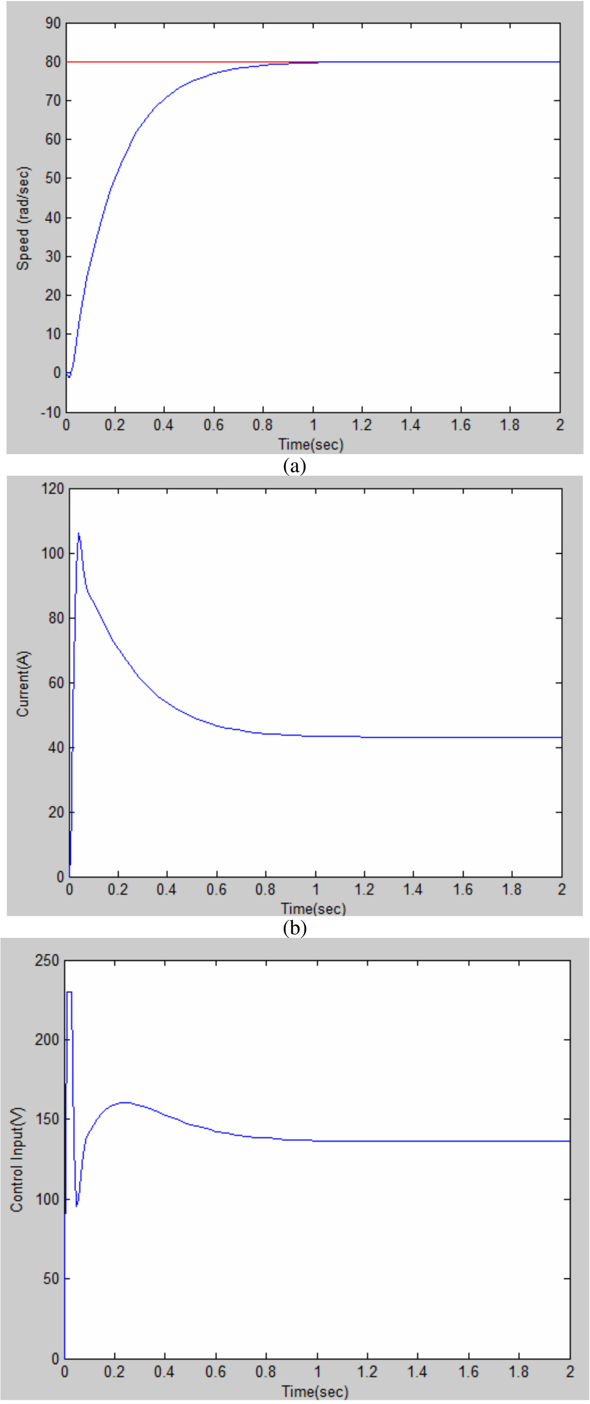
IV. Results

The designed controller is simulated in MATLAB/Simulink environment for various speed references and load torques. For a reference speed of 80rad/sec and load torque of 50Nm, the simulation results are shown in Fig. 3. With the same load torque applied, the controller is made to track a set of reference speeds ({90,100,110} rad/sec) while motor is already running at 80rad/sec. The results for this case are shown in Fig. 4. Then a sequence of jumping load torques ({60,70,80}Nm) are applied while the motor is running at reference speed of 100 rad/sec with initial load torque of 50Nm. The controller is able to reject the load torque disturbances as evident from Fig. 5.

From the simulation results, the settling time for the controller is found to be 0.72 sec. The deviation from the reference speed is recorded to be 0.36 rad/sec when the load torque disturbance is applied in steady state and is rejected in 0.64 sec by the two rule fuzzy controller. The ripples in the current are limited to ± 0.4A of the actual value during steady state.

The ripples in the current and voltage can be reduced by employing a larger rule base controller which implies the need of using more membership functions covering the universe of discourse.

The proposed TS FLC is also compared with pole placement controller (PPC). The pole placement controller is designed based on averaged system model for the same transient specifications ($T_s = 0.72s, \xi = 1.0$). The second order desired characteristic equation is therefore formed as: $s^2 + 11.1s + 30.8025 = 0$. The third pole is placed 20 times farther than the dominant closed loop poles. The comparison of TS-FLC and PPC for a reference speed of 50rad/sec is shown in Fig. 7. It can be observed that PPC shows both undershoot and overshoot while tracking the set speed. However, no overshoot is seen in case of TS-FLC. The rejection of load torque disturbances ({60,70,80}Nm) by both the controllers is shown in Fig. 7 while the motor is running at a reference speed of 50 rad/sec. The deviation in speed is found to be less in case of TS-FLC as compared to PPC.

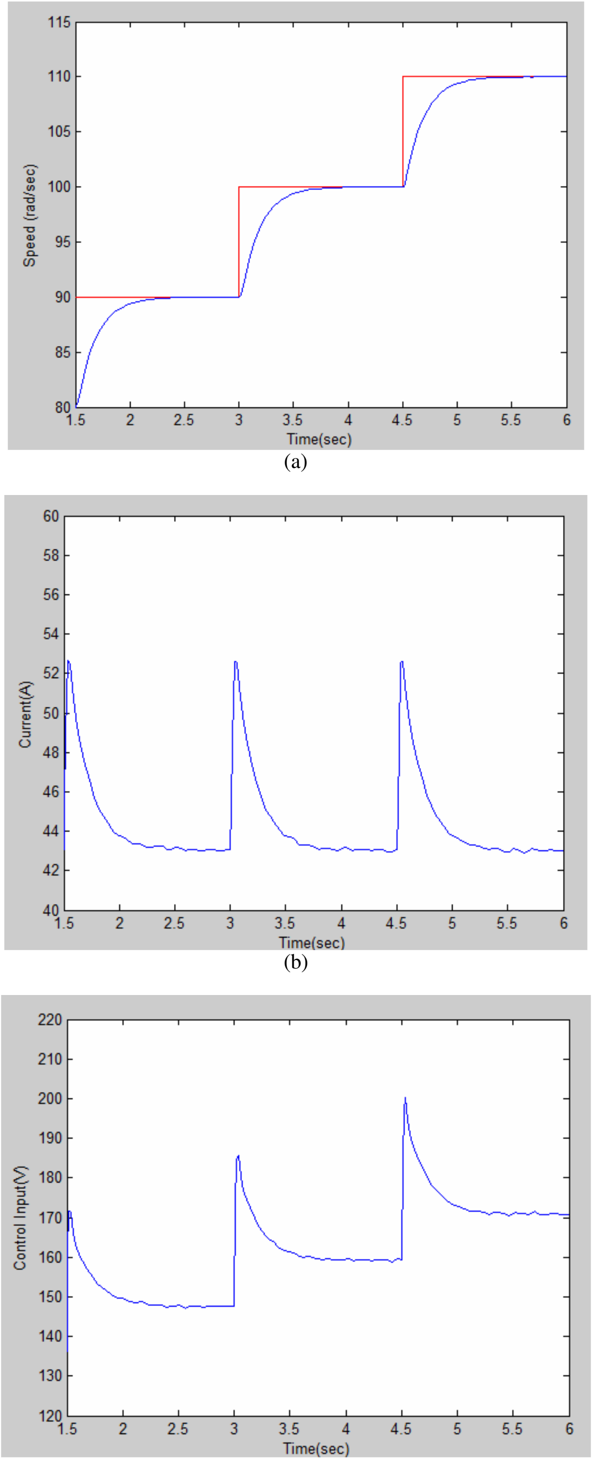


(a)

(b)

(c)

Figure 3. Simulation results for ω_{ref} = 80 rad/s and T_L=50 Nm

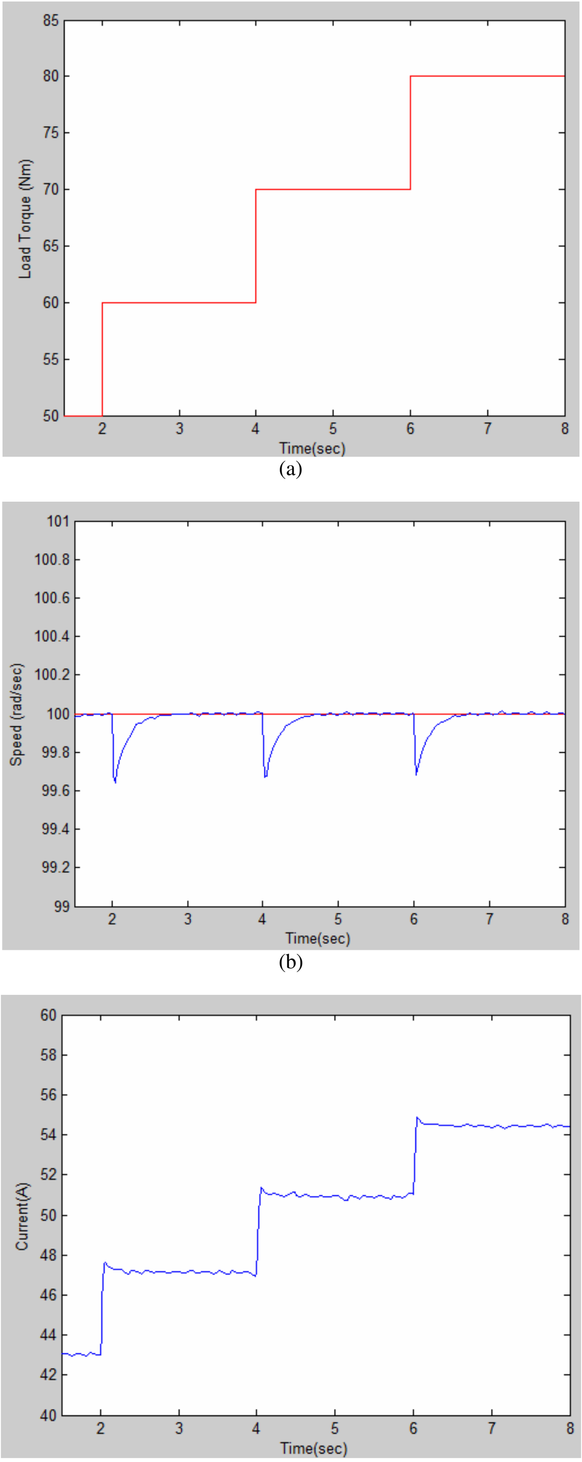


(a)

(b)

(c)

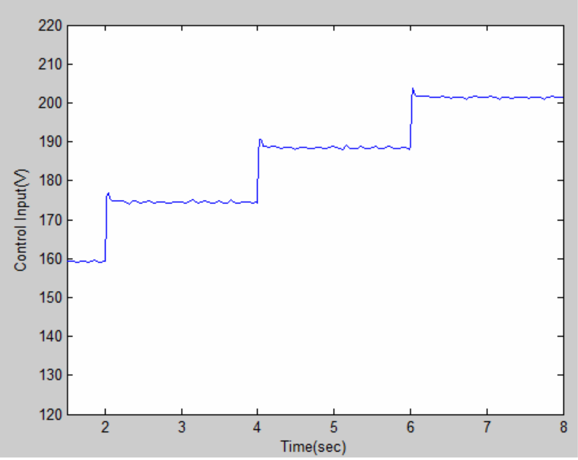
Figure 4. Simulation results for ω_{ref} = {90,100,110} rad/s and T_L=50 Nm



(a)

(b)

(c)



(d)

Figure 5. Simulation results for ω_{ref} = 100 rad/s and T_L= {60,70,80} Nm

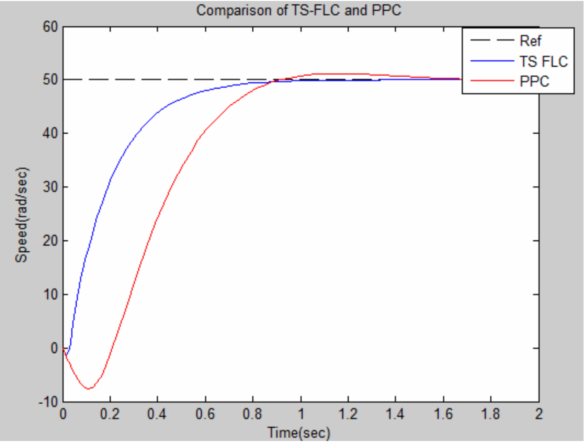


Figure 6. Comparison results for ω_{ref} = 50 rad/s and T_L=50 Nm

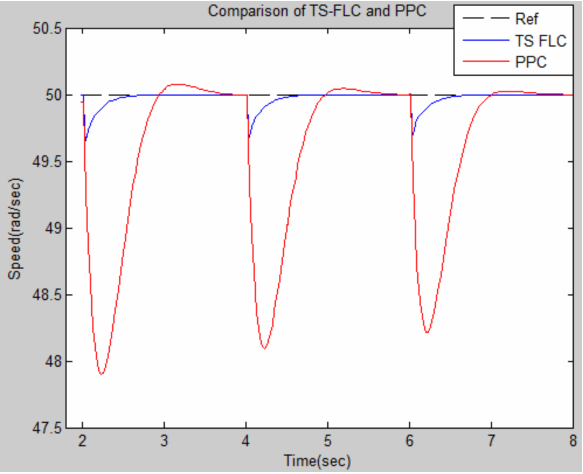


Figure 7. Comparison results for ω_{ref} = 50 rad/s and T_L= {60,70,80} Nm

V. Conclusions

A simple two rule TS fuzzy model of the DC series motor is constructed followed by a two rule integral controller which shares the same premise membership functions as that of model. The control gains are determined through a set of LMIs which guarantee the closed loop stability of the system for fuzzy regions. The performance of the controller is evaluated through simulation runs in MATLAB environment for various reference speeds and load torque disturbances. Future work involves the design of type-2 fuzzy logic controller.

References

[1] R. D. Begamudre, Electro-Mechanical Energy Conversion with Dynamics of Machines, New York: Wiley, 1988.

[2] F. R. Fuentes, and G. J. Estrada, “A novel four quadrant DC series motor control drive for traction applications,” Proc. IEEE International Electric Vehicle Conference, 2012, pp. 1-4.

[3] R. J. Hill, “Electric railway traction-I: Electric traction and dc traction motor drives,” Power Engineering Journal, vol. 8, no. 1, 1994.

[4] Samir Mehta, and John Chiasson, “Nonlinear control of a series DC Motor: Theory and Experimeny,” IEEE Transactions on Industrial Electronics, vol. 45, no. 1, Feb 1998, pp.131-141.

[5] Z. Z. Liu, F. L. Luo, M. H. Rashid, “Nonlinear speed controllers for series DC motor,” Proc. IEEE International Conference on Power Electronics and Drive Systems, 1999, pp. 333-338.

[6] Richard Pothin, Claude H. Moog, X. Xia, “Stabilization of a series DC motor by dynamic output feedback,” Lecture Notes in Control and Information Sciences, vol. 259, 2001, pp.257-263.

[7] Dimitrios P. Iracleous, “Series connected DC motor tracking using port controlled hamiltonian systems equivalence,” Proc. WSEAS International Conference on Systems, 2009, pp. 591-595.

[8] M. J. Burridge, and Z. Qu, “An improved nonlinear control design for series DC motors,” Computers and Electrical Engineering, vol. 29, 2003, pp. 730-735.

[9] Dongbo Zhao, and Ning Zhang, “An improved nonlinear speed controller for series DC motors,” Proc. World Congress of International Federation of Automatic Control, July 6-11, 2008, pp. 11047-11053.

[10] M. Muruganandam, and M. Madheswaran, “Experimental verification of chopper fed DC series motor with ANN controller,” Frontier in Electrical and Electronics Engineering, vol. 7, no. 4, 2012, pp. 477-489.

[11] Hasan A. Yousaf, and H. M. Khalil, “A Fuzzy logic based control of series DC motor drives,” Proc. IEEE International Symposium on Industrial Electronics, Jul 10-14, 1995, pp.517-522.

[12] H. L. Tan, N. A. Rahim, W. P. Hew, “A simplified fuzzy logic controller for DC series motor with improved performance,” Proc. IEEE International Conference on Fuzzy Systems, 2001, pp. 1523-1526.

[13] Majed Jabri, Houda Chouiref, Houssem Jebri, Naceur Benhadj Braiek, “Fuzzy logic parameter estimation of an electrical system,” Proc. International Multi-Conference on Systems, Signals and Devices, 2008, pp. 1-6.

[14] B. Eskandari, H. Valizadeh Haghi, M. Tavakoli Bina, M. A. Golkar, “An experimental prototype of buck converter fed series dc motor implementing speed and current controls,” Proc. International Conference on Computer Applications and Industrial Electronics, Dec 5-7, 2010, pp. 606-609.

[15] K. Tanaka, and H. O. Wang, Fuzzy control systems design and analysis: A linear matrix inequality approach, John Wiey & Sons, Inc., 2001.

Rice-Planted Area Detection by Using Self-Organizing Feature Map

Sigeru Omatu

Department of of Electronics, Information, and
Communication Engineering
Osaka Institute of Technology
Osaka, JAPAN 535–8585
Email: omtsgr@gmail.com

Mitsuaki Yano

Department of of Electronics, Information, and
Communication Engineering
Osaka Institute of Technology
Osaka, JAPAN 535–8585
Email: yano@elc.oit.ac.jp

***Abstract*—This paper considers a classification of estimation of rice planted area by using remote sensing data. The classification method is based on a competitive neural network and the sattelite data are remote sensing data observed before and after planting rice in 1999 in Hiroshima, Japan. Three RADAR Satellite (RADARSAT) and one Satellite Pour l'Observation de la Terre(SPOT)/High Resolution Visible (HRV) data are used to extract rice-planted area. Synthetic Aperture Radar (SAR) back-scattering intensity in rice-planted area decreases from April to May and increases from May to June. Thus, three RADARSAT images from April to June are used in this study. The SOM classification was applied the RADARSAT and SPOT to evaluate the rice-planted area estimation. It is shown that the Self-Organizing feature Map (SOM) of competitive neural networks is useful for the classification of the satellite data by SAR to estimate the rice planted area.**

Keywords–Remote Sensing; RADAR Satellite; Synthetic Aperture Radar; Self-Organizing Feature Map

I. Introduction

We have noticed that rice is the most important agricultural product and widely planted in the wide area in Japan. But, it is still difficult to estimate rice-planted areas every year. Therefore, the development of a system for monitoring the rice crop will be prefarable. Satellite remote sensing images by optical sensors like LANDSAT TM or SPOT HRV, have been used to estimate a rice-planted area. However, these optical sensors have been unable to get necessary data at a suitable time because it is often cloudy or rainy during the rice planting season in Japan [1][2][3][4].

On the other hand, SAR penetrates through the cloud covered. Hence, it can observe the land surface under all weather conditions [5][6][7][8]. The back-scattering intensity of C-band SAR images, such as RADARSAT or ERS1/SAR, changes greatly from non-cultivated bare soil condition before rice planting to inundated condition just after rice planting [1]. In addition, RADARSAT images are rather sensitive to the change of rice biomass in a growing period of rice [9][10].

Thus, a rice area estimation is expected to be achieved in an early stage. In previous works [2][3][4], we attempted to estimate rice-planted area using RADARSAT fine-mode data in an early stage. The estimation accuracy of a rice-planted area by Maximum Lkelihhood Method (MLH) was approximately 40% by comparing with the estimated area by SPOT multi-spectral data. In this study, we attempt to detect the rice-planted area from RADARSAT data using SOM that is the unsupervised classification [11].

In Section II, test site and remote sensing data used here will be explained and in Section III, data correction of geometric distortion will be shown. In Section IV, SOM algorithm will be explained and in Section V, evaluation indexes will be introduced. In Section VI, training results will be shown and in Section VII, selection of training iteration will be discussed. After that, in Section VIII, classification results will be shown and in Section IX, experimental results will be shown. Finally, in Section X, we will conclude the results and state some future aspects.

II. Test Site and Data

The test area has a size of about 7.5×5.5 km in Higashi-Hiroshima, Japan shown, as a white block in Fig. 1; the enlarged image is shown in Fig. 2. This site is located at the eastern part of Hiroshima, Japan. Three multi-temporal RADARSAT fine-mode (F1F) images, taken on April 8, May 26, and June 19, in 1999 were used as the test data. SPOT/HRV multi- spectral data taken on June 21, 1999 were used to generate a reference image for rice-planted area extraction.

Three merged RADARSAT and one SPOT images in a part of the test site are shown in Figs. 3 and 4. The land surface condition in the rice-planted area of April 8 is a non-cultivated bare soil before rice planting with rather rough soil surface. The surface condition of May 26 is almost smooth water surface just after rice planting, and that of June 19 is a mixed condition of growing rice and water surface. It is found that the rice-planted areas are shown in a dark tone in the RADARSAT image. The RADARSAT raw data were processed using Vexcel SAR Processor (VSARP) and single-look power images with 6.25 meters ground resolution were generated. Then, the images were filtered using median filter with 7×7 moving window. All RADARSAT and SPOT images were overlaid onto the topographic map with 1:25,000 scale. As RADARSAT images are much distorted by foreshortening due to topography, the digital elevation model (DEM) with 50 meters spatial resolution issued by Geographical Survey Institute (GSI) of Japan [2] was used to correct foreshortening of RADARSAT images.

The specifications of RADARSAT-1/SAR fine mode parameters and SPOT-2/HRV parameters are given by Table I and Table II, respectively.

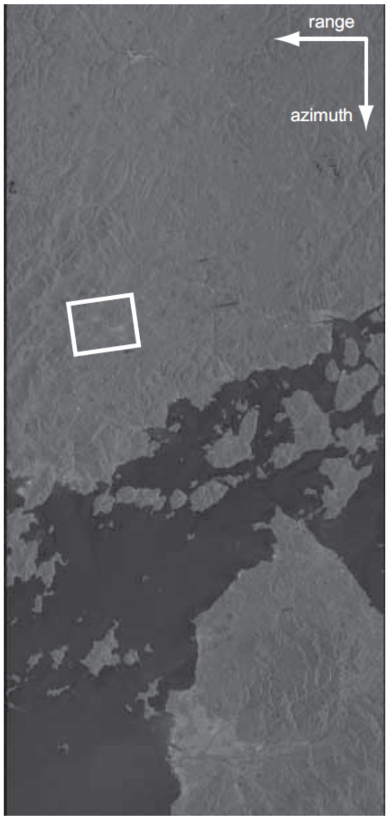


Figure 1. RADARSAT fine mode image of Higashi-Hiroshima, Japan where the area enclosed by a white line denotes the analised area.

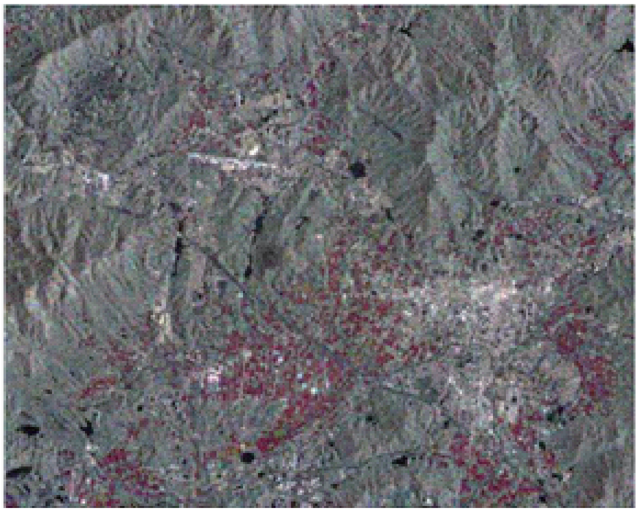


Figure 2. RADARSAT fine mode image of Higashi-Hiroshima where the area enclosed by a white line in Fig. 1 is enlarged..

III. Data correction of geometric distortion

SAR or SPOT data observe the land surface from an oblique position as shown in Fig. 5. In Fig. 5, θ denotes incident angles and in case of optical remote sensing data, a deformation D at the height h is given by $D = h \tan\theta$. In case of SAR remote sensing data, it becomes $D = h/\tan\theta$. Therefore, we must correct the remote sensing data according to the hight of the target pixcel. We adopt the mesh data published at GSI of Japan[2].

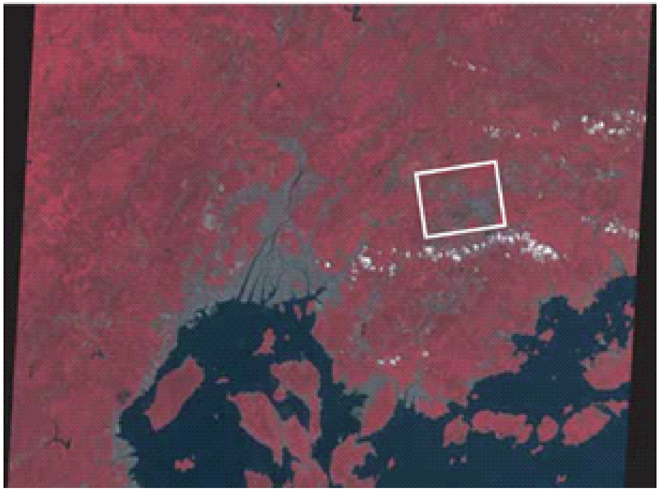


Figure 3. SPOT-2/HRV image(1999/6/21, R:Band 3, G:Band 2, B:Band 1) where the area enclosed by a white line denotes the test area 息 CNESS 1999.

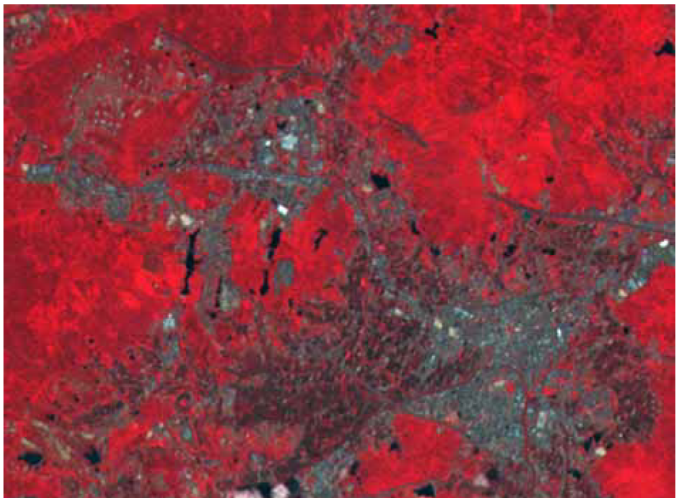


Figure 4. SPOT-2/HRV image(1999/6/21, R:Band 3, G:Band 2, B:Band 1) in the test site 息 CNESS 1999.

TABLE I. The parameters of RADARSAT-1/SAR fine mode parameters

Launching agency	Canadian Space Agency
Launching date	1995/11/04
Attitude	798km
Repeat cycle	24 days
Frequency	53GHz
Polarization	HH
Resolution	8m
Observation range	45km

TABLE II. The parameters of SPOT-2/HRV parameters

Launching agency	CNES
Launching date	202/05/4
Attitude	822km
Repeat cycle	26 days
Band width	Multi-spectral mode XS1 0.50-0.59 μ m XS2 0.61-0.68 μ m XS3 0.79-0.89 μ m
Resolution	5m
Observation range	60km

$$
\begin{aligned}
D_P &= a_1P + a_2L + a_3h + a_4 && (1)\\
D_L &= b_1P + b_2L + b_3h + b_4. && (2)
\end{aligned}
$$

Here, $D_P$and $D_L$are distortions of pixel and line directions, P and L denote coordinate values of pixel and line directions, respectively. Furthermore, h is the hight of area and $a_i, i = 1, 2, 3, 4$ and $b_i, i = 1, 2, 3, 4$ are regression coefficients.

IV. SOM ALGORITHM

We briefly summarize the algorithm for SOM. The structure of SOM consists of two layers. One is an input layer and the other is a competitive layer. The total input to a neuron j is denoted by net_j and modeled by the following equations:

$$
\begin{aligned}
\text{net}_j &= \sum_i^n w_{ji}x_i = (w_{j1}, w_{j2}, \ldots, w_{jn})(x_1, x_2, \ldots, x_n)^t \\
&= W_jX^t && (3)\\
W_j &= (w_{j1}, w_{j2}, \ldots, w_{jn}),\ X = (x_1, x_2, \ldots, x_n) && (4)
\end{aligned}
$$

where $(\cdot)^t$denotes the transpose of $(\cdot)$ and x_i and w_{ji} show an input in the input layer and a weighting function from the input x_i to a neuron j in the competitive layer, respectively. For simplicity, we assume that the norms of X and W_j are equalt to one, that is,

$$\|x\| = 1, \quad \|W_j\| = 1, j = 1, 2, \ldots, N \qquad (5)$$

where $\|\cdot\|$ shows Eucridean norm and N denotes the total number of neurons in the competitive layer.

When an input vector X is applied to the input layer, we find the nearest neighboring weight vector W_c to the input vector X such that

$$\|W_c - X\| = \min_i \|W_i - X\|. \qquad (6)$$

The neuron c corresponding to the weight vector W_c is called a winner neuron. We select neighborhood neurons within the distance d which are shown in Fig. 6 and a set of indices for neurons located in the neighborhood of c is denoted by N_c. Then, the weighting vectors of the neurons contained in N_c are changed such that those weighting vectors could become similar to the input vector X as close as possible. In other words, the weighting vectors are adjusted as follows:

$$
\begin{aligned}
\Delta W_j &= \eta(t)(W_j - X) \qquad \forall j \in N_c && (7)\\
\Delta W_j &= 0 \qquad \forall j \notin N_c && (8)
\end{aligned}
$$

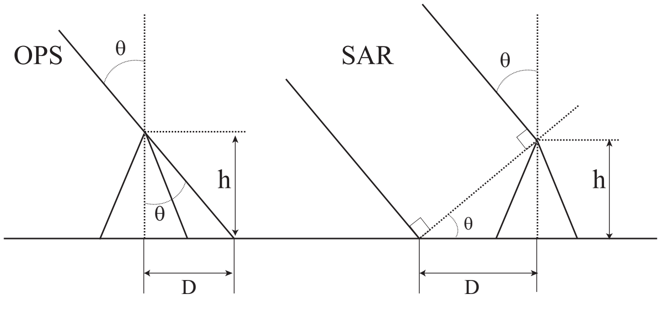


Figure 5. The principle of distortion of remote sensing data depending on the hights in case of optical sensory data and SAR data.

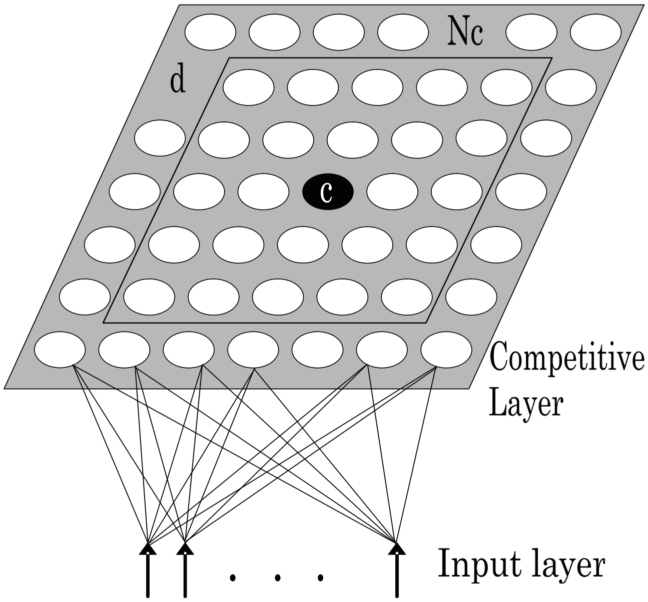


Figure 6. SOM structure of the neural network.

where

$$
\begin{aligned}
\Delta W_j &= W_j(\text{new}) - W_j(\text{old}) && (9)\\
\eta(t) &= \eta_0(1 - \frac{t}{T}), \qquad d(t) = d_0(1 - \frac{t}{T}). && (10)
\end{aligned}
$$

Here, t and T denote an iteration number and the total iteration number for learning, respectively. η_0 and d_0 are positive and denote initial values of $\eta(t)$ and $d(t)$, respectively where $d(t)$ denotes an Euclid distance from the winner neuron c.

V. EVALUATION OF SOM

We will introduce two criteria, namely, precision and recall to find a suitable SOM. The precision is defined by

$$P_r = \frac{R}{N} \times 100(\%) \qquad (11)$$

and the recall is defined by

$$R_r = \frac{R}{C} \times 100(\%) \qquad (12)$$

where P_r and R_r denote the precision rate and the recall rate, respectively and N is a trial number, R is a correct classified number, and C is a total correct number. As shown in Fig. 7, if we try to increase P_r, then, R_r will decrease. Therefore, we adopt a criterion f-measure defined by

$$f_m = \frac{2 \times P_r \times R_r}{P_r + R_r} \times 100(\%). \qquad (13)$$

VI. TRAINING OF SOM

In order to find a suitable size of competitive layer, we assume map size m as 3×3, 4×4, 5×5, 6×6, and 7×7. We take an initila learning rate α_0=0.2 and an initial value of neighborhood d_0=2. Furthermore, we assume initial values of connection weights as random numbers of [0.1, 0.9] and itelation number t=5. The training results of SOM are shown

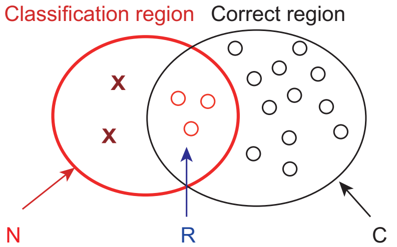


Figure 7. Classification region for the precision and the recall.

in Fig. 8 and Fig. 9 for SPOT remote sensing data and RADARSAT remote sensing data, respectively.

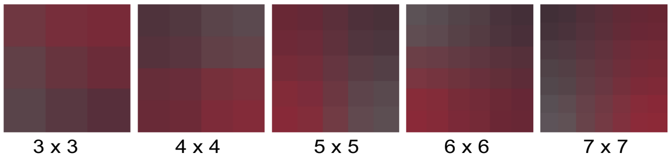


Figure 8. Learning results for SPOT remote sensing data.

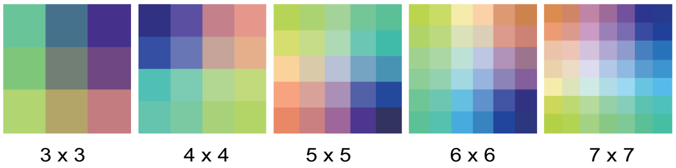


Figure 9. Learning results for RADARSAT remote sensing data

After taining the neural network of SOM, we calculate the P_r, R_r, and f_m for SPOT and RADARSAT remote sensing data. The results are shown in Table III and Table IV, respectively. From Table III, we can see that the classification result of the map size 4×4 is highest value of f_m=84.49. Thus, in case of a small number of category, we can set small number of neurons in the competitive layer. From Table IV, we can see that f_m becomes largest in case of 4×4.

TABLE III. P_r,R_r, AND f_m FOR SIZES OF SPOT REMOTE SENSING DATA

Size of map	P_r	R_r	f_m
$3times3$	98.95	64.67	78.22
$4times4$	100	73.15	84.49
$5times5$	93.62	76.65	84.29
$6times6$	85.65	68.65	76.21
$7times7$	96.16	63.00	76.13

TABLE IV. P_r,R_r, AND f_m FOR SIZES OF RADARSAT REMOTE SENSING DATA

Size of map	P_r	R_r	f_m
$3times3$	63.08	47.72	54.34
$4times4$	60.99	51.38	55.77
$5times5$	62.73	46.41	53.35
$6times6$	60.48	51.02	55.35
$7times7$	47.63	64.70	54.87

VII. Selection of Training Iteration

We consider the iteration t for learning. Here, we count t=1when an input image of 1,800,000 pixels of 1,500pixels per line $\times$1,200 lines has been trained. In this experiment, we take t=1 (1,800,000 training), t=5 (9,000,000 training), t=10 (18,000,000 training), t=20 (36,000,000 training). Except for t, we take a size of neurons in the competitive layer as 4×4, α_0=0.2, and an initial vector W_0 of weighting functions are random numbers of [0.1, 0.9]. The results are shown in Table V and Table VI for SPOT and RADARSAT remote sensinr data, respectively.

TABLE V. P_r, R_r, AND f_m FOR ITERATIONS OF SPOT

Iteration	P_r	R_r	f_m
1	100	64.67	78.22
5	100	73.15	84.49
10	100	76.65	84.29
20	100	68.65	76.21

TABLE VI. THE VALUES OF P_r,R_r, AND f_m FOR ITERATIONS OF RADARSAT

Iteration	P_r	R_r	f_m
1	58.40	50.29	54.04
5	60.99	51.38	55.77
10	61.00	51.31	55.74
20	61.01	51.34	55.76

From Table V, we can see the maximum value of f_m=84.55 is obtained when t=5 for SPOT remote sensing data and from Table VI, the maximum value of f_m=55.77 is obtained when t=5. Therefore, we set t=5 in what follows.

VIII. Classification Results

A rice-planted area was extracted by using SOM from three temporal RADARSAT images and one SPOT image. SOM is a classification method based on competitive neural networks without teacher. It was applied to the remote sensing data by using the parameters as shown in Table VII. RADARSAT and SPOT images were classified into 16 categories by SOM. Then, we labeled the categories into a rice-planted area, a forest area and an urban area.

IX. Experimental Results and Discussion

In order to make the proposed method effective, we classify the satellite image data by SOM. Fig. 10 shows the classification image by SPOT and Fig. 11 shows the classification image by RADARSAT. By comparing Fig. 10 with Fig. 11, one can see that the rice-planted area by RADARSAT was extracted less than that by SPOT. The speckle noise was still seen in the image of rice-planted area, the majority filter with 7×7 window was applied to the rice extracted images by RADARSAT and SPOT. For the evaluation of the rice-planted area extraction, we defined two indices, True Production Rate (TPR) and False Production Rate (FPR). The TPR and FPR are calculated by

$$TPR = \frac{\alpha}{\alpha+\beta} \times 100 \tag{14}$$

$$FPR = \frac{\gamma}{\alpha+\gamma} \times 100 \tag{15}$$

where α means the number of relevant rice-planted area extracted, β means the number of relevant rice-planted area not

TABLE VII. THE PARAMETERS OF SOM

Neurons	Training iterations	η_0	d_0
4 $\times$ 4	1080,000	0.03	2

extracted, and γmeans the number of irrelevant rice-planted area extracted. Extraction rice-planted area of SPOT image by supervised MLH was used as reference rice-planted area image.

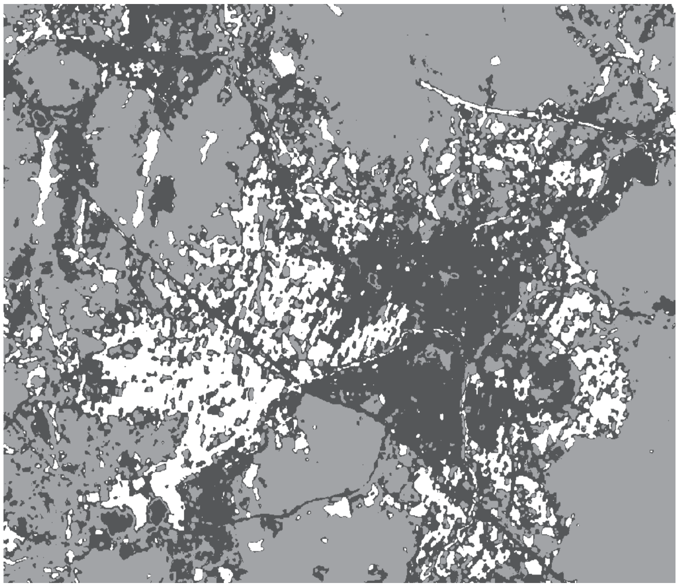


Figure 10. Classification result of SPOT image by SOM (White: rice, Light gray: forest, Dark gray: urban).

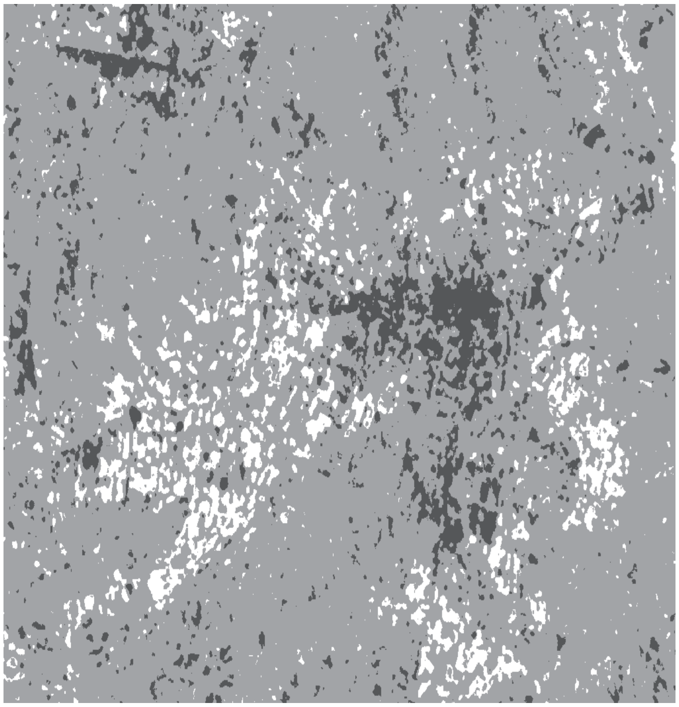


Figure 11. Classification result of RADARSAT image by SOM (White: rice, Light gray: forest, Dark gray: urban).

Table VIII shows the results of TPR and FPR by SOM and MLH classification for SPOT data and RADARSAT data. From this results we can see that in case of RADARSAT data, the values of TPR (extraction rate of rice-planted area) are, 50.97% and 60.96% for MLH and SOM, respectively. This means that SOM is better than MLH for extraction of rice-planted area by about 10%. As for FPR (misclassification rate) SOM is also better than MLH by about 3%. The SPOT data is very fine images like 5m per pixel and we can assume the image of SPOT reflects almost all land surface. Using SPOT data TPR and FPR are 70.41% and 21.51%, respectively. Thus, we could extract rice-planted area a certain level in practice fron Table VIII.

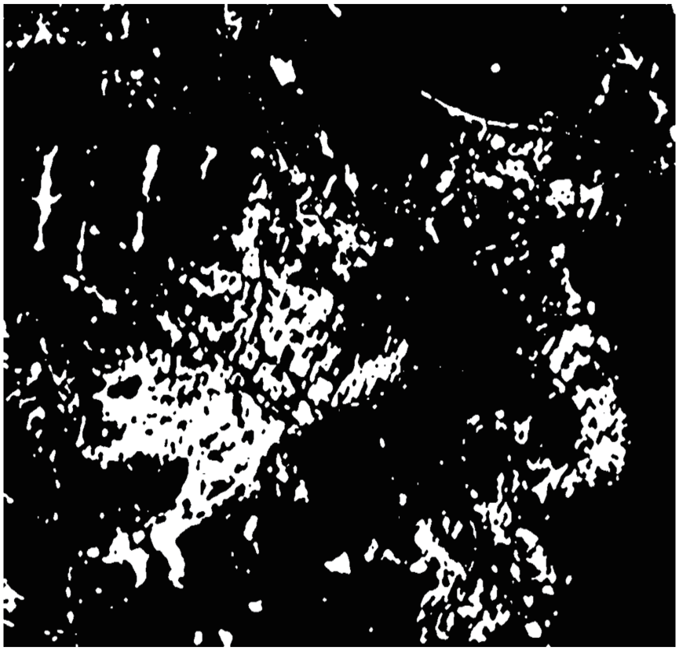


Figure 12. Extraction result of rice-planted area by SPOT data.

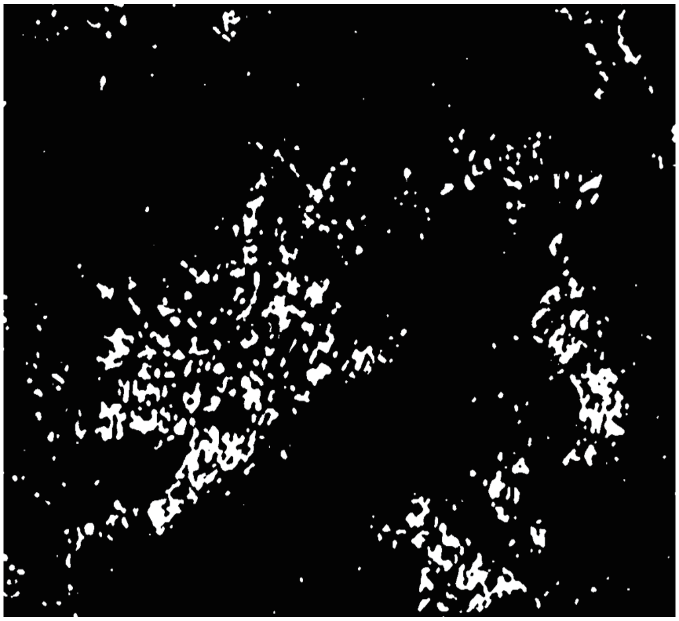


Figure 13. Extraction result of rice-planted area by multi temporal RADARSAT data.

Figs. 12 and 13 show the extraction result of rice-planted area by SPOT and RADARSAT, respectively. The results show that fine rice-planted area could be extracted by using SOM compared with MLH from there figures.

TABLE VIII. THE RESULTS OF TPR AND FPR FOR RICE-PLANTED AREA EVALUATION

Comparison data	Classification method	TPR(%)	FPR(%)
SPOT	SOM	70.4	21.51
RADARSAT	MLH	50.97	51.44
RADARSAT	SOM	60.97	48.59

X. Conclusion and Future Work

Rice-planted area extraction was attempted using multi-temporal RADARSAT data taken in an early stage of rice growing season by SOM classifications. The SOM is unsupervised classification whose computational time is shorter than the other supervised classification method like MLH [3] or LVQ [4][11]. We have been engaged with image analysis on remote sensing data analysis. Especially, we are concentrated on SAR data analysis to land cover classification by using neural network classification methods. Since the SAR data is completely different with optical sensor images, it is difficult to obtain usual land cover map although it can observe the earth at any time and under any weather conditions such as rainy or cloudy occasion. Thus, we must develop the special attention to get suitable classification results. In this paper, we have developed a method to allocate seasonal data to get the color image and show the change of rice field area in a visual way. It takes much time for extraction of features of SAR images, we must speed up the classification. we are now developing nearest neighbor algorithm. As for this results, we will show the results near future. The future study, we will apply this proposed and other neural network method to other SAR data due to the extraction rice-planted area. As we could obtain more high resolution images, we are checking the preset results for those big data and trying to make senario for them.

Acknowledgment

This work was supported by JSPS KAKENHI Grant-in-Aid for Challenging Exploratory Research (25630180). The authors would like to thank JSPS to support this research work.

References

[1] Y. Suga, Y. Oguro, and S. Takeuchi, "Comparison of Various SAR Data for Vegetation Analysis over Hiroshima City", *Advanced Space Research*, vol. 23, August, 1999, pp. 225–230.

[2] Y. Suga, S. Takeuchi, and Y. Oguro, "Monitoring of Rice-Planted Areas Using Space-borne SAR Data", *Proceedings of IAPRS*, XXXIII, B7, February, 2000, pp. 741–743.

[3] S. Omatu, "Rice-Planted Areas Extraction by RADARSAT Data Using Learning Vector Quantization Algorithm", *Proceedings of ADVCOMP 2013*, Sep., 2013, pp. 19–233.

[4] T. Konishi, S. Omatu, and Y. Suga, "Extraction of Rice-Planted Areas Using a Self-Organizing Feature Map ", *Artificial Life and Robotics*, 2007, vol. 11, pp. 215–218.

[5] I. H. Woodhouse, *Introduction to Microwave Remote Sensing*. CRC Press, London, Nov. 2005, ISBN: 0-415-27123-1.

[6] C. M. Ryan, T. Hill, E. Woollen, C. Ghee, E. Mitchard, G. Cassells, J. Grace, I. H. Woodhouse, and M. Williams, "Quantifying Small-Scale Deforestation and Forest Degradation in African Woodlands Using Radar Imagery ", *Global Change Biology*, vol. 18, pp. 243–257, 2012 ISSN: 1365-2486.

[7] E. T. A. Mitchard, S. S. Saatchi, L. J. T. White, K. A. Abernethy, K. J.Jeffery, S. L.Lewis, M. Collins, M. A.Lefsky, M. E. Leal, I. H. Woodhouse, and P. Meir, "Mapping Tropical Forest Biomass with Radar and Spaceborne LiDAR in Lope National Park, Gabon: Overcoming Problems of High Biomass and Persistent Cloud", *Biogeosciences*, vol. 9, pp. 179-191, 2012, doi: 10.5194/bg-9-179-2012.

[8] C. H. Chen, Ed.,*Signal and Image Processing*, London, Jan. 2007, chapter 27, pp. 607–619, M. Yoshioka, T. Fujinaka, and S. Omatu,*SAR Image Classification by Support Vector Machine* , ISBN: 0-8493-5091-3.

[9] M. Bicego and T. L. Toan, "Rice Field Mapping and Monitoring with RADARSAT Data", *International Journal of Remote Sensing*, vol. 20, April, 1999, pp. 745–765.

[10] S. C. Liew, and P. Chen, "Monitoring Changes in Rice Cropping System Using Space-borne SAR Imagery", *Proceedings of IGARSA'99*, Oct., 1999, pp. 741–743.

[11] T. Kohonen, "Self-Organizing Maps", *Springer*, 1997, pp. 206–217, ISBN 3-540-62017-6.

Topological Approach to Image Reconstruction in Electrical Impedance Tomography

Tomasz Rymarczyk, Paweł Tchórzewski
Department of Research and Development
NET-ART
Lublin, Poland
tomasz@rymarczyk.com, pawel.tchorzewski@netrix.com.pl

Jan Sikora
Electrotechnical Institute
Warszawa, Poland
sik59@wp.pl

***Abstract*—In this paper, we investigate the application of the topological derivative in combination with the level set method for the topology optimization. The level set method and the gradient technique are based on shape and topology optimization approach to the Electrical Impedance Tomography problems with piecewise constant conductivities. The Finite Element Method and the Boundary Element Method have been used to solve the forward problem. The cost of our numerical algorithm is moderate since the shape is captured on a fixed mesh. The proposed solution algorithm is initialized by using topological sensitivity analysis. Shape derivatives and topological derivatives have been incorporated with the level set method to investigate shape optimization problems. Then it relies on the notion of shape derivatives to update the shape of the domains where conductivity takes different values. The shape derivative measures the sensitivity of boundary perturbations while the topological derivative measures the sensitivity of creating a small object in the interior domain. The coupled algorithm is a relatively new procedure to overcome this problem.**

Keywords-Image Reconstruction; Inverse Problem; Level Set Method; Optimization Methods.

I. INTRODUCTION

Numerical methods of the shape and the topology optimization were based on the level set representation and the shape differentiation [8][10]. Level set methods have been applied very successfully in many areas of the scientific modelling, for example in propagating fronts and interfaces [6][7][12][13]. Therefore, they are used to study shape optimization problems. Instead of using the physically driven velocity, the level set method typically moves the surfaces by the gradient flow of an so-called energy functional [1]. These approaches based on shape sensitivity include the elastic boundary design. There are two features that make these methods suitable for the topology optimization. The structure is represented by an implicit function such that its zero level set defines the boundary of the object. This function is often discretized on a regular grid that conveniently coincides with the finite or boundary element mesh used for structural analysis. The next valid feature is the simple update of the implicit function using the Hamilton-Jacobi equation [8], where the velocity function is determined by the shape sensitivity of the structure. These properties enable natural topology changes. The discussed technique can be applied to the solution of inverse problems in the Electrical Impedance Tomography [6][7][10][11][14].

In this work, there were implemented the novel algorithms to identify unknown conductivities. The purpose of the presented method is obtaining the better image reconstruction than gradient methods. We also want to accelerate the iterative process by using different shapes of the zero level set functions.

In the second section, we present some information about Electrical Impedance Tomography. In the third section, discussion of numerical methods is given, and in the fourth section, numerical results are shown. The last section contains conclusions.

II. ELECTRICAL IMPEDANCE TOMOGRAPHY

The Electrical Impedance Tomography (EIT) is a non-destructive imaging technique which has various applications. Its purpose is to reconstruct the conductivity of hidden objects inside a medium with the help of boundary field measurements. Efficient algorithms for solving forward and inverse problems have to be developed in order to use this approach for practical tasks. Moreover, it is necessity to improve performance of selected numerical methods. Typical problem in EIT requires the identification of the unknown internal area from near-boundary measurements of the electrical potential. It is assumed that the value of the conductivity is known in subregions whose boundaries are unknown. The forward problem in EIT is described by following partial differential equation:

$$\nabla \cdot (\gamma \nabla u) = 0, \tag{1}$$

where γ denotes conductivity. Symbol u represents electrical potential. Function u is taken under Dirichlet condition [7] in boundary points adjacent to electrodes and Neumann condition [7] on remaining part of the boundary. The problem can be reduced to determination of the minimum value of the functional:

$$I[u] = \frac{1}{2}\int_{\Omega} \gamma \left|\nabla u\right|^2 \mathrm{dxdy}. \tag{2}$$

III. NUMERICAL METHODS

Our optimization algorithm relays on several numerical methods. This section is devoted to them.

ISBN: 978-1-61208-354-4

A. *Boundary Element Method*

Boundary Element Method (BEM) is a well known numerical technique used to solve partial differential equations [3]. In literature, there are a lot of extensions of BEM. For example, a lot of effort has been put into combining BEM and the Finite Element Method (FEM). Another example is coupling BEM with infinite elements [4][5]. It gives us the possibility to solve equations with boundaries described by open curves. In the forward problem, we start our considerations from the following formula (proper for all boundary points) [3]:

$$\frac{1}{2}u(\vec{r}_i)+\sum_{j=1}^{N}\int_{\Gamma_j}u(\vec{r})\,q^*(\vec{r},\vec{r}_i)\,d\gamma_j=\sum_{j=1}^{N}\int_{\Gamma_j}q(\vec{r})\,u^*(\vec{r},\vec{r}_i)\,d\gamma_j. \tag{3}$$

The symbol u represents electrical potential, whereas q defines its normal derivative. The Green's function [4,14] and its normal derivative are denoted by u* and q*, respectively. In (3), we have N finite boundary elements. Next, we have introduced infinite boundary elements and the governing equation (4) has been derived. This integral equation is given by:

$$\begin{aligned}&\frac{1}{2}u(\vec{r}_i)+\sum_{j=2}^{N-1}u_j\int_{\xi=-1}^{\xi=+1}q^*(\vec{r}_j(\xi),\vec{r}_i)\,d\gamma_j+\\&+u_1\int_{\xi\to-\infty}^{\xi=+1}S_\infty(\xi)q^*(\vec{r}_1(\xi),\vec{r}_i)\,d\gamma_1+u_N\int_{\xi=-1}^{\xi\to+\infty}S_\infty(\xi)q^*(\vec{r}_N(\xi),\vec{r}_i)\,d\gamma_N\\&=\sum_{j=2}^{N-1}q_j\int_{\xi=-1}^{\xi=+1}u^*(\vec{r}_j(\xi),\vec{r}_i)\,d\gamma_j+\\&+q_1\int_{\xi\to-\infty}^{\xi=+1}S_\infty(\xi)u^*(\vec{r}_1(\xi),\vec{r}_i)\,d\gamma_1+q_N\int_{\xi=-1}^{\xi\to+\infty}S_\infty(\xi)u^*(\vec{r}_N(\xi),\vec{r}_i)\,d\gamma_N.\end{aligned} \tag{4}$$

Symbol S_∞ denotes the sum of the interpolation functions with exponential decay along infinite boundary elements. One should notice that in our model there is only one open boundary curve. However, generalizations of (4) can be easy done. In mathematical model, we assume that in N – 2 nodes the normal derivatives q equal zero. Only in two nodes we set the electrical potential.

B. *Level Set Method*

The level set function ϕ has the following properties:

$$\begin{aligned}&\phi(\mathrm{r},t)=0\ \text{ for }\ (\mathrm{x,y})\in\partial\Omega(t)\equiv\Gamma(t).\\&\phi(\mathrm{r},t)>0\ \text{ for }\ (\mathrm{x,y})\in\Omega(t),\\&\phi(\mathrm{r},t)<0\ \text{ for }\ (\mathrm{x,y})\notin\Omega(t).\end{aligned} \tag{5}$$

The motion is seen as the convection of values (levels) from the function ϕ with the velocity field $\vec{v}$. Such process is described by the Hamilton-Jacobi equation:

$$\frac{\partial\phi}{\partial t}+\vec{v}\cdot\nabla\phi=0. \tag{6}$$

Here, $\vec{v}$ is the desired velocity on the interface, and is arbitrary elsewhere. Actually, only the normal component of $\vec{v}$ is needed ($v_n\equiv\vec{v}\cdot\vec{n}\equiv\vec{v}\cdot\nabla\phi/|\nabla\phi|$), so (6) becomes:

$$\frac{\partial\phi}{\partial t}+v_n\,|\nabla\phi|=0. \tag{7}$$

We can update the level set function ϕ by solving discretized version of the Hamilton-Jacobi equation:

$$\frac{\phi^{k+1}-\phi^k}{\Delta t}+v_n^k\,\left|\nabla\phi^k\right|=0. \tag{8}$$

Transforming above equation, we get:

$$\phi^{k+1}=\phi^k-v_n^k\,\left|\nabla\phi^k\right|\Delta t. \tag{9}$$

The gradient of the level set function in the k-th time step ($\left|\nabla\phi^k\right|$) has been calculated by the essentially non-oscillatory (ENO) polynomial interpolation scheme. The stability of received solution is achieved by Courant-Friedreichs-Lewy condition (CFL condition):

$$\varDelta t<\frac{\min(\varDelta x,\varDelta y)}{\max(|\,\vec{v}\,|)}. \tag{10}$$

Inequality (10) is satisfied by choosing the CFL number α:

$$\varDelta t\,\frac{\max(|\,\vec{v}\,|)}{\min(\varDelta x,\varDelta y)}=\alpha, \tag{11}$$

where 0 < α < 1. The optimum value equals 0.9.

The calculated velocity must be extended off the interface to the whole domain. This process is called the extension of velocity and is based on the solution of the additional partial differential equation. The reference [1] suggests:

$$\frac{\partial v_n}{\partial t}+S(\phi)\,\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla v_n=0, \tag{12}$$

where S(ϕ) is defined as following [1]:

$$S(\phi)=\frac{\phi}{\sqrt{\phi^2+\varepsilon^2}}. \tag{13}$$

In (13) |ε| << 1. Additionally, we need to extend the velocity to neighborhood of the interface, by defining velocity along normal direction (see Fig. 1).

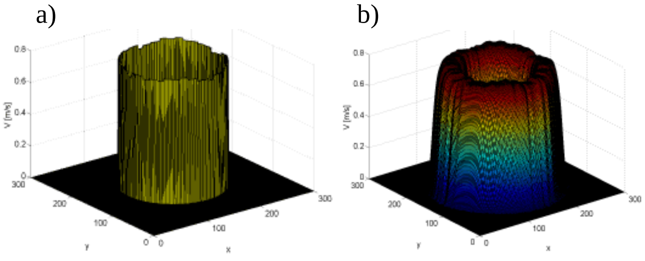


Figure 1. The velocity calculated for the first iteration step: a) - before extension; b) - after extension.

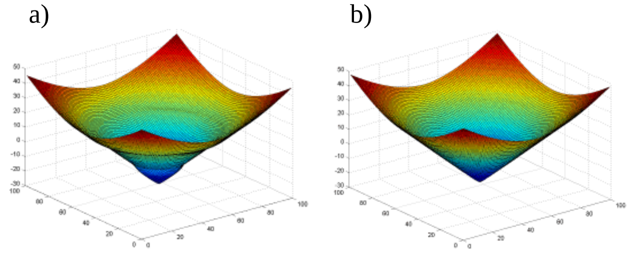


Figure 2. The level set function ϕ: a) - before reinitialization process; b) - after reinitialization process.

ISBN: 978-1-61208-354-4

Reinitialization is necessary when flat or steep regions complicate the determination of the zero contour. The level set function ϕ is signed distance function if at given time for every point (x,y):

$$|\nabla\phi| = 1. \qquad (14)$$

Reinitialization is based on replacing ϕ by another function that has the same zero level set, but satisfies condition (14) (see Fig. 2). This process is described by following partial differential equation [1]:

$$\frac{\partial\phi}{\partial t} + S(\phi)\,(|\nabla\phi| - 1) = 0. \qquad (15)$$

Differential equation (15) is solved until a steady state is achieved. Similar to the velocity extension, a first order upwind scheme for the spatial dimension and forward Euler time discretization is used.

C. Shape derivative

The topological methods are used in order to solve the inverse problem in EIT. Very important concept for our research is so-called shape derivative. The shape derivative is often used in optimization problems [7].

Let λ be the adjoint function satisfying:

$$-\Delta\lambda = u - u_m. \qquad (16)$$

The material derivative (Lagrangian derivative) $\dot{u}(x)$ is given by:

$$\dot{u}(\vec{r}) \equiv \lim_{t\to 0}\frac{u_t(\vec{r} + t\vec{v}(\vec{r})) - u(\vec{r})}{t}, \qquad (17)$$

where $(x,y)\in\Omega_t$. The shape derivative is following:

$$u'(\vec{r}) \equiv \lim_{t\to 0}\frac{u_t(\vec{r}) - u(\vec{r})}{t} = \dot{u}(\vec{r}) - \vec{v}(\vec{r})\cdot\nabla u(\vec{r}). \qquad (18)$$

The steepest descent direction $\vec{v}$ is given by [7]:

$$\vec{v} = -(\nabla u\cdot\nabla\lambda)\,\vec{n}. \qquad (19)$$

We need the shape derivative so that derive formula for velocity (19). The normal velocity is evaluated by using weighted least squares interpolation to get:

$$v_n^k = \nabla u^k\cdot\nabla\lambda^k + \varepsilon\kappa^k. \qquad (20)$$

In next step of our procedure the level set function ϕ is updated:

$$\phi^{k+1} = \phi^k - \left(\nabla u^k\cdot\nabla\lambda^k + \varepsilon\kappa^k\right)\left|\nabla\phi^k\right|\Delta t, \qquad (21)$$

where Δt is obtained from CFL condition (11).

D. Optimization algorithm

For the minimization problem iterative coupling of the level set method and the topological gradient method has been proposed. Both methods are gradient-type algorithms, and the coupled approach can be cast into the framework of alternate directions descent algorithms.

The level set method relies on the shape derivative, while the topological gradient method is based on the topological derivative. The proposed algorithm is iterative method, structured as follows:

- From the level set function at initial time, find necessary interface information.
- Use FEM or BEM to solve the equation (1) and next compute the difference of the obtained solution with the observed data.
- Solve the Poisson’s equation (adjoint equation) – (16).
- Find velocity in the normal direction – (20).
- Update the level set function – (21).
- Reinitialize the level set function – (15).
- Calculate value of the objective function.

IV. Numerical Results

In the examples reported below, several numerical models with different discretization elements are presented. Additionally, we present different geometries of the conductivity distributions. We assume that the electrical conductivity of searched objects is known. The representation of the boundary shape and its evolution during an iterative reconstruction process is achieved by the level set method and the gradient method coupled together. In forward problem which is given by equation (1), we have used FEM or BEM. Additionally, different zero level set functions have been selected. Therefore, quality of the image reconstruction can be evaluated in different cases.

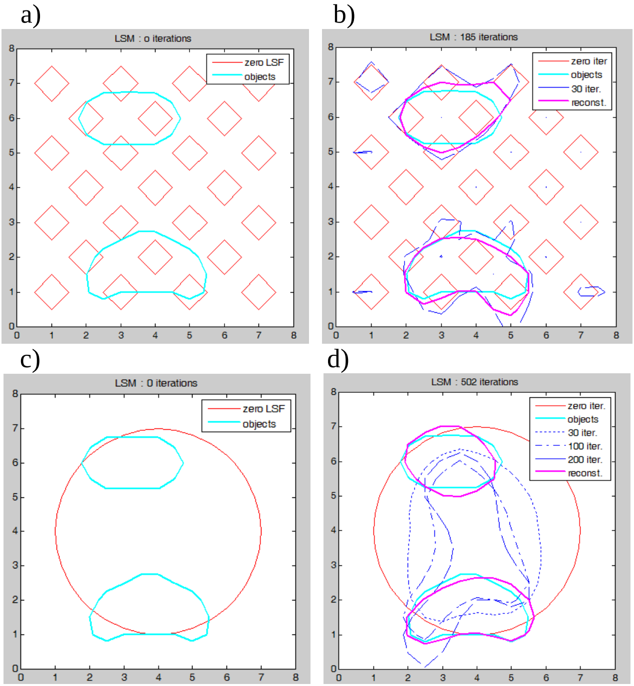


Figure 3. Images reconstruction: a), c) - the original objects and the zero contour from the level set function; b), d) - the process of the image reconstruction.

Fig. 3 shows the image reconstruction with two different groups of objects. Fig. 3a) and Fig 3c) depict two objects. The zero contour curve from the level set function is red, the following iterations are blue. The images from Fig. 3 show the original objects and reconstruction after indicated number of iterations. In the example from Fig. 3c), the zero level at initial step is represented by circle. The process of reconstruction is good, because the region borders are located nearly the object edges. The object function in the Fig. 3b) achieves minimum after 185 iterations, whereas the same object in the Fig. 3d) achieves the minimum after 502 iterations.

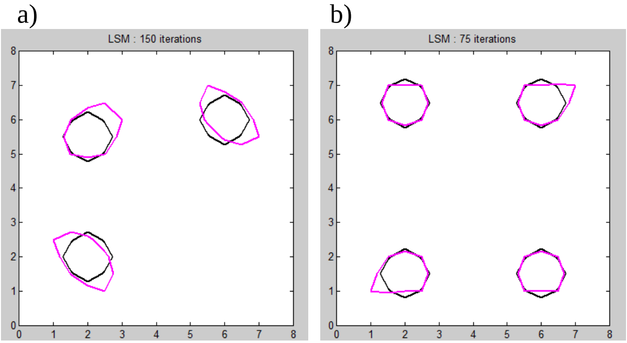


Figure 4. The image reconstruction: a) - 3 objects; b) - 4 objects.

Results of the iteration process as described above are shown in Fig. 4. Unknown structures are marked by the black line; simulated objects are marked by the pink line. For indicated numbers of iterations, the unknown structures have been found.

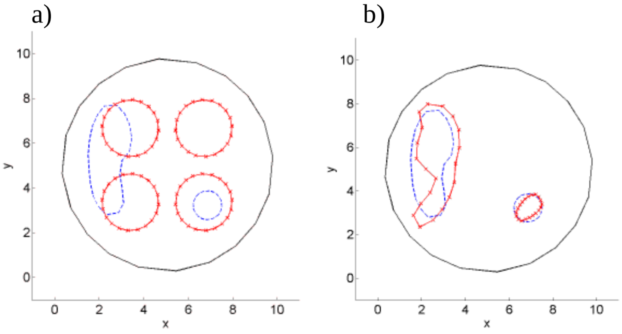


Figure 5. The black line marks the outside border of the examined structure with the internal unknown objects (marked by the blue lines). The red line marks simulated objects of zero level contours: a) - first iteration step; b) - the last iteration step (300th).

The last example of the reconstruction technique is given in Fig. 5. The image reconstructions were achieved by coupling of the level set method, the gradient technique and BEM. Our optimization algorithm works well.

V. Conclusions

An algorithm based on topological and shape derivative and the level set method have been proposed in this work. It is iterative algorithm where repeatedly the shape boundary evolves smoothly and new small objects are detected. An efficient algorithm for solving the forward and inverse problems would also improve a lot of the numerical performances of the proposed methods. In the model problem from EIT, it is required to identify unknown conductivities from near-boundary measurements of the potential. The level set function techniques have been shown to be successful to identify the unknown boundary shapes. The accuracy of the image reconstruction is better than gradient methods. The number of iterations determine the position and shape of zero level set functions. In this algorithm, we can control the process of the image reconstruction. Next advantage of this algorithm is obtaining a good quality results for the poor mesh (16x16 and 32x32 resolution). Other methods have not such properties.

References

[1] S. Osher and R. Fedkiw, “Level Set Methods and Dynamic Implicit Surfaces”, Springer, New York 2003.

[2] J. A. Sethian, “Level Set Methods and Fast Marching Methods”, Cambridge Univeristy Press, 1999.

[3] P. K. Kythe, “An introduction to Boundary Elements Methods”, CRC Press, 1995.

[4] G. Beer and J. O. Watson, “Infinite boundary elements”, International Journal for Numerical Methods in Engineering, vol. 28, 1989, pp. 1233-1247.

[5] G. Beer, J. O. Watson, and G. Swoboda, “Three-dimensional analysis of tunnels using infinite boundary elements”, Computers and Geotechnics, vol. 3, 1987, pp. 37-58.

[6] G. Allaire, F. De Gournay, F. Jouve, and A. M. Toader, “Structural optimization using topological and shape sensitivity via a level set method”, Control and Cybernetics, vol. 34, 2005, pp. 59-80.

[7] K. Ito, K. Kunish, and Z. Li, “The Level-Set Function Approach to an Inverse Interface Problem”, Inverse Problems, vol. 17, no. 5, 2001, pp. 1225-1242.

[8] S. Osher and J. A. Sethian, “Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations”, Journal of Computational Physics, vol. 79, 1988, pp. 12-49.

[9] S. Osher and R. Fedkiw, “Level Set Methods: An Overview and Some Recent Results”, Journal of Computational Physics, vol. 169, 2001, pp. 463-502.

[10] S. Osher and F. Santosa, “Level set methods for optimization problems involving geometry and constraints. Frequencies of a two-density inhomogeneous drum”, Journal of Computational Physics, vol. 171, 2001, pp. 272-288.

[11] T. Rymarczyk, S. F. Filipowicz, J. Sikora, and K. Polakowski, “A piecewise-constant minimal partition problem in the image reconstruction”, Electrical Review, vol. 12, 2009, pp. 141-143.

[12] J. Sokolowski and A. Zochowski, “On the topological derivative in shape optimization”, SIAM Journal on Control and Optimization, vol. 37, 1999, pp. 1251–1272.

[13] C. Tai, E. Chung, and T. Chan, “Electrical impedance tomography using level set representation and total variational regularization”, Journal of Computational Physics, vol. 205, no. 1, 2005, pp. 357–372.

[14] T. Rymarczyk, J. Sikora, and B. Waleska, “Coupled Boundary Element Method and Level Set Function for Solving Inverse Problem in EIT”, Proc. 7th World Congress on Industrial Process Tomography, Sep. 2013, pp. 312-319.

The Development and Analysis of Analytic Method as Alternative for Backpropagation in Large-Scale Multilayer Neural Networks

Mikael Fridenfalk
Department of Game Design
Uppsala University
Visby, Sweden
mikael.fridenfalk@speldesign.uu.se

***Abstract*—This paper presents a least-square based analytic solution of the weights of a multilayer feedforward neural network with a single hidden layer and a sigmoid activation function, which today constitutes the most common type of artificial neural networks. This solution has the potential to be effective for large-scale neural networks with many hidden nodes, where backpropagation is known to be relatively slow. At this stage, more research is required to improve the generalization abilities of the proposed method.**

Keywords-analytic; FNN; large-scale; least square method; neural network; sigmoid

I. Introduction

The artificial neural network constitutes one of the most interesting and popular computational methods in computer science. The most well-known category is the multilayer Feedforward Neural Network, here called FNN, where the weights of the network are estimated by an iterative training method called backpropagation [6]. Although this iterative method is, given the computational power of modern computers, relatively fast for small networks, it is rather slow for large networks [1]. To accelerate the training speed of FNNs, many approaches has been suggested based on the least square method [2]. Although the presentation on the implementation, as well as of the data on the robustness of these methods may be improved, the application of the least square method as such seems to be a promising path to investigate [7].

What we presume to be required for a new method to replace backpropagation in such networks, is not only that it is efficient, but also that it is superior compared to existing methods and is easy to understand and implement. The goal of this paper is therefore to investigate the possibility to find an *analytic* solution for the weights of an FNN, i.e., without any iterations involved, that is easily understood and that may be implemented relatively effortlessly, using a mathematical application such as Matlab [5]. As a brief overview, the analytic solution proposed in this paper is formulated in Section II, followed by a description of the experimental setup in Section III, and by experimental results in Section IV, presented in Tables I-V.

II. Analytic Solution

To start with, a textbook FNN is vectorized, based on a sigmoid activation function $S(t) = 1/(1+e^{-t})$. The weights $\mathbf{V}$ and $\mathbf{W}$ of such a system (often denoted as W_{IH} versus W_{HO}), may be expressed according to Figs. 1-2. In this representation, defined here as the normal form, the output of the network may be expressed as:

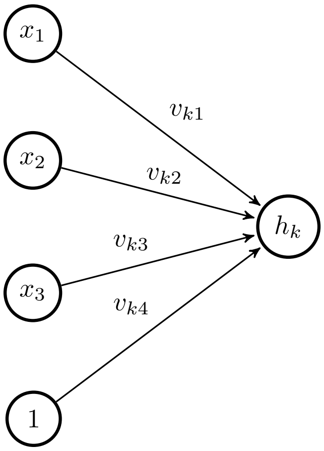


Figure 1. An example with three input nodes ($M = 3$), $h_k = S(\mathbf{v}_k\mathbf{u}) = S(v_{k1}x_1 + v_{k2}x_2 + v_{k3}x_3 + v_{k4})$, using a sigmoid activation function S.

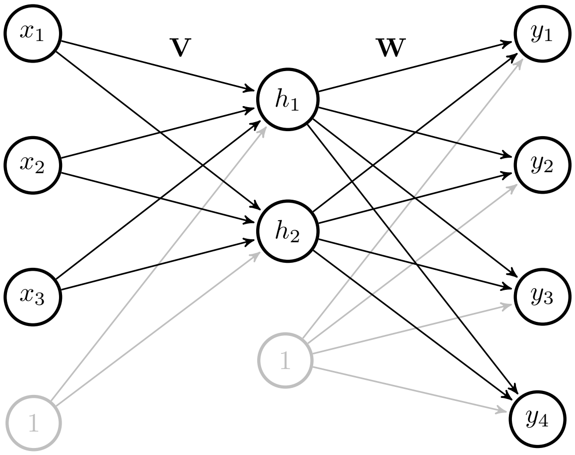


Figure 2. A vectorized model of a standard FNN with a single hidden layer, in this example with $M = 3$ input nodes, $H = 2$ hidden nodes, $K = 4$ output nodes and the weight matrices $\mathbf{V}$ and $\mathbf{W}$, using a sigmoid activation function for the output of each hidden node. In this model, the biases for the hidden layer and the output layer correspond to column $M + 1$ in $\mathbf{V}$ versus column $H + 1$ in $\mathbf{W}$.

$$\mathbf{y} = \mathbf{W}\mathbf{h} = \mathbf{W}\left[\frac{S(\mathbf{V}\mathbf{u})}{1}\right], \ \mathbf{u} = \begin{bmatrix}\mathbf{x}\\1\end{bmatrix} \tag{1}$$

where $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_M]^T$ denotes the input signals, $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_K]^T$ the output signals, and S, an element-wise

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{\mathbf{V}^T} = \underbrace{\left(\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} * & * & 1 \\ * & * & 1 \\ * & * & 1 \\ * & * & 1 \\ * & * & 1 \\ * & * & 1 \end{bmatrix} \right)^{-1}}_{\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{(\mathbf{UU}^T)^{-1}}} \underbrace{\left[\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \right]}_{\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{\mathbf{UH}_0^T}}$$

$$\underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{H}} = \left[\frac{S\left(\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{U}} \right)}{\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{1}^T}} \right]$$

$$\underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\mathbf{W}^T} = \underbrace{\left(\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * & * & 1 \\ * & * & * & * & * & 1 \\ * & * & * & * & * & 1 \\ * & * & * & * & * & 1 \\ * & * & * & * & * & 1 \\ * & * & * & * & * & 1 \end{bmatrix} \right)^{-1}}_{(\mathbf{HH}^T)^{-1}} \underbrace{\left[\underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\mathbf{Y}_0^T} \right]}_{\underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\mathbf{HY}_0^T}}$$

$$\underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{\mathbf{H}}$$

Figure 3. A visual representation of the evaluation of weights $\mathbf{V}$ and $\mathbf{W}$ by the analytic method presented in this paper, and the actual output $\mathbf{Y}$, in this example as a function of six training points, $N = 6$, the training input and output sets $\mathbf{U}$ and $\mathbf{Y}_0$, with two inputs, $M = 2$, four outputs, $K = 4$, and five hidden nodes, $H = N - 1 = 5$. In this figure, an asterisk denotes a floating-point number. To facilitate bias values, certain matrix elements are set to one.

sigmoid function. In this paper, a winner-take-all classification model is used, where the final output of the network is the selection of the output node that has the highest value. Since the sigmoid function is a constantly increasing function and identical for each output node, it can be omitted from the output layer, as $\max(\mathbf{y})$ results in the same node selection as $\max(S(\mathbf{y}))$. Further on, presuming that the training set is highly fragmented (the input-output relations in the training sets were in our experiments established by a random number generator), denoting N as the number of training points, the number of hidden nodes is preferred to be set to $H = N - 1$. Defining a batch (training set), the input matrix $\mathbf{U}$, may be expressed as:

$$\mathbf{U} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2}$$

where column vector i in $\mathbf{U}$, corresponds to training point i, column vector i in $\mathbf{Y}_0$ (target output value) and in $\mathbf{Y}$ (actual output value). Further, defining $\mathbf{H}$ of size $N \times N$, as the batch values for the hidden layer, given a training set of input and output values and $M^+ = M + 1$, the following relations hold:

$$\mathbf{U} = \left[\frac{\mathbf{X}}{\mathbf{1}^T} \right] : [M^+ \times N] \tag{3}$$

$$\mathbf{H} = \left[\frac{S(\mathbf{VU})}{\mathbf{1}^T} \right] : [N \times N] \tag{4}$$

$$\mathbf{Y} = \mathbf{WH} : [K \times N] \tag{5}$$

To evaluate the weights of this network analytically, we need to evaluate the target values (points) of $\mathbf{H}_0$ for the hidden layer. In this paper, the initial assumption is that any point is feasible, as long as it is unique for each training set. Therefore, in this model, $\mathbf{H}_0$ is merely composed of random numbers. Thus, the following evaluation scheme is preliminary suggested for the analytic solution of the weights of such network:

$$\mathbf{V}^T = (\mathbf{UU}^T)^{-1}\mathbf{UH}_0^T : [M^+ \times H] \tag{6}$$

$$\mathbf{W}^T = (\mathbf{HH}^T)^{-1}\mathbf{HY}_0^T : [N \times K] \tag{7}$$

where a least square solution is used for the evaluation of each network weight matrix. Such an equation is nominally expressed as $\mathbf{Ax} = \mathbf{b}$, with the least square solution [2]:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{8}$$

where (8) corresponds to input row vectors $\mathbf{u}_i^T$. It is the use of *column* input vectors, $\mathbf{u}_i$, that yield the expressions found in (6)-(7). Since the mathematical expressions for the analytic solution of the weights of a neural network may still be difficult to follow, an attempt has been made in Fig. 3 to visualize the matrix operations involved. Note that while a nonlinear activation function (such as the sigmoid function) is vital for the success of such network, the inclusion of a bias is not essential. It is for instance possible to omit the biases and to replace $\mathbf{H}_0$ with an identity matrix $\mathbf{I}$. Such configuration would instead yield the following formula for the evaluation of $\mathbf{V}$ and $\mathbf{H}$ (where $\mathbf{UI}$ can further be simplified as $\mathbf{U}$):

$$\mathbf{V}^T = (\mathbf{UU}^T)^{-1}\mathbf{UI} : [M^+ \times N] \tag{9}$$

$$\mathbf{H} = S(\mathbf{VU}) : [H \times N] \tag{10}$$

III. Experimental Setup

To test the solution presented in this paper, a minimal mathematical engine was developed in C++, with the capability to solve $\mathbf{X}$ in a linear matrix equation system of the form:

$$\mathbf{AX} = \mathbf{B} \tag{11}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{X}$ denote matrices of appropriate sizes, since it is numerically more efficient to solve a linear equation system directly, than by matrix inversion. In this system, the column vectors of $\mathbf{X}$ are evaluated using a single Gauss-Jordan elimination cycle, where each column vector $\mathbf{x}_i$ in $\mathbf{X}$ corresponds to the column vector $\mathbf{b}_i$ in $\mathbf{B}$, thereby increasing the evaluation speed compared with standard equation solvers.

Backpropagation was in our experiments implemented in C++, using the code presented by Jones [4], as a reference. To measure the efficiency of the new method compared with the standard method (backpropagation), we used a typical definition for the mean-squared error:

$$\epsilon = \frac{1}{2N}\sum_i^K\sum_j^N(a_{ij} - y_{ij})^2 \tag{12}$$

where a_{ij} denotes an element in $\mathbf{Y}_0$ (target value), and y_{ij} the corresponding element in $\mathbf{Y}$ (actual value). Although the new method does not intrinsically benefit from such definition (since there is no need here for differentiation of the mean-squared error, which is however useful for backpropagation), to simplify comparison in Tables I-V, the same definition of the mean-squared error was also used for the new method.

IV. Experimental Results

The experimental results presented in this paper, as shown in Tables I-V, are based on ten individual experiments for each parameter setting using different random seeds, where $\bar{t}$ denotes average execution time, using a single CPU-core on a modern laptop computer, $\bar{\epsilon}$, the average value of the mean-squared errors, and $\tilde{\epsilon}$, the median value. The success rate, $\bar{s}$, is similarly based on an average value. Since the variation of the results is large between the experiments, the average values are in general larger than the median values. If the number of experiments per parameter setting is increased, according to our experiments, the average value tends to increase as well.

On a note of preliminary experiments with respect to robustness, regarding the generalization abilities of the network, the new method showed to steeply lose accuracy with the addition of noise to the input values, compared with a network trained by backpropagation. This shows that although the results seem to be in order according to the tables presented in this paper, the new method lacks robustness for direct use. Further experiments showed however that even small measures, such as an increase in the input range of the network by doubling the size of the training set with the addition of perturbation and a more conscious design of $\mathbf{H}_0$, by for instance the clustering of the random values as a function of the output values, or for layers with few hidden nodes, the binary encoding [3] of $\mathbf{H}_0$, led to significant improvements of the robustness of the new method. This is an encouraging sign, since the sizes of $\mathbf{UU}^T$ and $\mathbf{HH}^T$ are as shown in Fig. 3, independent of N (assuming H is kept intact as N is increased). There is thus a chance that these robustness issues can be solved, in this context, without any significant impact to the computational speed of the new method.

V. Conclusion

The method proposed in this paper is fast, accurate for networks with a sufficient number of hidden nodes, and straightforward to implement; but is, at this stage, based on preliminary robustness tests, significantly much less robust (thus, resembling an overtrained network), compared with a well-trained FNN through backpropagation. Even small modifications of the new method showed however to increase robustness significantly, which is promising for further research.

References

[1] R. P. W. Duin, "Learned from Neural Networks", ASCI2000, Lommel, Belgium, 2000, pp. 9-13.

[2] C. H. Edwards and D. E. Penney, Elementary Linear Algebra, Prentice Hall, 1988.

[3] F. Gray, Pulse Code Communication, Patent, U.S., no. 2632058, 1947.

[4] M. T. Jones, AI Application Programming, 2nd ed., Charles River, 2005.

[5] Matlab, The MathWorks, Inc. <http://www.mathworks. com/> [retrieved: April 11, 2014].

[6] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3nd ed., Prentice Hall, 2009.

[7] Y. Yam, "Accelerated Training Algorithm for Feedforward Neural Networks Based on Least Squares Method", Neural Processing Letters, vol. 2, no. 4, 1995, pp. 20-25.

Table I. BACKPROPAGATION WITH $H = N - 1$ AND AVERAGE SUCCESS RATE $\bar{s}$

M	N	H	K	Iterations	$\bar{t}$	$\bar{\epsilon}$	$\tilde{\epsilon}$	$\bar{s}$ (%)
5	20	19	5	10^4	46.6 ms	0.0671	0.0620	93.0
5	20	19	5	10^6	4.39 s	0.0175	$4.64 \cdot 10^{-5}$	96.5
10	50	49	10	10^4	182 ms	0.116	0.114	83.2
10	50	49	10	10^6	18.1 s	0.0680	0.0650	86.4
20	50	49	20	10^4	333 ms	0.0394	0.0392	94.6
20	50	49	20	10^6	33.3 s	0.0170	0.0200	96.6
40	100	99	40	10^4	1.27 s	0.0671	0.0670	88.9
40	100	99	40	10^6	127 s	0.0180	0.0200	96.4

Table II. NEW METHOD WITH $H = N - 1$

M	N	H	K	$\bar{t}$	$\bar{\epsilon}$	$\tilde{\epsilon}$	$\bar{s}$ (%)
5	20	19	5	332 μs	$3.34 \cdot 10^{-8}$	$2.00 \cdot 10^{-13}$	100.0
10	50	49	10	3.74 ms	$6.99 \cdot 10^{-9}$	$2.26 \cdot 10^{-12}$	100.0
20	50	49	20	4.79 ms	$2.68 \cdot 10^{-13}$	$5.93 \cdot 10^{-16}$	100.0
40	100	99	40	36.7 ms	$3.93 \cdot 10^{-12}$	$2.33 \cdot 10^{-13}$	100.0

Table III. BACKPROPAGATION WITH $H < N - 1$ AND 10^4 ITERATIONS

M	N	H	K	$\bar{t}$	$\bar{\epsilon}$	$\tilde{\epsilon}$	$\bar{s}$ (%)
5	20	5	5	14.7 ms	0.130	0.125	86.5
10	50	10	10	43.1 ms	0.227	0.237	71.6
20	50	20	20	144 ms	0.0624	0.0599	94.2
40	100	40	40	531 ms	0.115	0.115	84.8

Table IV. NEW METHOD WITH $H < N - 1$

M	N	H	K	$\bar{t}$	$\bar{\epsilon}$	$\tilde{\epsilon}$	$\bar{s}$ (%)
5	20	5	5	59 μs	0.281	0.289	58.5
10	50	10	10	370 μs	0.350	0.350	50.0
20	50	20	20	1.30 ms	0.279	0.277	91.8
40	100	40	40	9.24 ms	0.290	0.290	98.5

Table V. SELECTIVE USE OF BIAS FOR THE NEW METHOD, WITH $M = 40$, $N = 100$, $H = 99$, AND $K = 40$

H_b	Y_b	$\bar{t}$	$\bar{\epsilon}$	$\tilde{\epsilon}$	$\bar{s}$ (%)
No	No	35.7 ms	0.00514	0.00513	100.0
No	Yes	36.4 ms	$1.35 \cdot 10^{-12}$	$1.91 \cdot 10^{-14}$	100.0
Yes	No	36.2 ms	0.00544	0.00534	100.0

A Formal Online Monitoring approach to Test Network Protocols

Xiaoping Che, Stephane Maag and Jorge Lopez

Institut Mines-Telecom/Telecom SudParis, CNRS UMR 5157 SAMOVAR.
Evry, France
Email: {xiaoping.che,stephane.maag,jorge.eleazar.lopez_coronado}@telecom-sudparis.eu

***Abstract*—While the current network protocols and systems become more and more complex, the testing process of their functional and non-functional behaviors comes to be crucial. Among the testing techniques, we herein focus on their test at runtime in an online way which requires the ability to handle numerous messages in a short time with the same offline testing preciseness. Meanwhile, since online testing is a long term continuously process, the tester has to undergo severe conditions when dealing with large amount of nonstop traces. In this paper, we present a novel logic-based online passive testing approach to test, at runtime, the protocol conformance and performance through formally specified properties with new definitions of verdicts. Furthermore, we experimented our approach with several Session Initiation Protocol (SIP) properties in a real IP Multimedia Subsystem environment and obtained relevant verdicts.**

Keywords–Online Testing; Network Protocols; Monitoring

I. Introduction

In order to evaluate the quality and conformance of a system or an implementation under test (IUT) in relation with a standard, the testing process is a crucial activity. Among the well known and commonly applied approaches, the *passive* testing techniques (also called *monitoring*) are today gaining efficiency and reliability [1]. These techniques are divided in two main groups: *online* and *offline* testing approaches. Offline testing computes test scenarios before their execution on the IUT and gives verdicts afterwards, while online testing continuously tests during the operation phase of the IUT.

When we apply online testing approaches, the collection of traces is avoided and the traces are eventually not finite. Indeed, testing a protocol at runtime may be performed during a normal use of the system without disturbing the process. Several online testing techniques have been studied by the community in order to test systems or protocol implementations [2] [3] [4]. These methods provide interesting studies and have their own advantages, but they also have several drawbacks such as the presence of false negatives, space and time consumption, often related to a needed complete formal model [5], etc. Although they bring solutions, new results and perspectives to the protocol and system testers, they also raise new challenges and issues. The main ones are the non-collection of traces and their on-the-fly analysis. The traces are observed (through an interface and an eventual sniffer) and analyzed on-the-fly to provide test verdicts and no trace sets should be studied a posteriori to the testing process. In this work, we propose a novel formal online passive testing approach that is applied at runtime to test the functional and non-functional requirements of an implementation under test.

Based on a previous proposed methodology [6] [7], we propose its extension to present a logic-based passive testing approach for checking the requirements of communicating protocols. In [6] and [7], we presented our formalism that was applied to test in an offline way the conformance and performance of an IUT. In this new paper, we develop our approach to test these two aspects in an online way in considering the above mentioned inherent constraints and challenges. Furthermore, our framework is designed to test them at runtime, with new required verdicts definitions of '*Pass*', '*Fail*', '*Time-Fail*', '*Data-Inc*' and '*Inconclusive*'. Finally, in order to demonstrate the efficiency of our online approach, we apply it on a real IP Multimedia Subsystem (IMS) communicating environment.

Our paper's primary contributions are:

- A formal passive online testing approach to avoid stopping the execution of the testing process when monitoring a tested protocol. New verdicts are provided in order to consider that the monitored traces are not cut.
- A testing process that is executed in a transparent way without overloading, overcharging the CPU and memory of the used equipment on which the tester will be run. Mechanisms of notifications is defined.

The reminder of the paper is organized as follows. In Section II, a short review of the related works and problems are provided. In Section III, we describe the architecture and testing process in detail. Our approach has been implemented and relevant experiments are depicted in Section IV. Finally, we conclude and provide perspectives in Section V.

II. Related Works

When studying the literature, we note that there are very few papers tackling online passive testing. We can however cite the following ones.

In [8], the authors proposed two online algorithms to detect 802.11 traffic from packet-header data collected passively at a monitoring point. They built a system for online wireless traffic detection using these algorithms. Besides, some researchers presented a tool for exploring online communication and analyzing clarification of requirements over the time in [9]. It supports managers and developers to identify risky requirements. In [10], the authors defined a formal model based on Symbolic Transition Graph with Assignment (STGA) for both peers and choreography with supporting complex data types. The local and global conformance properties are formalized by the Chor language in their works. We should also cite the work [11] from which an industrial testing tool has been developed. This work is based on formal timed extended invariant to analyze runtime traces with deep packet inspection techniques. However, while most of the functional properties can be easily designed, complex ones with data causality can not. Moreover, although their approach is efficient with an important data flow, the process is still offline with finite traces that are considered as very long. To be complete, we have to mention that studies have also been performed to generate invariant from model-checkers. However, it requires a formal model and it still raises unresolved issues [12].

We may also cite some online active testing approaches from which we got inspired. In [4], the authors presented a framework that automatically generates and executes tests for conformance testing of a composite of Web services described in BPEL. The proposed framework considers unit testing and it is based on a timed modeling of BPEL specification, and an online testing algorithm that assigns verdicts to every generated state. In [13], they presented an event-based approach for modeling and testing the functional behavior of Web Services (WS). Functions of WS are modeled by event sequence graphs (ESG) and they raised the holistic testing concept that integrates positive and negative testing. [14] proposed a data-centric approach to test a protocol by taking account the control parts of the messages as well as the data values carried by the message parameters contained in an extracted execution trace. Interesting and promising results were obtained while testing the SIP protocol.

Inspired from all these above cited works, we propose an online formal passive testing approach by defining functional properties of IUT, without modeling the complete system (contratry to Model Based Testing - MBT) and by considering eventual false negatives. For this latter, we introduce a new verdict '*Time-Fail*' for distinguishing the real functional faults and the faults caused by timeouts. In addition, since online protocol testing is a long-term continuously testing process, we provide a temporary storage to keep the integrity of incoming traces. Furthermore, for the lacking attention to test data portions of messages in current researches, our approach provides the ability to test both the data portion and control portion, accompanying with another new verdict '*Data-Inc*' triggered when no data portions are available during the testing process.

III. Online Testing Approach

In this section, we describe the architecture and testing process of our online testing approach. We also provide the new definitions of online testing verdicts.

A. Architecture of the approach

In our approach, the Horn logic [15] is used for formally expressing properties as formulas. This logic has the benefit of allowing the re-usability of clauses. And it provides better expressibility and flexibility when analyzing protocols. A syntax tree generated from the formulas will be used for filtering incoming traces and optimizing evaluation processes. For the evaluation part, we use the SLD-resolution algorithm for evaluating formulas. The architecture of our online testing approach is illustrated in Figure 1.

B. Testing Process

As shown in Figure 1, the testing process consists of eight parts: Formalization, Construction, Capturing, Generating Filters/Setup, Filtering, Transfer/Buffering, Load Notification and Evaluation.

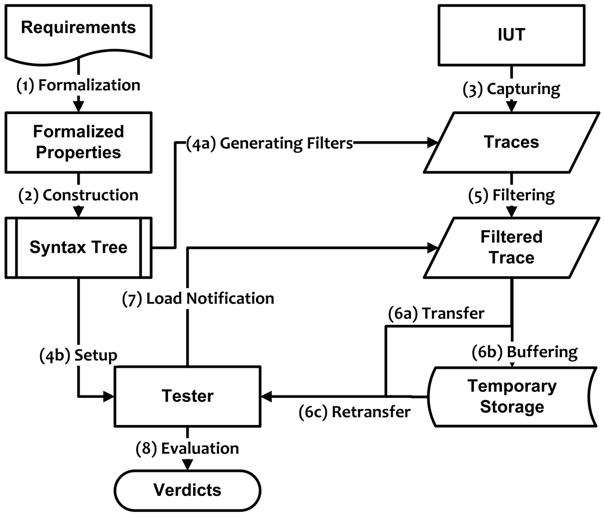


Figure 1. Architecture of our online testing approach

a) Formalization: Initially, informal protocol requirements are formalized using Horn-logic based syntax. A *message* of a protocol P is any element $m \in M_p$. For each $m \in M_p$, we add a real number $t_m \in \mathbb{R}^+$ which represents the time when the message m is received or sent by the monitored entity. Data domains are defined as *atomic* or *compound*. Once given a network protocol P, a *compound* domain M_p can be defined by the set of labels and data domains derived from the message format defined in the protocol specification/requirements.

A *term* is defined in Backus-Naur Form (BNF) as $term ::= c \mid x \mid x.l.l...l$ where c is a constant in some domain, x is a variable, l represents a label, and $x.l.l...l$ is called a *selector variable*. An *atom* is defined as the relations between *terms*, $A ::= p(term, ..., term) \mid term = term \mid term \neq term \mid term < term$. The relations between *atoms* are stated by the definition of *clauses*. A *clause* is an expression of the form $A_0 \leftarrow A_1 \wedge ... \wedge A_n$, where $A_1, ..., A_n$ are *atoms*. Finally, a *formula* is defined by the BNF: $\phi ::= A_1 \wedge ... \wedge A_n \mid \phi \rightarrow \phi \mid \forall_x \phi \mid \forall_{y>x} \phi \mid \forall_{y<x} \phi \mid \exists_x \phi \mid \exists_{y>x} \phi \mid \exists_{y<x} \phi$, where $\exists$ and $\forall$ represent for "it exists" and "for all" respectively. The semantics used in our work is related to the traditional Apt-Van Emdem-Kowalsky semantics for logic programs [16], from which an extended version has been provided in order to deal with messages and trace temporal quantifiers. Due to the space limitation, we will not go into details of the semantics. The interested readers may have a look at the works [6] and [1].

Then the verdicts $\{$'*Pass*', '*Fail*', '*Time-Fail*', '*Inconclusive*', '*Data-Inc*'$\}$ are provided to the interpretation of obtained formulas on real protocol execution traces. However, different from offline testing, definite verdicts should be immediately returned in online testing process. This indicates that only '*Pass*', '*Fail*' and '*Time-Fail*' should be emitted in the final report, and indefinite verdicts '*Data-Inc*' and '*Inconclusive*' will be used as temporary unknown status, but finally must be transformed to one of the definite verdicts at the end of the testing process.

b) Construction: From formalized formulas, a syntax tree is constructed for further testing processes. In this process, each formula representing a requirement will be transformed to an Abstract Syntax Tree (AST) using the TREEGEN algorithm [17]. The standard BNF representation of each formula is the input to construct an AST. All the generated ASTs are finally combined to a syntax tree using a fast merging algorithm [18]. The syntax tree will be transferred to the tester as requirements and will be used to filter the captured traces.

c) Capturing: The monitor consecutively captures traces of the protocol to be tested from points of observations (P.Os) of the IUT, until the testing process finishes. When messages are captured, they are tagged with a time-stamp t_m in order to test the properties with time constraints and to provide verdicts on the performance requirements of the IUT.

d) Generating Filters and Setup: Once the syntax tree is constructed, it will be applied to captured traces for playing the role of a filter. Meanwhile, the tree will also be sent to the tester with the definition of verdicts. According to different conditions, verdicts are defined as below:
- PASS: The message or trace satisfies the requirements.
- FAIL: The message or trace does not satisfy the requirements.
- TIME-FAIL: The target message or trace cannot be observed within the maximum time limitation. Since we are working on online testing, a timeout is used to stop searching target message in order to provide the real-time status. The timeout value should be the maximum response time written in the protocol standard. If we cannot observe the target message within the timeout time, then a *Time-Fail* verdict will be assigned to this property. It has to be noticed that this verdict is only provided when no time constraint is required in the requirement. If any time constraint is required, the violation of this requirement will be concluded as *Fail*, not as a *Time-Fail* verdict.
- INCONCLUSIVE: Uncertain status of the properties. Different from offline testing, this verdict will not appear in the final results. It only exists at the beginning of the test or when the test is paused, in order to describe the indeterminate state of the properties (e.g., a property that requires a special occurrence on the protocol that did not occur yet).
- DATA-INC (Data Inconclusive): In the testing process, some properties may be evaluated through traces containing only control portion (there is no data portion or the latter case mentioned in Step ‘Transferring’). If any property requires for testing the data portion, *Data-Inc* verdicts will be assigned to the property, due to the fact that no data portion can be tested. However, these *Data-Inc* verdicts will be eventually updated to *Pass* or *Fail* based on the data (coming from complete traces) analyzed on the tested properties. Currently we are using worst-case solution (all concluded as *Fail* verdicts). It won’t affect the overall results, since *Data-Inc* verdicts only represent a tiny proportion (less than 0.1%) of the whole traces in our experiments. However, expecting eventual contingencies, we plan, in the future, to apply a support vector machine (SVM) approach [19] in order to train our testing processes and predicate the *Data-Inc* verdicts.

e) Filtering: The incoming captured traces will go through the filtering module, and messages in the traces are filtered into different sets. The unnecessary messages irrelevant to any of the requirements are filtered into the “Unknown” set, and they will not go through the testing process. Finally, traces will be filtered to multiple optimized streams. This step will obviously reduce the processing time, since futile comparisons with irrelevant messages are omitted.

f) Transferring: The filtered traces are transferred (6a) to the tester when the tester is capable for testing. If the tester priority has to be decreased (e.g., the CPU and RAM must be used for another task on this computer of the end-user), a ”load notification” (7) is provided to the monitor in order to transfer/store incoming traces. Based on the message format of the protocols to be tested, different buffering methods will be applied. itemsep=2.5pt, topsep=2pt, partopsep=0pt

- If in the message format, the size of its header is larger than its body. Then the whole message will be buffered in the temporary storage.
- On the contrary, if the size of its header is equal or less than its body, then only the control portion of the packets are buffered (6b) in the temporary storage. Since not all the protocol requirements have specific needs on the data portion, only buffering the control portion will save a lot of memory space when buffering millions of messages.

When the tester is available (notification obtained), the stored traces are retransferred (6c) to the tester. In the latter case mentioned above, only the control portion of packets are provided. In both cases, the continuity of traces is ensured, since no packet will be dropped in any condition. If the protocol requirement has specific needs on the data portion, then the new verdict *Data-Inc* can be given and will be eventually updated to final verdicts by future analysis with the entire traces (the tester is indeed available again).

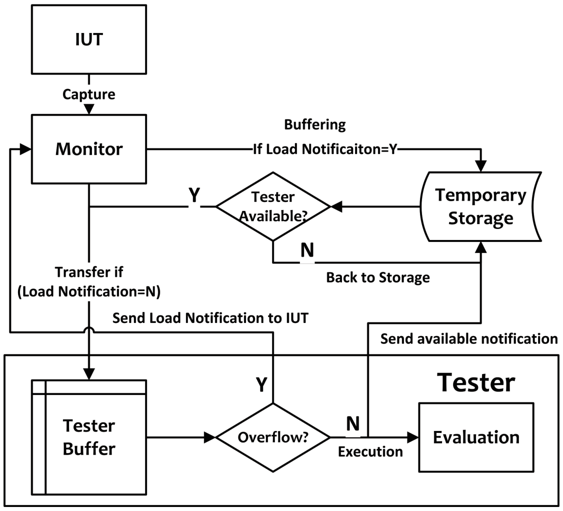


Figure 2. Process of buffering and notification

g) Load Notification: When the tester reaches its limit regarding the amount of data processable or is given a lower priority (e.g., to discharge the CPU / RAM), it sends a ”Load Notification Y” to pause incoming filtered traces and store them in the temporary storage. When the tester is available back, a ”Load Notification N” to release stored traces and to pursue incoming packets is sent. A brief description of processes 6 and 7 is shown in Figure 2.

As the figure illustrates, when captured traces from the IUT are transferred to the tester buffer, a checking overflow function will be called. If the buffer already reached to its maximum capacity, it will notify the IUT to redirect incoming traces to temporary storage in order to avoid the overflow. On the contrary, if the buffer is in a stable condition, it will send the available notification N to the temporary storage for releasing stored messages and to the IUT for returning back to normal transport process.

h) Evaluation: The tester checks whether the incoming traces satisfy the formalized requirements, and provides the final verdicts *Pass, Fail* or *Time-Fail* and temporary verdicts *Inconclusive* or *Data-Inc*.

IV. Experiments

A. Environment

The IMS is a standardized framework for delivering IP multimedia services to users in mobility. It aims at facilitating the access to voice or multimedia services in an access independent way, in order to develop the fixed-mobile convergence. Most communication with its core network and between the services is done using the Session Initiation Protocol (SIP) [20].

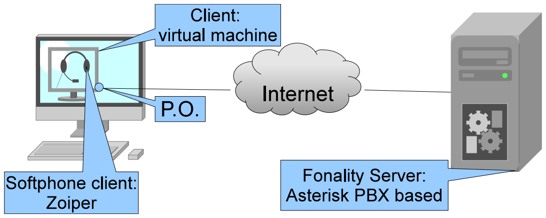


Figure 3. Experiments environment

For our experiments, communication traces were obtained through ZOIPER [21] which is a VoIP soft client, meant to work with any IP-based communication systems and infrastructure. We run four ZOIPER VoIP clients on the virtual machines using VirtualBox for Mac version 4.2.16. On the other side, the server is provided by Fonality [22], which is running Asterisk PBX 1.6.0.28-samy-r115. As Figure 3 shows, the tests are performed in the virtual machines by opening a live capture on the client local interface. This live capture is processed by the clients using an implementation of the formal approach above mentioned and was developed in C code.

B. Test Results

For better understanding how our approach works, we illustrate a simple use case tested on one of the clients. As shown in Figure 4, we have a SIP requirement to be tested: "Every 2xx response for **INVITE** request must be responded with an ACK within 2s", which can be formalized to a formula: $\forall_x(request(x) \wedge x.method = \textbf{INVITE} \rightarrow \exists_{y>x}(responds(y,x) \wedge success(y)) \rightarrow \exists_{z>y}(ackResponse(z,x,y) \wedge withintime(z,y,2s)))$.

This formula will be transformed to a syntax tree. When the syntax tree is generated and transferred to the IUT monitor, it will start to capture the trace and apply the syntax tree as a filter (step 3 and 4) for captured messages. Meanwhile, the syntax tree will be applied in the tester as requirement. Once the captured trace is filtered into different sets (step 5), it will check the Load Notification value first. Currently, the Load Notification value equals to *N*, which makes the tester available to test incoming traces. Then all incoming traces will be sent to the tester directly (step 6a). As soon as the tester receives the trace, it tests the trace through the formalized property. When the tester is almost reaching to its maximum capacity, it will send a load notification value Y back to the monitor (step 7 and 8). In this case, all incoming traces will be stored in the temporary storage (step 6b) until the tester recovers to an available state (step 6c). Finally, after our 2 hours testing process, we got 18864 '*Pass*' verdicts, 5 '*Fail*' verdicts caused by violation of the time constraint and no *Time-Fail* verdicts.

Secondly, we test our approach in a more complex environment. It has been performed to concurrently test five properties on a huge set of messages: "Prop.1: Every request must be responded", "Prop.2: Every request must be responded within 8s", "Prop.3: Every **INVITE** request must be responded", "Prop.4: Every **INVITE** request must be responded within 4s" and "Prop.5: Every **REGISTER** request must be responded".

The table I shows a snapshot of temporary testing verdicts after 3 hours online continuously testing. Benefited from the filtering function, more than 70% irrelevant messages are filtered out before testing process, which apparently reduce the cost of computing resources. Besides, numbers of *Fail* and *Time-Fail* verdicts can be observed. *Time-Fail* verdicts in Prop.1, Prop.3 and Prop.5 indicate that there are 61432, 29673 and 3924 messages respectively that cannot be observed within the timeout, in other words, they are lost during the communication between the client and the server. Besides, the '0' *Fail* verdict indicates there is no error observed in the data portion for these three properties currently. On the other side, *Fail* verdicts reported in Prop.2 and Prop.4 indicate that there are 194579 and 97339 messages that cannot satisfy the time requirement. These *Fail* verdicts include the *Time-Fail* verdicts reported in Prop.1 and Prop.3, since lost messages also violate the time requirement.

Moreover, several '*Inconclusive*' verdicts indicating the numbers of pending procedures for each property can be observed. We also used the control-portion-only buffering mechanism to test the usage of '*Data-Inc*'. All the buffered messages without data portion are successfully reported as '*Data-Inc*' shown in Table I. Since they take a tiny proportion of whole traces (between 0.015% and 0.09%), we conclude them as *Fail* in the worst-case. During the whole testing process, our approach successfully handled this huge set of messages and did not suspend.

V. Conclusion

This paper presents a new logic-based online passive testing approach to test conformance and performance of network protocol implementation. Our approach allows to formally define relations between messages and message data, and then to use such relations in order to define the conformance and performance properties that are evaluated on real-time protocol traces. The evaluation of the property returns a *Pass*, *Fail*, *Time-Fail*, *Inconclusive* or *Data-Inc* result, derived from the given trace. The approach also includes an online testing framework. To verify and test the approach, we designed several SIP properties to be evaluated by our approach. Our methodology has been implemented into an environment which provides the real-time IMS communications, and we successfully obtained relevant results from testing several properties online.

Furthermore, as future works, we aim at applying our approach under billions of messages and extending more

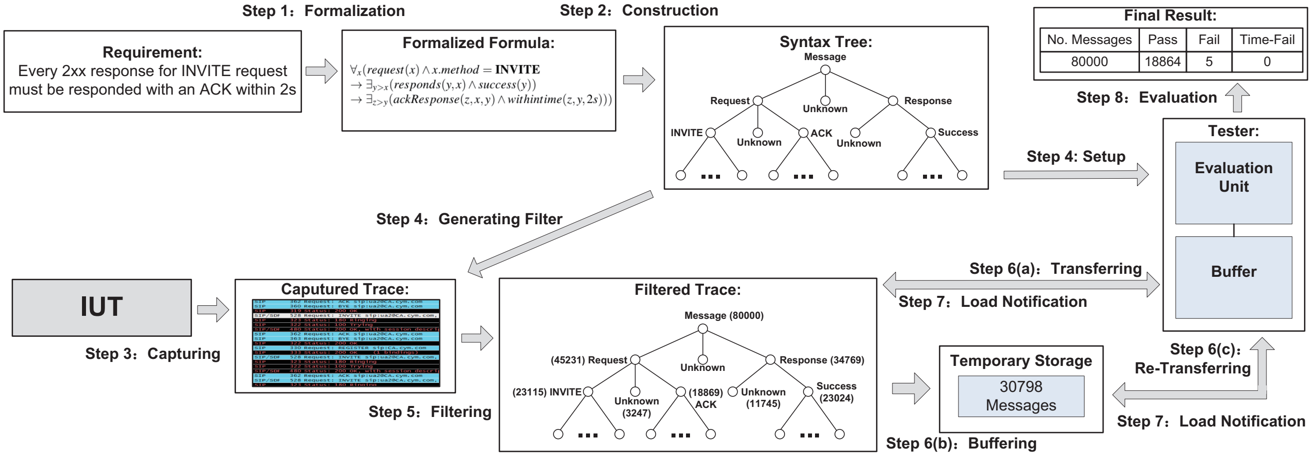

Step 1：Formalization
Step 2：Construction
Requirement:
Every 2xx response for INVITE request must be responded with an ACK within 2s
Formalized Formula:
$\forall_x(request(x) \wedge x.method = \textbf{INVITE}$
$\rightarrow \exists_{y>x}(responds(y,x) \wedge success(y))$
$\rightarrow \exists_{z>y}(ackResponse(z,x,y) \wedge withintime(z,y,2s)))$
Syntax Tree:
Message
Request
Unknown
Response
INVITE
Unknown
ACK
Unknown
Success
Final Result:
No. Messages | Pass | Fail | Time-Fail
80000 | 18864 | 5 | 0
Step 8：Evaluation
Step 4: Setup
Tester:
Evaluation Unit
Buffer
Step 4：Generating Filter
Step 6(a)：Transferring
Step 7：Load Notification
IUT
Step 3：Capturing
Caputured Trace:
SIP 362 Request: ACK sip:ua2@CA.cym.com
SIP 360 Request: BYE sip:ua2@CA.cym.com
SIP 319 Status: 200 OK
SIP/SDF 528 Request: INVITE sip:ua2@CA.cym.com,
SIP 323 Status: 180 Ringing
SIP 322 Status: 100 Trying
SIP/SDF 480 Status: 200 OK, with session descri
SIP 362 Request: ACK sip:ua2@CA.cym.com
SIP 363 Request: BYE sip:ua2@CA.cym.com
SIP 322 Status: 200 OK
SIP 330 Request: REGISTER sip:CA.cym.com
SIP 333 Status: 200 OK (1 bindings)
SIP/SDF 528 Request: INVITE sip:ua2@CA.cym.com,
SIP 323 Status: 180 Ringing
SIP 322 Status: 100 Trying
SIP/SDF 480 Status: 200 OK, with session descri
SIP 362 Request: ACK sip:ua2@CA.cym.com
SIP/SDF 528 Request: INVITE sip:ua2@CA.cym.com,
SIP 323 Status: 180 Ringing
Step 5：Filtering
Filtered Trace:
Message (80000)
(45231) Request
Unknown
Response (34769)
(23115) INVITE
Unknown (3247)
(18869) ACK
Unknown (11745)
Success (23024)
Temporary Storage
30798 Messages
Step 6(b)：Buffering
Step 6(c)：Re-Transferring
Step 7：Load Notification

Figure 4. Use case for Testing Process

Properties	Total Messages	Filtered out Messages	Filtered out Rate	Pass	Fail	Time-Fail	Incon	Data-Inc
Prop.1	2324506	1631797	70.19%	631271	0	61432	52	2164
Prop.2	2324506	1631797	70.19%	498124	194579	0	52	2164
Prop.3	2324506	1979904	85.17%	314923	0	29673	14	1086
Prop.4	2324506	1979904	85.17%	247257	97339	0	14	1086
Prop.5	2324506	2259032	97.18%	61550	0	3924	6	371

TABLE I. Online Testing result for Properties

testers in a distributed environment. Thus, the efficiency and processing capacity of the approach will be scalably tested. Meanwhile, we will work on the optimization of our algorithms to severe situations in case of several distributed P.Os, and try to use SVM for predicting *Data-Inc* verdicts and thus to avoid non relevant situations.

REFERENCES

[1] F. Lalanne and S. Maag, "A formal data-centric approach for passive testing of communication protocols," in IEEE / ACM Transactions on Networking, vol. 21, no. 3, 2013, pp. 788–801.

[2] M. Veanes, C. Campbell, W. Schulte, and N. Tillmann, "Online testing with model programs," in Proceedings of the 10th European Software Engineering Conference, 2005, pp. 273–282.

[3] D. Lee and R. Miller, "Network protocol system monitoring-a formal approach with passive testing," IEEE/ACM Transactions on Networking, 2006, pp. 14(2):424–437.

[4] T.-D. Cao, P. Félix, R. Castanet, and I. Berrada, "Online testing framework for web services," in Third International Conference on Software Testing, Verification and Validation, 2010, pp. 363–372.

[5] A. C. Viana, S. Maag, and F. Zaïdi, "One step forward: Linking wireless self-organizing network validation techniques with formal testing approaches," ACM Comput. Surv., vol. 43, no. 2, 2011, p. 7.

[6] X. Che, F. Lalanne, and S. Maag, "A logic-based passive testing approach for the validation of communicating protocols," in Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering, 2012, pp. 53–64.

[7] X. Che and S. Maag, "A formal passive performance testing approach for distributed communication systems," in Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering, 2013, pp. 74–84.

[8] W. Wei, K. Suh, B. Wang, Y. Gu, J. F. Kurose, D. F. Towsley, and S. Jaiswal, "Passive online detection of 802.11 traffic using sequential hypothesis testing with tcp ack-pairs," IEEE Transactions on Mobile Computing, vol. 8, no. 3, 2009, pp. 398–412.

[9] E. Knauss and D. Damian, "V:issue:lizer: Exploring requirements clarification in online communication over time," in 35th International Conference on Software Engineering (ICSE), 2013, pp. 1327–1330.

[10] H. N. Nguyen, P. Poizat, and F. Zaïdi, "Online verification of value-passing choreographies through property-oriented passive testing," in 14th International IEEE Symposium on High-Assurance Systems Engineering, 2012, pp. 106–113.

[11] G. Morales, S. Maag, A. R. Cavalli, W. Mallouli, E. M. de Oca, and B. Wehbi, "Timed extended invariants for the passive testing of web services," in IEEE International Conference on Web Services (ICWS), 2010, pp. 592–599.

[12] G. Fraser, F. Wotawa, and P. Ammann, "Testing with model checkers: a survey," Software Testing and Verification Reliability, vol. 19, no. 3, 2009, pp. 215–261.

[13] F. Belli and M. Linschulte, "Event-driven modeling and testing of real-time web services," Service Oriented Computing and Applications, vol. 4, no. 1, 2010, pp. 3–15.

[14] F. Lalanne, X. Che, and S. Maag, "Data-centric property formulation for passive testing of communication protocols," in Proceedings of the 13th IASME/WSEAS, ser. ACC'11/MMACTEE'11, 2011, pp. 176–181.

[15] A. Horn, "On sentences which are true of direct unions of algebras," Journal of Symbolic Logic, vol. 16, no. 1, 1951, pp. 14–21.

[16] M. V. Emden and R. Kowalski, "The semantics of predicate logic as a programming language," Journal of the ACM, 1976, pp. 23(4):733–742.

[17] R. E. Noonan, "An algorithm for generating abstract syntax trees," Computer Languages, vol. 10, no. 3-4, 1985, pp. 225–236.

[18] M. R. Brown and R. E. Tarjan, "A fast merging algorithm," Journal of the ACM, vol. 26, no. 2, 1979, pp. 211–226.

[19] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, 1995, pp. 273–297.

[20] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, and J. Peterson, "Sip: Session initiation protocol," 2002.

[21] "Zoiper," http://www.zopier.com/softphone/, 2014.

[22] "Fonality," http://www.fonality.com, 2014.

Multiple-World Extension of Clausal Logical Structures

Kiyoshi Akama
Information Initiative Center
Hokkaido University
Sapporo, Hokkaido, Japan
Email: akama@iic.hokudai.ac.jp

Ekawit Nantajeewarawat
Computer Science Program
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand
Email: ekawit@siit.tu.ac.th

Tadayuki Yoshida
Faculty of Computer Science
Hokkaido University
Sapporo, Hokkaido, Japan
Email: tadayuki@sh.rim.or.jp

***Abstract*—In a simple clausal logical structure, a set of clauses determines models, each of which is a subset of some predetermined set $\mathcal{G}$ and represents a possible state of a world. A multiple-world clausal logical structure is an extension of a simple clausal logical structure, wherein multiple worlds are considered simultaneously. Each world model may be related not only to its corresponding set of clauses, but also to other world models. We define a method for multiple-world extension of a clausal logical structure. By applying this extension, we propose a new multiple-world clausal logical structure for S-expression atoms with definitions of several concrete constraints, which together provide rich expressive power including function variables and concepts such as argmax and negation.**

***Keywords*–Multiple-world logical structure; Referential constraint; Specialization system; Declarative description.**

I. Introduction

An abstract notion of a logical structure has been introduced since 2006 [1], by which the general concepts of descriptions (formulas), interpretations, models, and their interrelations are formalized axiomatically. Let $\mathcal{G}$ denote the set of all ground first-order atoms. First-order formulas constitute a logical structure, where interpretations and models are subsets of $\mathcal{G}$. Similarly, clause sets form another logical structure, where interpretations and models are again subsets of $\mathcal{G}$. A logical structure defines a relation between a description k and a model x of k, i.e., $x \in \mathit{Models}(k)$, where *Models* is a mapping that determines the set of all models of a given description.

Basic methods for construction of logical structures have been proposed by Akama and Nantajeewarawat [2], including generation of a logical structure from a specialization system, construction of a conjunctive logical structure, and logical structure morphing. Such construction methods are essential for formulating and understanding new logical structures and extending existing logical structures to obtain richer expressive power, computation efficiency, and computation completeness [3].

This paper proposes a new method for constructing logical structures, by which a logical structure for dealing with multiple worlds is constructed from simple clausal logical structures. This is typically explained by the following system of membership constraints:

$$x_{\mathrm{J}} \in \mathit{Models}(k_{\mathrm{J}}, x_{\mathrm{J}}, x_{\mathrm{H}})$$
$$x_{\mathrm{H}} \in \mathit{Models}(k_{\mathrm{H}}, x_{\mathrm{J}}, x_{\mathrm{H}})$$

They together represent a situation in which there are two persons, say John and Henry, and their beliefs, say x_{J} and x_{H}, respectively, are described by k_{J} and k_{H} together with the beliefs themselves.

There are two main important features:

1) There may be multiple worlds; a pair of a description and a set of possible models is called a *world*.
2) The mapping *Models* is used and it takes models as arguments.

By applying this construction method to an S-expression clausal logical structure, we propose a multiple-world clausal logical structure for S-expression atoms. We show that this extension provides rich expressive power.

For example, when we want to find all persons who earn maximum salary in a department, we may write the following clause:

$$(h\ {*name}) \leftarrow (\mathit{argmax}\ {*name}\ ({*n}\ {*s})\ ((\mathit{emp}\ {*n}\ {*a}\ {*s}\ {*d}))).$$

The predicate $argmax$ takes three arguments. The third one is in general a sequence of atoms. Its second and third arguments are used to represent a relation between names and salaries:

$$S = \{(n, s) | (\mathit{emp}\ n\ a\ s\ d)\}.$$

We need to maximize s by changing n in the set S. The predicate $argmax$ can not be defined by usual clauses. By the theory to be proposed in this paper, $argmax$ is defined as a new type of constraint. By adding a clause C, given by

$$C = ((p\ {*n}\ {*s}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ {*d})),$$

to a set *Cs* of clauses, the new multiple-world semantics proposed in this paper allows us to construct a constraint that refers to a model of $\mathit{Cs} \cup \{C\}$, and thus, to refer to its subset S.

Multiple-world extension of clausal logical structures is also useful for representing other database constraints such as max, min, summation, and average. Moreover negation is also represented by using the same semantics. Together with function variables, the proposed logical structure has a very rich expressive power that is purely declarative and it is rigorously formulated based on an abstract definition of a logical structure.

The rest of the paper is organized as follows: Section II recalls an abstract notion of a logical structure and its related basic concepts, and identifies the main objective of the paper. Section III provides a general notion of a specialization system, and defines a specialization system for S-expressions and that for function variables and function constants. Section IV formalizes constraints, defines declarative descriptions along with their models, associates with a declarative description a system of membership constraints, and formulates a logical structure for declarative descriptions. Section V shows rich expressive power of an S-expression-based logical structure, which is directly created by instantiation of the general theory, by defining constraints and *func*-atoms, which cannot be supplied by the conventional first-order logic. Section VI provides conclusions.

Preliminary Notation

The following notation will be used. For any sets A and B, *pow*(A) denotes the power set of A, *Map*(A, B) the set of all mappings from A to B, and *partialMap*(A) the set of all partial mappings on A (i.e., from A to A itself). Given a binary relation r, $dom(r)$ and $ran(r)$ denote the domain and the range of r, respectively. For any partial mappings f and g such that $dom(f) \supseteq ran(g)$, $f \circ g$ denotes the composition of f with g, i.e., for each $a \in dom(g)$, $(f \circ g)(a) = f(g(a))$. Let *Bool* $= \{$*true*,*false*$\}$.

II. Logical Structures and Basic Related Concepts

A. *Logical Structures*

An abstract notion of a logical structure [1] is recalled first.

Definition 1: Let $\mathcal{G}$ be a set. A *logical structure* $\mathcal{L}$ on $\mathcal{G}$ is a triple $\langle \mathcal{K}, \mathcal{I}, \nu \rangle$, where

1) $\mathcal{K}$ is a set,
2) $\mathcal{I} = pow(\mathcal{G})$,
3) $\nu : \mathcal{K} \to Map(\mathcal{I}, Bool)$.

An element of $\mathcal{K}$ is called a *description*. An element of $\mathcal{I}$ is called an *interpretation*. ■

Given an arbitrary set $\mathcal{G}$, a logical structure $\mathcal{L} = \langle \mathcal{K}, \mathcal{I}, \nu \rangle$ on $\mathcal{G}$ is a representation system. Intuitively, when we have a subset G of $\mathcal{G}$ in mind as a target atom set to be represented, we can provide information about the target atom set G using some description $k \in \mathcal{K}$, i.e., G satisfies the condition $\nu(k)(G) = true$.

The concept of an interpretation in Definition 1 is very different from an interpretation in the usual logic [4][5]. The set $\mathcal{G}$ in Definition 1 is more similar to the Herbrand Base [6] for a given description, but they are not exactly the same. The Herbrand Base depends on a given description (logical formula). There is no common set like $\mathcal{G}$ in the definition of the Herbrand Base. However, in our theory, a unique set $\mathcal{G}$ is firstly given. The objective of defining a logical structure is to define a representation system in which a description determines a set of possible subsets of $\mathcal{G}$.

Despite its simplicity, the abstract notion given by Definition 1 provides a sufficient structure for defining the concepts of logical equivalence, models, satisfiability, and logical consequence, which will be given in Definition 2.

Definition 2: Let $\mathcal{L} = \langle \mathcal{K}, \mathcal{I}, \nu \rangle$ be a logical structure. Two descriptions $k_1, k_2 \in \mathcal{K}$ are *logically equivalent* in $\mathcal{L}$, denoted by $k_1 \equiv_{\mathcal{L}} k_2$, iff $\nu(k_1) = \nu(k_2)$. An interpretation $I \in \mathcal{I}$ is a *model* in $\mathcal{L}$ of a description $k \in \mathcal{K}$ iff $\nu(k)(I) = true$. A description $k \in \mathcal{K}$ is *satisfiable* in $\mathcal{L}$ iff there exists a model in $\mathcal{L}$ of k. A description $k_1 \in \mathcal{K}$ is a *logical consequence* in $\mathcal{L}$ of a description $k_2 \in \mathcal{K}$, denoted by $k_2 \models_{\mathcal{L}} k_1$, iff every model in $\mathcal{L}$ of k_2 is a model in $\mathcal{L}$ of k_1. ■

For these concepts to have practical meanings, it is necessary to give appropriate structure to elements of $\mathcal{K}$ and $\mathcal{I}$ and also to the mapping ν. This process is called logical structure instantiation, which is illustrated below.

B. *Logical Structure Instantiation: Examples*

Instantiation of the abstract notion of a logical structure into first-order logic and into a clausal logical system is illustrated below.

1) First-Order Logic: Let $\mathcal{G}$ be the set of all variable-free atomic formulas (atoms) and $\mathcal{K}$ the set of all closed first-order formulas. Let ν be defined by: for any $k \in \mathcal{K}$ and any $G \subseteq \mathcal{G}$, $\nu(k)(G) = true$ iff k is true with respect to G. Then $\langle \mathcal{K}, pow(\mathcal{G}), \nu \rangle$ is a logical structure on $\mathcal{G}$.

2) Clausal Logical Systems: Let $\mathcal{G}$ be the set of all variable-free atoms and $\mathcal{K}$ the set of all sets of clauses. Let ν be defined by: for any $k \in \mathcal{K}$ and any $G \subseteq \mathcal{G}$, $\nu(k)(G) = true$ iff all clauses in k are true with respect to G. Then $\langle \mathcal{K}, pow(\mathcal{G}), \nu \rangle$ is a logical structure on $\mathcal{G}$.

C. *The Primary Objective of the Paper*

The objective of the paper is twofold:

1) To develop a method for multiple-world extension of logical structures (Section IV).
2) To introduce a concrete logical structure in the S-expression space by the instantiation of the proposed method (Section V).

The resulting concrete logical structure provides multiple-world knowledge representation based on an extended space with (i) clauses and iff-formulas, and with (ii) function variables and function constants.

III. Specialization Systems

A. *Specialization Systems*

The notion of a specialization system [7] is recalled below.

Definition 3: A *specialization system* Γ is a quadruple $\langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ of three sets $\mathcal{A}$, $\mathcal{G}$, and $\mathcal{S}$, and a mapping μ from $\mathcal{S}$ to *partialMap*$(\mathcal{A})$ that satisfies the following conditions:

1) $(\forall s', s'' \in \mathcal{S})(\exists s \in \mathcal{S}) : \mu(s) = \mu(s') \circ \mu(s'')$.
2) $(\exists s \in \mathcal{S})(\forall a \in \mathcal{A}) : \mu(s)(a) = a$.
3) $\mathcal{G} \subseteq \mathcal{A}$.

Elements of $\mathcal{A}$, $\mathcal{G}$, and $\mathcal{S}$ are called *atoms*, *ground atoms*, and *specializations*, respectively. The mapping μ is called the *specialization operator* of Γ. A specialization $s \in \mathcal{S}$ is said to be *applicable* to $a \in \mathcal{A}$ iff $a \in dom(\mu(s))$. ■

In the sequel, assume that a specialization system $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ is given. A specialization in $\mathcal{S}$ will often be denoted by a Greek letter such as θ. A specialization $\theta \in \mathcal{S}$ will be identified with the partial mapping $\mu(\theta)$ and used as a postfix unary (partial) operator on $\mathcal{A}$ (e.g., $\mu(\theta)(a) = a\theta$), provided that no confusion is caused. Let ϵ denote the identity specialization in $\mathcal{S}$, i.e., $a\epsilon = a$ for any $a \in \mathcal{A}$. For any $\theta, \sigma \in \mathcal{S}$, let $\theta \circ \sigma$ denote a specialization $\rho \in \mathcal{S}$ such that $\mu(\rho) = \mu(\sigma) \circ \mu(\theta)$, i.e., $a(\theta \circ \sigma) = (a\theta)\sigma$ for any $a \in \mathcal{A}$. A subset A' of $\mathcal{A}$ is said to be *closed* iff $a\theta \in A'$ for any $a \in A'$ and any $\theta \in \mathcal{S}$.

B. A Specialization System for S-Expressions

Next, a specialization system for S-expressions is introduced. It will be used for providing concrete examples of constraints and declarative descriptions in Section V.

An alphabet $\Delta = \langle \mathbf{K}, \mathbf{V} \rangle$ is assumed, where $\mathbf{K}$ is a countably infinite set of constants, $\mathbf{V}$ is a countably infinite set of variables, $\mathbf{K}$ and $\mathbf{V}$ are disjoint, and $nil \in \mathbf{K}$. Assume that $\mathbf{K}$ includes the set of all numbers. As notational conventions, each variable begins with an asterisk (e.g., $*x$), while constants do not.

An *S-expression* (*symbolic expression*) on Δ is defined inductively as follows:

1) A constant in $\mathbf{K}$ is an S-expression on Δ.
2) A variable in $\mathbf{V}$ is an S-expression on Δ.
3) If a and a' are S-expressions on Δ, then $(a|a')$ is an S-expression on Δ.

An S-expression $(a_1|(a_2|\cdots|(a_n|nil)\cdots))$ is often written as $(a_1\ a_2\ \cdots\ a_n)$. The S-expression *nil* is often written as $()$.

A *substitution* on Δ is a finite set of bindings $\{v_1/a_1, \ldots, v_n/a_n\}$ such that for each $i \in \{1, \ldots, n\}$, $v_i \in \mathbf{V}$ and a_i is an S-expression on Δ such that $v_i \neq a_i$.

A specialization system $\Gamma_{\mathrm{S}} = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ for S-expressions is defined by:

1) $\mathcal{A}$ is the set of all S-expressions on Δ.
2) $\mathcal{G}$ is the set of all variable-free S-expressions on Δ.
3) $\mathcal{S}$ is the set of all substitutions on Δ.
4) $\mu : \mathcal{S} \to partialMap(\mathcal{A})$ such that for any $a \in \mathcal{A}$ and any $s \in \mathcal{S}$, $\mu(s)(a)$ is the S-expression obtained from a by simultaneously applying all bindings in s to a.

C. A Specialization System for Function Variables and Function Constants

The conventional Skolemization is not a meaning-preserving transformation [6]. Meaning-preserving Skolemization was recently developed in [3], in which function variables are used in place of Skolem functions. For example,

$$(motherOf\ *x\ *y) \leftarrow \langle f_{func}, h_1, *y, *x \rangle$$

is used for representing the statement "every person has his/her mother", where h_1 is a function variable, which determines a mother $*x$ of $*y$ by $*x = f_1(*y)$ using some unknown function f_1 obtained by instantiation of h_1. The specialization system Γ_{F} is introduced below for function variables and their instantiations.

Let $\mathcal{FV}$ be the set of all function variables, and let

$$\mathcal{FC} = \bigcup_{n \in \mathbb{N}} Map(\mathcal{G}^n, \mathcal{G}),$$

where $\mathbb{N}$ is the set of all nonnegative integers. Elements of $\mathcal{FC}$ are called *function constants*. Let Γ_{F} be defined as a specialization system $\langle \mathcal{A}_{\mathrm{F}}, \mathcal{G}_{\mathrm{F}}, \mathcal{S}_{\mathrm{F}}, \mu_{\mathrm{F}} \rangle$ as follows:

1) $\mathcal{A}_{\mathrm{F}} = \mathcal{FV} \cup \mathcal{FC}$,
2) $\mathcal{G}_{\mathrm{F}} = \mathcal{FC}$,
3) $\mathcal{S}_{\mathrm{F}}$ the set of all substitutions on $\langle \mathcal{FV}, \mathcal{FV} \cup \mathcal{FC} \rangle$,
4) $\mu_{\mathrm{F}} : \mathcal{S}_{\mathrm{F}} \to partialMap(\mathcal{A}_{\mathrm{F}})$ such that for any $f \in \mathcal{A}_{\mathrm{F}}$ and any $s \in \mathcal{S}_{\mathrm{F}}$, $\mu(s)(f)$ is obtained from f by applying the substitution s to it.

IV. Declarative Descriptions and Their Models

A description, called a *declarative description*, is introduced in this section. Declarative descriptions form a knowledge representation scheme with the following characteristics: (i) multiple-world representation, (ii) clauses and iff-formulas, and (iii) function variables and function constants.

A. Constraints

Assume that $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ and $\Gamma_{\mathrm{c}} = \langle \mathcal{A}_{\mathrm{c}}, \mathcal{G}_{\mathrm{c}}, \mathcal{S}, \mu_{\mathrm{c}} \rangle$ are specialization systems with the set $\mathcal{S}$ of specializations in common. They are used below, along with the specialization system Γ_{F} (Section III-C), for defining constraints, clauses, and declarative descriptions.

A constraint is an expression such that the truth value of its ground instantiation is predetermined. In this paper, constraint is represented by using a tuple enclosed by a pair of angle brackets. A constraint of the simplest type uses only terms as its arguments, e.g., an equality constraint $\langle f_{eq}, t_1, t_2 \rangle$. We also use function variables/constants as arguments. For example $\langle f_{func}, h_0, t, t' \rangle$ is a constraint with a unary function variable h_0, and it is true iff $h_0(t) = t'$. The most important type of constraints for multiple-world semantics is the one that refers to models of some sets of clauses. For example, $\langle f_{not}, (p\ *x), l_2 \rangle$ is a *not*-constraint, which has a world label l_2 that refers to a model of a set of clauses. Intuitively, $\langle f_{not}, (p\ *x), l_2 \rangle$ means that $(p\ *x) \notin model(l_2)$, where $model(l_2)$ is a model of the world l_2.

Definition 4: Let L be a set of labels. A *constraint* on $\langle \Gamma, \Gamma_{\mathrm{c}}, L, \Gamma_{\mathrm{F}} \rangle$ is an $(m+1)$-tuple $\langle \phi, d_1, d_2, \ldots, d_m \rangle$, where

1) $m \geq 0$,
2) $\phi : (\mathcal{G}_{\mathrm{c}} \cup \mathcal{G} \cup pow(\mathcal{G}) \cup \mathcal{G}_{\mathrm{F}})^m \to Bool$, and
3) for each $i \in \{1, \ldots, m\}$, $d_i \in \mathcal{A}_{\mathrm{c}} \cup \mathcal{A} \cup L \cup \mathcal{A}_{\mathrm{F}}$.

A constraint $\langle \phi, d_1, d_2, \ldots, d_m \rangle$ on $\langle \Gamma, \Gamma_{\mathrm{c}}, L, \Gamma_{\mathrm{F}} \rangle$ is called a *referential constraint* if d_i is a label in L for some $i \in \{1, \ldots, m\}$. It is called a *simple constraint* otherwise. It is called a *ground constraint* iff for any $i \in \{1, \ldots, m\}$, $d_i \in \mathcal{G}_{\mathrm{c}} \cup \mathcal{G} \cup L \cup \mathcal{G}_{\mathrm{F}}$. Let $\mathrm{Con}(\Gamma, \Gamma_{\mathrm{c}}, L, \Gamma_{\mathrm{F}})$ and $\mathrm{GCon}(\Gamma, \Gamma_{\mathrm{c}}, L, \Gamma_{\mathrm{F}})$ denote the set of all constraints and that of all ground constraints, respectively, on $\langle \Gamma, \Gamma_{\mathrm{c}}, L, \Gamma_{\mathrm{F}} \rangle$. ■

A specialization $\theta \in \mathcal{S}$ is always applicable to labels and elements in $\mathcal{A}_{\rm F}$ and its application does not change them, i.e., for any label l and any $a \in \mathcal{A}_{\rm F}$, the results of applying θ to l and a, denoted by $l\theta$ and $a\theta$, are l and a, respectively, themselves. Given a constraint $c = \langle \phi, d_1, d_2, \ldots, d_m \rangle$ and a specialization $\theta \in \mathcal{S}$, (i) θ is applicable to c iff it is applicable to each of $d_1, d_2, \ldots, d_m$, and (ii) when applicable, the result of applying θ to c, denoted by $c\theta$, is defined by $c\theta = \langle \phi, d_1\theta, d_2\theta, \ldots, d_m\theta \rangle$. A specialization $\theta \in \mathcal{S}_{\rm F}$ is always applicable to elements of $\mathcal{A}_{\rm c} \cup \mathcal{A} \cup L$, and its application does not change them. A specialization $\theta \in \mathcal{S}_{\rm F}$ is always applicable to an element f in $\mathcal{A}_F$, and its application changes it into $f\theta$.

B. Clauses and Iff-Formulas

A declarative description proposed in this paper consists of clauses and iff-formulas. Some set of clauses can be equivalently and more concisely represented by an iff-formula with higher chance of transformation. For example, the set of three clauses

$$(q\ {*A}\ {*B}), (r\ {*B}) \leftarrow (p\ {*A}\ {*B}),$$
$$(p\ {*A}\ {*B}) \leftarrow (q\ {*A}\ {*B}),$$
$$(p\ {*A}\ {*B}) \leftarrow (r\ {*B})$$

can be represented by the iff-formula

$$(p * A * B) \leftrightarrow (\{(q\ {*A}\ {*B})\} \vee \{(r\ {*B})\}).$$

Definition 5: Let L be a set of labels. A *clause* C on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$ is an expression of the form

$$a_1, a_2, \ldots, a_m \leftarrow b_1, b_2, \ldots, b_n,$$

where (i) $m, n \geq 0$, (ii) for each $i \in \{1, \ldots, m\}$, a_i is an atom in $\mathcal{A}$, and (iii) for each $j \in \{1, \ldots, n\}$, b_j is an atom in $\mathcal{A}$ or a constraint on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$ (i.e., $b_j \in \mathcal{A} \cup \textsc{Con}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$). The set $\{a_1, a_2, \ldots, a_m\}$ is called the *left-hand side* of C, denoted by $\mathit{lhs}(C)$, and the set $\{b_1, b_2, \ldots, b_n\}$ is called the *right-hand side* of C, denoted by $\mathit{rhs}(C)$. The set $\mathit{rhs}(C) - \mathcal{A}$ is denoted by $\mathit{con}(C)$. C is called a *referential clause* if there exists $i \in \{1, \ldots, n\}$ such that b_i is a referential constraint. It is called a *simple clause* otherwise. It is called a *ground clause* iff (i) for each $i \in \{1, \ldots, m\}$, $a_i \in \mathcal{G}$, and (ii) for each $j \in \{1, \ldots, n\}$, $b_j \in \mathcal{G} \cup \text{G}\textsc{Con}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$. Let $\text{CL}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$ and $\text{GCL}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$ denote the set of all clauses and that of all ground clauses, respectively, on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$. Let $\text{GCL}(\mathcal{G})$ denote the set of all ground clauses that consist only of atoms in $\mathcal{G}$. ■

Given a clause $C = (a_1, a_2, \ldots, a_m \leftarrow b_1, b_2, \ldots, b_n)$ and a specialization $\theta \in \mathcal{S} \cup \mathcal{S}_{\rm F}$, (i) θ is applicable to C iff it is applicable to each of $a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n$, and (ii) when applicable, the result of applying θ to C, denoted by $C\theta$, is defined by $C\theta = (a_1\theta, a_2\theta, \ldots, a_m\theta \leftarrow b_1\theta, b_2\theta, \ldots, b_n\theta)$.

Definition 6: An *if-and-only-if formula* (for short, *iff-formula*) I on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$ is a formula of the form

$$a \leftrightarrow (\mathit{conj}_1 \vee \mathit{conj}_2 \vee \cdots \vee \mathit{conj}_n),$$

where (i) $n \geq 0$, (ii) $a \in \mathcal{A}$, and (iii) each of the conj_i is a finite subset of $\mathcal{A} \cup \textsc{Con}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$. For each $i \in \{1, \ldots, n\}$, conj_i corresponds to the conjunction $\bigwedge\{b \mid b \in \mathit{conj}_i\}$ of atoms and constraints in conj_i, and this conjunction is denoted by $\mathbf{F}(\mathit{conj}_i)$. The atom a is called the *head* of the iff-formula I, denoted by $\mathit{head}(I)$; n is called the *number of conjunctions* in I, denoted by $\#\mathit{Conj}(I)$; and for each $i \in \{1, \ldots, n\}$, conj_i is called the *ith conjunction in the right-hand side* of I, denoted by $\mathit{rhs}_i(I)$. The set of all constraints occurring in I is denoted by $\mathit{con}(I)$. I is called a *ground* iff-formula if a, $\mathit{conj}_1, \mathit{conj}_2, \ldots, \mathit{conj}_n$ are all ground. When emphasis is given to its head, an iff-formula whose head is an atom a is often referred to as $\mathit{iff}(a)$. An iff-formula $I = (a \leftrightarrow (\mathit{conj}_1 \vee \cdots \vee \mathit{conj}_n))$ corresponds to the universally quantified formula $\forall(a \leftrightarrow (\mathbf{F}(\mathit{conj}_1) \vee \cdots \vee \mathbf{F}(\mathit{conj}_n)))$, which is denoted by $\mathbf{F}(I)$. Let $\text{IFF}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$ and $\text{GIFF}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F})$ be the set of all iff-formulas and that of all ground iff-formulas, respectively, on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$. Let $\text{GIFF}(\mathcal{G})$ denote the set of all ground iff-formulas that consist only of atoms in $\mathcal{G}$. ■

If I is an iff-formula $a \leftrightarrow (\mathit{conj}_1 \vee \mathit{conj}_2 \vee \cdots \vee \mathit{conj}_n)$ on $\langle \Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F} \rangle$, then:

- For any $\theta \in \mathcal{S} \cup \mathcal{S}_{\rm F}$ and any $i \in \{1, \ldots, n\}$, if $\mathit{conj}_i = \{b_1, b_2, \ldots, b_k\}$, then (i) θ is applicable to conj_i iff it is applicable to each of $b_1, b_2, \ldots, b_k$, and (ii) when applicable, the result of applying θ to conj_i, denoted by $\mathit{conj}_i\theta$, is defined by $\mathit{conj}_i\theta = \{b_1\theta, b_2\theta, \ldots, b_k\theta\}$.
- For any $\theta \in \mathcal{S} \cup \mathcal{S}_{\rm F}$, (i) θ is applicable to I iff it is applicable to a and each of $\mathit{conj}_1, \mathit{conj}_2, \ldots, \mathit{conj}_n$, and (ii) when applicable, the result of applying θ to I, denoted by $I\theta$, is defined by $I\theta = (a\theta \leftrightarrow (\mathit{conj}_1\theta \vee \mathit{conj}_2\theta \vee \cdots \vee \mathit{conj}_n\theta)$.

C. Declarative Descriptions

A declarative description consists of clauses and iff-formulas. They may have labels inside referential constraints, by which we can have richer expressive power. For example, the constraint in the right-hand side of the iff-formula

$$l_3 : (\mathit{odd}\ {*x}) \leftrightarrow \{\langle \phi_{\mathit{not}}, (\mathit{even}\ {*x}), l_3 \rangle, (\mathit{int}\ {*x})\}$$

means that $*x$ is an odd number iff $*x$ is an integer that is not even.

Definition 7: Let $\mathbb{L}$ be a set of labels. A *declarative description* D on $\langle \Gamma, \Gamma_{\rm c}, \mathbb{L}, \Gamma_{\rm F} \rangle$ is a triple $\langle l, L, f \rangle$, where

1) $l \in L$,
2) $L \subseteq \mathbb{L}$, and
3) $f : L \rightarrow \mathit{pow}(\text{CL}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F}) \cup \text{IFF}(\Gamma, \Gamma_{\rm c}, L, \Gamma_{\rm F}))$.

D is called a *referential declarative description* if there exists a referential clause in the set $\bigcup\{f(l) \mid l \in L\}$. It is called a *simple declarative description* otherwise. The set of all declarative descriptions on $\langle \Gamma, \Gamma_{\rm c}, \mathbb{L}, \Gamma_{\rm F} \rangle$ is denoted by $\text{DD}(\Gamma, \Gamma_{\rm c}, \mathbb{L}, \Gamma_{\rm F})$. ■

D. Evaluation of Constraints

Let $D = \langle l_0, L, f\rangle$ be a declarative description on $\langle \Gamma, \Gamma_\mathrm{c}, \mathbb{L}, \Gamma_\mathrm{F}\rangle$ and $l_0, l_1, \ldots, l_k$ are all the labels in L, listed according to some predetermined order. Mappings

- $\mathit{val} : \mathrm{G}\textsc{Con}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F}) \times \mathit{pow}(\mathcal{G})^{k+1} \to \mathit{Bool}$,
- $\mathrm{T}\textsc{Con} : \mathit{pow}(\mathcal{G})^{k+1} \to \mathit{pow}(\mathrm{G}\textsc{Con}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F}))$

are introduced for evaluation of constraints and determination of the sets of all true constraints with respect to ground atom sets associated with $l_0, l_1, \ldots, l_k$. They are defined as follows:

1) For any constraint

$$c = \langle \phi, d_1, d_2, \ldots, d_m\rangle \in \mathrm{G}\textsc{Con}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F})$$

and any $G_0, G_1, \ldots, G_k \subseteq \mathcal{G}$,

$$\mathit{val}(c, G_0, G_1, \ldots, G_k) = \phi(g_1, g_2, \ldots, g_m),$$

where for each $i \in \{0, 1, \ldots, m\}$,

$$g_i = \begin{cases} d_i & \text{if } d_i \in \mathcal{G} \cup \mathcal{G}_\mathrm{c}, \\ G_j & \text{if } d_i = l_j \text{ for some } j \in \{0, 1, \ldots, k\}. \end{cases}$$

2) For any $G_0, G_1, \ldots, G_k \subseteq \mathcal{G}$, $\mathrm{T}\textsc{Con}(G_0, G_1, \ldots, G_k)$ is the set

$$\{c \in \mathrm{G}\textsc{Con}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F}) \\ \mid \mathit{val}(c, G_0, G_1, \ldots, G_k) = \mathit{true}\}.$$

E. Evaluation of Clauses and Iff-Formulas

Using the mapping $\mathrm{T}\textsc{Con}$, ground clause sets and ground iff-formula sets, respectively, are associated with the labels of D by the mappings

- $\mathit{gclSet} : L \times \mathit{pow}(\mathcal{G})^{k+1} \to \mathit{pow}(\mathrm{GCL}(\mathcal{G}))$,
- $\mathit{giffSet} : L \times \mathit{pow}(\mathcal{G})^{k+1} \to \mathit{pow}(\mathrm{GIFF}(\mathcal{G}))$,

which are defined as follows:

1) For any $j \in \{0, 1, \ldots, k\}$ and any $G_0, G_1, \ldots, G_k \subseteq \mathcal{G}$, $\mathit{gclSet}(l_j, G_0, G_1, \ldots, G_k)$ is the set

$$\begin{aligned} &\{C' \in \mathrm{GCL}(\mathcal{G}) \\ &\quad \mid (\exists C \in f(l_j) \cap \mathrm{CL}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F})) \\ &\qquad (\exists \theta \in \mathcal{S})(\exists \sigma \in \mathcal{S}_\mathrm{F}) : \\ &\qquad\quad (C\theta\sigma \in \mathrm{GCL}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F})) \ \& \\ &\qquad\quad (\mathit{con}(C\theta\sigma) \subseteq \mathrm{T}\textsc{Con}(G_0, G_1, \ldots, G_k)) \ \& \\ &\qquad\quad (\mathit{lhs}(C') = \mathit{lhs}(C\theta\sigma)) \ \& \\ &\qquad\quad (\mathit{rhs}(C') = \mathit{rhs}(C\theta\sigma) - \mathit{con}(C\theta\sigma))\}. \end{aligned}$$

2) For any $j \in \{0, 1, \ldots, k\}$ and any $G_0, G_1, \ldots, G_k \subseteq \mathcal{G}$, $\mathit{giffSet}(l_j, G_0, G_1, \ldots, G_k)$ is the set

$$\begin{aligned} &\{I' \in \mathrm{GIFF}(\mathcal{G}) \\ &\quad \mid (\exists I \in f(l_j) \cap \mathrm{IFF}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F})) \\ &\qquad (\exists \theta \in \mathcal{S})(\exists \sigma \in \mathcal{S}_\mathrm{F}) : \\ &\qquad\quad (I\theta\sigma \in \mathrm{GIFF}(\Gamma, \Gamma_\mathrm{c}, L, \Gamma_\mathrm{F})) \ \& \\ &\qquad\quad (\mathit{con}(I\theta\sigma) \subseteq \mathrm{T}\textsc{Con}(G_0, G_1, \ldots, G_k)) \ \& \\ &\qquad\quad (\mathit{head}(I') = \mathit{head}(I\theta\sigma)) \ \& \\ &\qquad\quad (\#\mathit{Conj}(I') = \#\mathit{Conj}(I)) \ \& \\ &\qquad\quad (\forall i \in \{1, 2, \ldots, \#\mathit{Conj}(I)\} : \\ &\qquad\qquad\quad \mathit{rhs}_i(I') = \mathit{rhs}_i(I\theta\sigma) - \mathit{con}(I\theta\sigma))\}. \end{aligned}$$

F. Models of Declarative Descriptions

Truth values of ground clauses in $\mathrm{GCL}(\mathcal{G})$ and ground iff-formulas in $\mathrm{GIFF}(\mathcal{G})$ are defined below. Let G be an arbitrary subset of $\mathcal{G}$. Then:

- A ground clause $C \in \mathrm{GCL}(\mathcal{G})$ is true with respect to G iff $\mathit{lhs}(C) \cap G \neq \emptyset$ or $\mathit{rhs}(C) \not\subseteq G$.
- A ground iff-formula $I \in \mathrm{GIFF}(\mathcal{G})$ is true with respect to G iff one of the following conditions is satisfied:
  1) $\mathit{head}(I) \in G$ and for some $i \in \{1, 2, \ldots, \#\mathit{Conj}(I)\}$, $\mathit{rhs}_i(I) \subseteq G$.
  2) $\mathit{head}(I) \notin G$ and for any $i \in \{1, 2, \ldots, \#\mathit{Conj}(I)\}$, $\mathit{rhs}_i(I) \not\subseteq G$.

Given these definitions, a mapping

$$\mathit{Models} : \mathit{pow}(\mathrm{GCL}(\mathcal{G}) \cup \mathrm{GIFF}(\mathcal{G})) \to \mathit{pow}(\mathit{pow}(\mathcal{G}))$$

is then defined by: For any $K \subseteq \mathrm{GCL}(\mathcal{G}) \cup \mathrm{GIFF}(\mathcal{G})$ and any $G \subseteq \mathit{pow}(\mathcal{G})$, $G \in \mathit{Models}(K)$ iff for any $k \in K$, k is true with respect to G.

Now let $D = \langle l_0, L, f\rangle$ be a declarative description on $\langle \Gamma, \Gamma_\mathrm{c}, \mathbb{L}, \Gamma_\mathrm{F}\rangle$ and $l_0, l_1, \ldots, l_k$ are all the labels in L, listed according to some predetermined order. For any $j \in \{0, 1, \ldots, k\}$, a mapping

$$\mathit{model}_j : \mathit{pow}(\mathcal{G})^{k+1} \to \mathit{pow}(\mathit{pow}(\mathcal{G}))$$

is defined as follows: For any $G_0, G_1, \ldots, G_k \subseteq \mathcal{G}$,

$$\mathit{model}_j(G_0, G_1, \ldots, G_k) = \mathit{Models}(K_j),$$

where K_j is the union of the following two sets:

- $\mathit{gclSet}(l_j, G_0, G_1, \ldots, G_k) \subseteq \mathrm{GCL}(\mathcal{G})$
- $\mathit{giffSet}(l_j, G_0, G_1, \ldots, G_k) \subseteq \mathrm{GIFF}(\mathcal{G})$

Using the mappings $\mathit{model}_0, \mathit{model}_1, \ldots, \mathit{model}_k$, the declarative description D determines a collection of the following membership constraints:

$$\begin{gathered} x_0 \in \mathit{model}_0(x_0, x_1, \ldots, x_k) \\ x_1 \in \mathit{model}_1(x_0, x_1, \ldots, x_k) \\ \cdots \\ x_k \in \mathit{model}_k(x_0, x_1, \ldots, x_k) \end{gathered}$$

This collection is called the *system of membership constraints* for D, denoted by $\mathrm{SMC}(D)$.

Definition 8: $[G, G_1, G_2, \ldots, G_k]$ is an extended model of D iff the set of equations $\{x_0 = G, x_1 = G_1, x_2 = G_2, \ldots, x_k = G_k\}$ satisfies $\mathrm{SMC}(D)$. ■

Definition 9: G is a model of D iff there exist $G_1, G_2, \ldots, G_k$ such that $[G, G_1, G_2, \ldots, G_k]$ is an extended model of D. ■

G. A Logical Structure for Declarative Descriptions

Using the class of declarative descriptions proposed so far, a logical structure $\mathcal{L} = \langle \mathcal{K}, \mathcal{I}, \nu \rangle$ for declarative descriptions on $\langle \Gamma, \Gamma_{\mathrm{c}}, \mathbb{L}, \Gamma_{\mathrm{F}} \rangle$ is defined as follows:

1) $\mathcal{K} = \mathrm{DD}(\Gamma, \Gamma_{\mathrm{c}}, \mathbb{L}, \Gamma_{\mathrm{F}})$.
2) $\mathcal{I} = \mathit{pow}(\mathcal{G})$.
3) $\nu : \mathcal{K} \to \mathit{Map}(\mathcal{I}, \mathit{Bool})$ is defined as follows: Let $L \subseteq \mathbb{L}$. Let $D = \langle l_0, L, f \rangle$ be a declarative description on $\langle \Gamma, \Gamma_{\mathrm{c}}, \mathbb{L}, \Gamma_{\mathrm{F}} \rangle$ and $l_0, l_1, \ldots, l_k$ are all the labels in L, where $l_1, \ldots, l_k$ are listed according to some predetermined order of labels in $\mathbb{L}$. Then for any $G \subseteq \mathcal{G}$, $\nu(D)(G) = \mathit{true}$ iff there exist $G_1, \ldots, G_k \subseteq \mathcal{G}$ such that $(x_0 = G; x_1 = G_1; \ldots; x_k = G_k)$ satisfies $\mathrm{SMC}(D)$.

V. Constraints in the S-Expression Space

Referring to the specialization system $\Gamma_{\mathrm{S}} = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ given in Section III-B, we define important constraints in the specific domain of S-expressions. We also show that *func*-atoms can be realized in the same class of constraints. In the rest of this paper,

- we assume that $\Gamma = \Gamma_{\mathrm{c}} = \Gamma_{\mathrm{S}}$, and we consider $\mathrm{DD}(\Gamma_{\mathrm{S}}, \Gamma_{\mathrm{S}}, \mathbb{L}, \Gamma_{\mathrm{F}})$;
- let $H = \mathcal{G}_{\mathrm{c}} \cup \mathcal{G} \cup \mathit{pow}(\mathcal{G}) \cup \mathcal{G}_{\mathrm{F}}$.

A. Reference to a Database

The iff-formula in Fig. 1 represents an employee table consisting of four columns, i.e., name, age, salary, and department, where *eq*-atoms are defined by the set of iff-formulas

$$\begin{aligned}&\{((\mathit{eq}\ \bar{a}\ \bar{b}) \leftrightarrow (\mathit{true})) \mid \\ &\quad (\bar{a} \text{ and } \bar{b} \text{ are variable-free S-expressions in } \mathcal{G})\ \& \\ &\quad (\bar{a} = \bar{b})\}.\end{aligned}$$

Reference from a world to a table in another world is realized by the use of a referential constraint. For example, for referring to the 4th argument (i.e., the department column) of employee table above, the constraint $\langle \phi_{\mathrm{refer}}, l_0, \mathit{emp}, 4, *a4 \rangle$ can be used by assuming a mapping $\phi_{\mathrm{refer}} : H^4 \to \mathit{Bool}$ given by: $\phi_{\mathrm{refer}}(G, t_1, t_2, t_3) = \mathit{true}$ iff $G \subseteq \mathcal{G}$, $t_1, t_2, t_3 \in \mathcal{G}_{\mathrm{c}}$, and $(\mathit{refer}\ t_1\ t_2\ t_3) \in G$, where *refer* is defined by the clause

$$l_0\text{:}\ \ (\mathit{refer}\ \mathit{emp}\ 4\ {*d}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ {*d}).$$

Several kinds of constraints can be constructed for formulating aggregate operations concerning a referenced database, including operations that are commonly supported by a standard SQL, e.g., max, min, average, and sum [8][9][10], and those that are not, e.g., argmax and argmin. Formulation of the sum, argmax, and argmin operations using constraints is illustrated below.

$$\begin{aligned}l_0\text{:}\ \ &(\mathit{emp}\ {*n}\ {*a}\ {*s}\ {*d}) \leftrightarrow \\ &(\{(\mathit{eq}\ {*n}\ \mathit{John}), (\mathit{eq}\ {*a}\ 34), (\mathit{eq}\ {*s}\ 8600), (\mathit{eq}\ {*d}\ d1)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Morgan}), (\mathit{eq}\ {*a}\ 24), (\mathit{eq}\ {*s}\ 6300), (\mathit{eq}\ {*d}\ d2)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Lewis}), (\mathit{eq}\ {*a}\ 42), (\mathit{eq}\ {*s}\ 9000), (\mathit{eq}\ {*d}\ d3)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Long}), (\mathit{eq}\ {*a}\ 34), (\mathit{eq}\ {*s}\ 8500), (\mathit{eq}\ {*d}\ d2)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Henry}), (\mathit{eq}\ {*a}\ 29), (\mathit{eq}\ {*s}\ 12300), (\mathit{eq}\ {*d}\ d1)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Thomas}), (\mathit{eq}\ {*a}\ 31), (\mathit{eq}\ {*s}\ 7300), (\mathit{eq}\ {*d}\ d3)\} \vee \\ &\ \{(\mathit{eq}\ {*n}\ \mathit{Martin}), (\mathit{eq}\ {*a}\ 45), (\mathit{eq}\ {*s}\ 7500), (\mathit{eq}\ {*d}\ d1)\})\end{aligned}$$

Figure 1. Representing an employee table

B. Summation

Summation can be formulated using a constraint defined by the mapping

$$\phi_{\mathit{sum}} : H^3 \to \mathit{Bool},$$

where $\phi_{\mathit{sum}}(id, G, t) = \mathit{true}$ iff t is the sum of the elements of the multi-set $\{s \mid (\mathit{sum}\ id\ n\ s) \in G\}$. Using ϕ_{sum}, the query "find the sum of salaries of all the employees who work in the department $d1$ or the department $d2$" is represented by:

$$\begin{aligned}l_0\text{:}\ \ &(\mathit{ans}\ \ {*x}) \leftarrow \langle \phi_{\mathit{sum}}, id, l_0, {*x} \rangle. \\ l_0\text{:}\ \ &(\mathit{sum}\ id\ {*n}\ {*s}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ d1). \\ l_0\text{:}\ \ &(\mathit{sum}\ id\ {*n}\ {*s}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ d2).\end{aligned}$$

C. Argmax and Argmin

Argmax stands for the argument of the maximum. Given a mapping f, argmax gives a value x that maximizes $f(x)$. The argmax constraint is defined using

$$\phi_{\mathit{argmax}} : H^3 \to \mathit{Bool},$$

given by: $\phi_{\mathit{argmax}}(id, G, t) = \mathit{true}$ iff $id \in \mathcal{G}_{\mathrm{c}}$, $G \subseteq \mathcal{G}$, $t \in \mathcal{G}_{\mathrm{c}}$, and there exists $t' \in \mathcal{G}_{\mathrm{c}}$ such that

1) $(\mathit{argmax}\ id\ t\ t') \in G$, and
2) for any $t_1, t_2 \in \mathcal{G}_{\mathrm{c}}$, if $(\mathit{argmax}\ id\ t_1\ t_2) \in G$, then $t_2 \leq t'$.

For instance, the query "find all persons who earn the maximum salary" is represented by:

$$\begin{aligned}l_0\text{:}\ \ &(\mathit{ans}\ {*x}) \leftarrow \langle \phi_{\mathit{argmax}}, id, l_0, {*x} \rangle. \\ l_0\text{:}\ \ &(\mathit{argmax}\ id\ {*n}\ {*s}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ {*d}).\end{aligned}$$

Similarly, given a mapping f, argmin gives a value x that minimizes $f(x)$. The argmin constraint is defined using

$$\phi_{\mathit{argmin}} : H^3 \to \mathit{Bool},$$

given by: $\phi_{\mathit{argmin}}(id, G, t) = \mathit{true}$ iff $id \in \mathcal{G}_{\mathrm{c}}$, $G \subseteq \mathcal{G}$, $t \in \mathcal{G}_{\mathrm{c}}$, and there exists $t' \in \mathcal{G}_{\mathrm{c}}$ such that

1) $(\mathit{argmin}\ id\ t\ t') \in G$, and
2) for any $t_1, t_2 \in \mathcal{G}_{\mathrm{c}}$, if $(\mathit{argmin}\ id\ t_1\ t_2) \in G$, then $t_2 \geq t'$.

The query "find all the youngest persons who work at the department $d1$," for example, is represented by:

$$\begin{aligned}l_0\text{:}\ \ &(\mathit{ans}\ {*x}) \leftarrow \langle \phi_{\mathit{argmin}}, id, l_0, {*x} \rangle. \\ l_0\text{:}\ \ &(\mathit{argmin}\ id\ {*n}\ {*a}) \leftarrow (\mathit{emp}\ {*n}\ {*a}\ {*s}\ d1).\end{aligned}$$

D. *func-atoms*

Atoms of a special kind, called *func*-atoms, are used for meaning-preserving Skolemization [3]. They can be represented by constraints with function variables and function constants as arguments. A mapping

$$\phi_{\mathit{func}} : \mathbb{FC} \times \mathcal{G}_{\mathrm{c}}^{n+1} \to \mathit{Bool}$$

is defined by: $\phi_{\mathit{func}}(f, t_1, t_2, \ldots, t_n, t_{n+1}) = \mathit{true}$ iff f is an n-ary function constant and $t_1, t_2, \ldots, t_n, t_{n+1}$ are variable-free terms such that $f(t_1, t_2, \ldots, t_n) = t_{n+1}$.

Let a first-order formula F be given by:

$$F\colon\ \exists x\colon (\mathit{hasChild}(\mathit{Peter}, x) \wedge (\exists y\colon \mathit{motherOf}(x, y)))$$

By meaning-preserving Skolemization [3], F is converted into $\{C_1, C_2\}$, given as follows:

C_1: $(\mathit{hasChild}\ \mathit{Peter}\ {*x}) \leftarrow \langle \phi_{\mathit{func}}, {*h_1}, {*x} \rangle$
C_2: $(\mathit{motherOf}\ {*x}\ {*y}) \leftarrow \langle \phi_{\mathit{func}}, {*h_1}, {*x} \rangle, \langle \phi_{\mathit{func}}, {*h_2}, {*y} \rangle$

E. *Negation*

Negation [8] can also be represented as constraints using

$$\phi_{\mathit{not}} : \mathcal{G} \times \mathit{pow}(\mathcal{G}) \to \mathit{Bool},$$

given by: for any $g \in \mathcal{G}$ and any $G \subseteq \mathcal{G}$, $\phi_{\mathit{not}}(g, G) = \mathit{true}$ iff $g \notin G$.

Example 1: Assume that $L = \{l_0\}$ and a is an arbitrary atom. Let definite clauses C_1 and C_2 be given by:

C_1 : $a \leftarrow (\mathit{not}\ a)$
C_2 : $(\mathit{not}\ a) \leftarrow \langle \phi_{\mathit{not}}, a, l \rangle$

Let D be a declarative description $\langle l_0, L, f \rangle$, where $f(l_0) = \{C_1, C_2\}$. Let $G = \mathit{rep}(a) \cup \mathit{rep}(\mathit{not}(a))$. We show that G is a model of D as follows: Let $C_1' = (g \leftarrow (\mathit{not}\ g))$ be any ground instance of C_1. C_1' is true since $g \in G$. So C_1 is true with respect to G. Let $C_2' = ((\mathit{not}\ g) \leftarrow \langle \phi_{\mathit{not}}, g, l_0 \rangle)$ be any ground instance of C_2. C_2' is true since $(\mathit{not}\ g) \in G$. So C_2 is true with respect to G. ■

Example 2: Assume again that $L = \{l_0\}$ and a is an arbitrary atom. Let I_1 and I_2 be the following iff-formulas:

I_1 : $a \leftrightarrow \{(\mathit{not}\ a)\}$
I_2 : $(\mathit{not}\ a) \leftrightarrow \{\langle \phi_{\mathit{not}}, a, l \rangle\}$

Let D be a declarative description $\langle l_0, L, f \rangle$, where $f(l_0) = \{I_1, I_2\}$. We show that D is not satisfiable as follows: Assume that G is a model of D. Let $\theta \in \mathcal{S}$ such that $a\theta \in G$ and let $a\theta = g$. From $I_1\theta = (g \leftrightarrow \{(\mathit{not}\ g)\})$, we have: $g \in G$ iff $(\mathit{not}\ g) \in G$. From $I_2\theta = ((\mathit{not}\ g) \leftrightarrow \{\langle \phi_{\mathit{not}}, g, l_0 \rangle\})$, we have: $(\mathit{not}\ g) \in G$ iff $\phi_{\mathit{not}}(g, l_0) = \mathit{true}$. From the definition of ϕ_{not}, $\phi_{\mathit{not}}(g, l_0) = \mathit{true}$ iff $g \notin G$. Then $g \in G$ iff $g \notin G$, which is a contradiction. ■

VI. Conclusions

We have defined a method for multiple-world extension of a clausal logical structure, and by applying this extension, we have introduced a multiple-world clausal logical structure for S-expression atoms. We can consider multiple worlds simultaneously. Not only usual terms but also models themselves can be considered as arguments of a relation. This class of declarative descriptions can be used for representing open worlds, closed worlds, and their mixtures, by using clauses and iff-formulas. This gives one solution for unifying open and closed world. Function variables in constraints enables us to use *func*-atoms, which are essential for meaning-preserving Skolemization [3].

Acknowledgment

This research was partially supported by JSPS KAKENHI Grant Number 25280078, and the Collaborative Research Program 2013, Information Initiative Center, Hokkaido University.

References

[1] K. Akama and E. Nantajeewarawat, "Logical Structures on Specialization Systems: Formalization and Satisfiability-Preserving Transformation," in Proceedings of the 7th International Conference on Intelligent Technologies, Taipei, Taiwan, 2006, pp. 100–109.

[2] ——, "Construction of Logical Structures on Specialization Systems," in Proceedings of the 2011 World Congress on Information and Communication Technologies, Mumbai, India, 2011, pp. 1030–1035.

[3] ——, "Meaning-Preserving Skolemization," in Proceedings of the 2011 International Conference on Knowledge Engineering and Ontology Development, Paris, France, 2011, pp. 322–327.

[4] M. Fitting, First-Order Logic and Automated Theorem Proving, 2nd ed. Springer-Verlag, 1996.

[5] J. W. Lloyd, Foundations of Logic Programming, second, extended ed. Springer-Verlag, 1987.

[6] C.-L. Chang and R. C.-T. Lee, Symbolic Logic and Mechanical Theorem Proving. Academic Press, 1973.

[7] K. Akama, "Declarative Semantics of Logic Programs on Parameterized Representation Systems," Advances in Software Science and Technology, vol. 5, 1993, pp. 45–63.

[8] S. Abiteboul, R. Hull, and V. Vianu, Foundations of Databases. Addison-Wesley, 1995.

[9] M. Dahr, Deductive Databases: Theory and Applications. Coriolis Group, 1996.

[10] C. J. Date, SQL and Relational Theory, 2nd Edition. O'Reilly Media, 2011.

The Zero-Sum Tensor

Mikael Fridenfalk
Department of Game Design
Uppsala University
Visby, Sweden
mikael.fridenfalk@speldesign.uu.se

***Abstract*—The zero-sum matrix, or in general, tensor, reveals some consistent properties at multiplication. In this paper, three mathematical rules are derived for multiplication involving such entities. The application of these rules may provide for a more concise and straightforward way to formulate mathematical proofs that rely on such matrices.**

Keywords-matrix; multiplication; n-simplex; tensor; zero-sum

I. Introduction

On the topic of rare matrices, some properties of a matrix, here defined as a *zero-sum matrix*, are analyzed and three rules are derived governing multiplication involving such matrices. The suggested category (the zero-sum matrix) does not seem to presently exist, and is as expected neither included in lists, such as [3]. In this paper, a *zero-sum matrix* is defined as a matrix where the sum of the column vectors is equal to a zero column vector and/or the sum of the row vectors is equal to a zero row vector, or in the general case, a zero-sum tensor of size $N_1 \times N_2 \times \cdots \times N_Q$, where summation along one, or several dimensions, results in a P-dimensional tensor (with $P = Q - 1$), that consists of zero-elements only. This rule applies to any tensor T of dimension $Q \in \mathbb{N}_2$ (all integers equal or greater than two). A matrix where the sum of the columns and rows both are equal to zero vectors, could further be defined as a *complete* zero-sum matrix, and similarly in the general case, a *complete* zero-sum tensor could be defined as a tensor where summation along all dimensions results in a P-dimensional tensor that consists only of zero-elements.

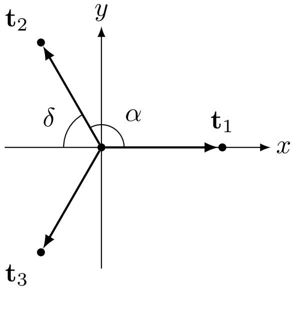


Figure 1. An example with a 2-simplex matrix $\mathbf{T}_2 = [\mathbf{t}_1\ \mathbf{t}_2\ \mathbf{t}_3]^T$, with the dihedral angle $\delta = \pi - \alpha$.

An example of a zero-sum matrix is a regular n-simplex matrix, based on the n-dimensional geometric object called the n-simplex. A few examples are the 0-simplex (point), the 1-simplex (line segment), the 2-simplex (triangle) and the 3-simplex (tetrahedron). If the object is fully symmetric (all edges are of equal length), it is called *regular*. Scaled appropriately, the regular n-simplex exhibits the following properties:

$$\mathbf{t}_i \cdot \mathbf{t}_j = \begin{cases} 1, & i = j \\ -1/n, & i \neq j \end{cases} \tag{1}$$

$$\sum_{i=1}^{n+1} \mathbf{t}_i = \mathbf{0} \tag{2}$$

where $\mathbf{t}_i$ and $\mathbf{t}_j$ with $i, j \in \{1, 2, \ldots, N\}$ and $N = n + 1$ denote any unit vectors i and j pointing from the center of the regular n-simplex to its i:th and j:th vertices. These properties were confirmed in [4] and [6] in context with an elementary mathematical proof of the relation $\delta = \arccos(\frac{1}{n})$, where δ denotes the dihedral angle of the regular n-simplex. For $n = 1$, $t_1 = -t_2 = 1$. For $n = 2$, as shown in Fig. 1:

$$\mathbf{t}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{t}_2 = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad \mathbf{t}_3 = \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{3}$$

In this example, the vectors $\mathbf{t}_i$ spanning the coordinate system of the regular n-simplex, are placed so that $\mathbf{t}_1$ coincides with the x-axis.

As a brief overview, we start by the derivation of three mathematical rules, followed by an example for the demonstration of Rule III (which is slightly more complex than the other two), and finally conclude, by the application of Rule III to reconfirm an already existing mathematical proof.

II. General Case

The idea behind the derivation of the rules presented in this paper originated from the evaluation of $\mathbf{H} = \mathbf{T}^T\mathbf{T}$ in [1], with the proposition of the extension of minimax [5], and alpha-beta pruning [2], from the two-person case to the general N-person case, which as a side effect led to the discovery of a new elementary method for the calculation of the dihedral angle of the regular n-simplex. The relation in (2), was used in this context by Fridenfalk [1], to derive a generic algorithm for the recursive calculation of $\mathbf{T} = [\,\mathbf{t}_1\,\mathbf{t}_2 \ldots \mathbf{t}_N\,]$, for $N \in \mathbb{N}_2$, with $N = n + 1$:

$$\left.\begin{aligned} t_{ii} &= \sqrt{1 - \sum_{j=1}^{i-1} \gamma_j^2} \\ \gamma_i &= -\frac{t_{ii}}{n + 1 - i} \end{aligned}\right\} 1 \leq i \leq n \tag{4}$$

ISBN: 978-1-61208-354-4

$$\mathbf{T} = \begin{bmatrix} 1 & \gamma_1 & \gamma_1 & \cdots & \gamma_1 & \gamma_1 & \gamma_1 & \gamma_1 \\ 0 & t_{22} & \gamma_2 & \cdots & \gamma_2 & \gamma_2 & \gamma_2 & \gamma_2 \\ 0 & 0 & t_{33} & \cdots & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & t_{(n-2)(n-2)} & \gamma_{n-2} & \gamma_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & \cdots & 0 & t_{(n-1)(n-1)} & \gamma_{n-1} & \gamma_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & t_{nn} & \gamma_n \end{bmatrix} \tag{5}$$

An alternative and concise proof of the relation in (2) follows by the derivation of Rule III in this paper. Before the presentation of this rule, we start by the establishment of two basic rules.

Rule I. Given the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{H}$ of size $M \times N$, $N \times K$, and $M \times K$, respectively, such that $\mathbf{H} = \mathbf{AB}$, if sum $\mathbf{a}$ of the row vectors of $\mathbf{A}$, is equal to a zero row vector $\mathbf{0}_N^T$ of size $1 \times N$, then sum $\mathbf{u}$ of the row vectors of $\mathbf{H}$, is equal to a zero row vector $\mathbf{0}_K^T$ of size $1 \times K$.

Proof. Given:

$$\mathbf{u} = \begin{bmatrix} b_{11}(a_{11}+\ldots+a_{M1})+\ldots+b_{N1}(a_{1N}+\ldots+a_{MN}) \\ b_{12}(a_{11}+\ldots+a_{M1})+\ldots+b_{N2}(a_{1N}+\ldots+a_{MN}) \\ \vdots \\ b_{1K}(a_{11}+\ldots+a_{M1})+\ldots+b_{NK}(a_{1N}+\ldots+a_{MN}) \end{bmatrix}^T \tag{6}$$

$\mathbf{a} = \mathbf{0}_N^T \to \mathbf{u} = \mathbf{0}_K^T$.

■

Rule II. Given the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{H}$ of size $M \times N$, $N \times K$, and $M \times K$, respectively, such that $\mathbf{H} = \mathbf{AB}$, if sum $\mathbf{b}$ of the column vectors of $\mathbf{B}$ is equal to a zero column vector $\mathbf{0}_N$ of size $N \times 1$, then sum $\mathbf{v}$ of the column vectors of $\mathbf{H}$ is equal to $\mathbf{0}_M$ of size $M \times 1$.

Proof 1. Given:

$$\mathbf{v} = \begin{bmatrix} a_{11}(b_{11}+\ldots+b_{1K})+\ldots+a_{1N}(b_{N1}+\ldots+b_{NK}) \\ a_{21}(b_{11}+\ldots+b_{1K})+\ldots+a_{2N}(b_{N1}+\ldots+b_{NK}) \\ \vdots \\ a_{M1}(b_{11}+\ldots+b_{1K})+\ldots+a_{MN}(b_{N1}+\ldots+b_{NK}) \end{bmatrix} \tag{7}$$

$\mathbf{b} = \mathbf{0}_N \to \mathbf{v} = \mathbf{0}_M$.

■

Proof 2. Given Rule I and the rules for matrix transpose, $\mathbf{H} = \mathbf{AB} \Leftrightarrow \mathbf{H}^T = \mathbf{B}^T\mathbf{A}^T$, thus, $\mathbf{b} = \mathbf{0}_N \to \mathbf{v} = \mathbf{0}_M$.

■

Rule III. Given the real matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{H}$ of size $M \times N$, $N \times M$, and $M \times M$, respectively, such that $\mathbf{H} = \mathbf{AB}$, if $\mathbf{A} = \mathbf{B}^T$, $\mathbf{v}$, defined as the sum of the column vectors of the symmetric matrix $\mathbf{H}$, is equal to a zero column vector $\mathbf{0}_M$ of size $M \times 1$, then $\mathbf{b}$, defined as the sum of the column vectors of $\mathbf{B}$, is equal to a zero column vector $\mathbf{0}_M$ of size $M \times 1$.

Proof. Given (7), $\mathbf{1}_M = [1 \quad 1 \quad \ldots \quad 1]^T$ of size $M \times 1$ and s, a positive-definite scalar, equal to the sum of the elements of the symmetric matrix $\mathbf{H}$:

$$s = \mathbf{1}_M^T \mathbf{H} \cdot \mathbf{1}_M = \mathbf{1}_M^T \mathbf{v} \tag{8}$$

as $a_{kj} = b_{jk}$ and $\mathbf{v} = \mathbf{0}_M \to s = \sum_{j=1}^{N} (b_{j1} + b_{j2} + \ldots + b_{jM})^2 = 0 \to \mathbf{b} = \mathbf{0}_N$, since:

$$s = \mathbf{1}_M^T \mathbf{v} = 0 \Rightarrow \begin{cases} b_{11} + b_{12} + \ \ldots \ + b_{1M} = 0 \\ b_{21} + b_{22} + \ \ldots \ + b_{2M} = 0 \\ \vdots \\ b_{N1} + b_{N2} + \ldots + b_{NM} = 0 \end{cases} \tag{9}$$

Thus, $\mathbf{v} = \mathbf{0}_M \to \mathbf{b} = \mathbf{0}_N$.

■

Once $\mathbf{u}$ and $\mathbf{v}$ are derived, as shown in (6)-(7), the derivation of the first two rules is straightforward. To concretize, the following example demonstrates the third rule for a 2×3 matrix, $\mathbf{B} = \mathbf{A}^T$. Given:

$$\mathbf{B} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \tag{10}$$

If $\mathbf{H} = \mathbf{AB}$, $\mathbf{1}_3 = [1 \quad 1 \quad 1]^T$ and $s = \mathbf{1}_3^T \mathbf{H} \cdot \mathbf{1}_3 = \mathbf{1}_3^T \mathbf{v}$, then:

$$\begin{aligned} s = \mathbf{1}_3^T & \begin{bmatrix} a(a+b+c) + d(d+e+f) \\ b(a+b+c) + e(d+e+f) \\ c(a+b+c) + f(d+e+f) \end{bmatrix} \\ &= (a+b+c)^2 + (d+e+f)^2 \end{aligned} \tag{11}$$

Thus, $\mathbf{v} = \mathbf{0}_3 \to s = (a+b+c)^2 + (d+e+f)^2 = 0 \to \mathbf{b} = \mathbf{0}_2$, since $s = 0 \to a+b+c = d+e+f = 0$. Or in other words, if $\mathbf{H} = \mathbf{B}^T\mathbf{B}$ is a 3×3 zero-sum matrix, then the sum of the columns of $\mathbf{B}$ is a zero-vector of size 2×1.

III. Application

As an example of the application of Rule III, given (1), a new and a more straightforward proof of (2) is hereby produced:

Proposition. The sum of the unit vectors $\mathbf{t}_i$ of a regular n-simplex, where each vector i points from the center of the object to its i:th vertex, is equal to $\mathbf{0}$.

Proof. Given a n-simplex matrix $\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ldots \mathbf{t}_{n+1}]$, $\mathbf{H} = \mathbf{T}^T\mathbf{T}$ (of size $n+1 \times n+1$), and:

$$h_{ij} = \mathbf{t}_i \cdot \mathbf{t}_j = \begin{cases} 1 & i = j \\ -1/n & i \neq j \end{cases} \tag{12}$$

given (1), where h_{ij} denotes an element in $\mathbf{H}$. Thus, the sum of any row (or column) in $\mathbf{H}$ is equal to $1 - \frac{1}{n} \cdot n = 0$, and since $\mathbf{H}$ is a (symmetric) zero-sum matrix, according to Rule III, $\sum_{i=1}^{n+1} \mathbf{t}_i = \mathbf{0}$.

■

IV. Conclusion

In this paper, a *zero-sum matrix* (or tensor) has been closely defined, along with the related concept *complete*. Three rules have been presented governing multiplications involving such entities, along with an example of the application of Rule III for a concise reconfirmation of (2), exemplified in this paper by a proposition.

References

[1] M. Fridenfalk, Method for Optimal N-Person Extension of Minimax and Alpha-Beta Pruning, Patent Pending, no. SE1230120-6, November, 2012.

[2] D. E. Knuth and R. W. Moore, "An Analysis of Alpha-Beta Pruning", Artificial Intelligence, vol. 6, 1975, pp. 293-326.

[3] MathWorld, Wolfram Research, Inc. <http://mathworld.wolfram.com/topics/MatrixTypes.html> [retrieved: March, 2014].

[4] H. R. Parks and D. C. Wills, "An Elementary Calculation of the Dihedral Angle of the Regular n-Simplex", The American Mathematical Monthly, vol. 109, no. 8, 2002, pp. 756-758.

[5] J. von Neumann, "On the theory of games" (in German: "Zur Theorie der Gesellschaftsspiele"), Mathematische Annalen, vol. 100, 1928, pp. 295-320.

[6] D. C. Wills, Connections Between Combinatorics of Permutations and Algorithms and Geometry, Doctoral Thesis, Oregon State University, 2009.

Mobile Edge Computing: Challenges for Future Virtual Network Embedding Algorithms

Michael Till Beck, Marco Maier
Ludwig-Maximilians-Universität München
München, Germany
{michael.beck,marco.maier}@ifi.lmu.de

***Abstract*—Mobile edge computing aims at reducing network latency and network stress by deploying mobile applications at the network edge. This paper proposes network virtualization in the context of mobile edge computing networks as an enabler for future, flexible and shared network infrastructures (IaaS). Network virtualization leads to the Virtual Network Embedding (VNE) problem, which aims at deploying virtual networks onto a shared, physical infrastructure. This paper is a position paper discussing new challenges for future VNE algorithms. To this end, new parameters specific to the mobile edge computing scenario are analyzed which are not considered by state-of-the-art VNE algorithms. Furthermore, novel and edge-specific VNE optimization objectives are derived.**

Keywords—Virtual Network Embedding, Next-gen Cellular Networks, Edge Computing

I. Introduction

Starting with the introduction of the first smartphones, one of the most apparent challenges for mobile network operators is handling future bandwidth demands, which are expected to further increase dramatically over the next years [1].

Mobile data traffic is predicted to continue doubling each year. Video contributes heavily to overall mobile network traffic. The usage of mobile applications (apps) accessing services deployed in the Internet is expected to further contribute to this trend, resulting in a growth of around 12 times by 2018 [2]. This trend becomes even more remarkable as novel mobile devices (like Google Glass and other, wearable devices) and applications arise. Offering more and more hardware capabilities, these devices pave the way for novel application types like augmented reality [2].

Network operators spend enormous efforts to keep up with these demands in order to satisfy the needs of their users, providing low-latency network access. Upgrading base stations and core network routers to higher-capacity equipment reduces high utilization in the core network, but comes with significant operational cost. Furthermore, network operators are impelled to quickly integrate upcoming technologies (like Long Term Evolution, LTE) at the edge of their networks, offering better quality of experience. Higher bandwidth capacities at the network edge, however, directly affect core network utilization and require further investments.

Two technologies have recently been proposed as a remedy to this dilemma: 1) Mobile edge computing and 2) network virtualization. Mobile edge computing aims at reducing latency by shifting computational efforts from the Internet cloud to the mobile edge. Network virtualization increases management flexibility of the mobile infrastructure and enables resource sharing between multiple providers.

1) *Mobile edge computing (MEC)* is an emerging technology that is seen as an alleviating factor in this context [3], [4], [5]. MEC aims at reducing both network latency and resource demands by shifting computing and storage capacities from the Internet cloud to the mobile edge. Instead of uploading or downloading content generated or demanded by the mobile user, mobile applications refer to a service located close to the current position of the user. Services are hosted on devices directly attached to base stations or smart cells (i.e., macrocells, microcells, or picocells). These hosts are also known as MEC servers and are operated by the mobile infrastructure provider. The proximity of the MEC servers to the mobile device not only takes load from the mobile core but also increases responsiveness of mobile applications.

2) *Network virtualization* is commonly seen as a key technology for the Future Internet and has recently been proposed in the context of mobile networks [1], [6], [7], [8]. Network Virtualization enables sharing of physical network resources like base stations, core routers, and MEC servers between multiple network operators. Network virtualization enables operators to fully isolate their (virtual) resources from those hosted by others on the same physical device (data and control plane). Network sharing not only reduces cost for deployment of new hardware resources but also operational cost [1][6][8]. Increasing network management flexibilities, virtualization of network resources is also seen as a key technology to mitigate the ossification of the core protocols.

This paper proposes a fully virtualized MEC infrastructure. Virtualizing both the mobile core and the mobile edge network enables infrastructure providers to shared resources between several mobile operators (Infrastructure as a service, IaaS), including computing and storage capacities of the MEC servers servers, and enhances management flexibilities. One major challenge of network virtualization is the embedding of virtualized resources onto the physical network. This problem is known as the VNE problem. VNE algorithms aim to embed multiple Virtual Network Requests (VNRs) onto a shared substrate network, enabling virtual network operators to share a common substrate infrastructure flawlessly by assigning sufficient resources. While virtualization is a technique that is well-known both in mobile networks and computer networks, the VNE problem has not been discussed in the context of MEC so far. Therefore, this paper addresses this shortcoming and identifies new research directions for future, MEC-specific VNE approaches.

The remainder of this paper is structured as follows: In section II, network topologies of MEC networks are explained. Furthermore, the VNE problem is formalized and the formalization is extended with respect to the mobile edge scenario. Section III-A introduces new VNE parameters and Section III-B introduces VNE optimization objectives for MEC networks. In section III-C, challenges for future VNE approaches are discussed. Section IV discusses related work and section V concludes the paper.

II. Virtualization of Mobile Edge Infrastructures

This section discusses mobile edge computing, motivates network virtualization in the context of mobile networks, and

ISBN: 978-1-61208-354-4

formulates the VNE problem in the context of mobile edge networks.

A. Mobile Edge Computing

Mobile edge computing is an emerging concept becoming more and more feasible with the shift towards the LTE wireless communication standard. LTE's new core network, *System Architecture Evolution (SAE)*, is an all-IP network with a simplified architecture, allowing for greater flexibility of the network's topology and more heterogeneous access networks, integrating legacy systems (e.g., air interfaces of GPRS or UMTS) and LTE's new *Evolved Universal Terrestrial Radio Access (E-UTRA)*. Due to LTE's low-latency and high-bandwidth radio access networks, deployment of new computing resources at the mobile edge becomes a promising approach for supporting novel latency-sensitive applications.

MEC servers provide computing, storage and bandwidth capacity that is shared by multiple virtual machines installed on top of them. Fig. 1 depicts the mobile edge computing scenario. MEC servers, being owned and managed by the infrastructure provider, are directly attached to the base stations. Traditionally, all data traffic originating at the data centers is forwarded by Internet routers to the mobile core network. The traffic is routed through the core network to a base station which delivers the content to the mobile devices. In the mobile edge computing scenario, MEC servers take over some or even all of the tasks originally performed in a data centers. Being located at the mobile edge, this eliminated the need of routing these data through the core network, leading to low communication latency.

Two different, but related usage scenarios have been proposed in the context of mobile edge computing: The first one proposes mobile devices to delegate calculations to the MEC servers (*offloading* of resource- or power-intense tasks) [5], while the second one proposes application service providers (ASPs) to deploy services traditionally hosted within data centers on the MEC servers (*Edge Deployment*) [3], [4]. Both aim at making the edge of the mobile network smarter, leading to a reduction of core network utilization and decreased latency:

1) Offloading Some applications running on a mobile device are capable of offloading resource- or power-intense tasks to MEC servers. Therefore, the mobile application invokes additional services deployed at a virtual machine hosted on a MEC servers. MEC servers are placed nearby, offer excellent Internet connectivity, and are easily reachable by the mobile device: In the best case, there is a one-hop communication between mobile device and host, offering low-latency access. The concept is expected to increase limited computing, storage, or bandwidth capacities of mobile devices by referring to external, resource-rich resources. Another objective of offloading is to reduce power consumption of a mobile device.

If no MEC server is available, the mobile device degrades gracefully to a more distant MEC server, a remote Internet cloud server, or use its own hardware resources [5], [9].

2) Edge Deployment In the edge deployment scenario, network providers offer MEC capacities for the deployment of ASP-operated virtual machines running at the edge. ASPs offer additional services at the network edge, increasing responsiveness of their applications. Services or parts of services traditionally hosted in data centers are now shifted to the network edge. Since traffic between MEC servers and mobile devices has not to be routed through the core network, this leads to decreasing core network utilization and lower communication latency.

B. Network Virtualization

Network virtualization has been proposed both in the context of computer networks and for mobile core networks [3], [4], [5]. This paper proposes the application of network virtualization techniques for the whole network infrastructure, including network core, base stations, and MEC servers. Network virtualization is proposed as a key technology to overcome the ossification of core protocols, since it enables the deployment of several, isolated virtual networks on top of a shared physical infrastructure. Virtual networks are co-hosted on a common substrate infrastructure and, since they are fully isolated, are even capable of deploying different communication protocols (e.g., IPv4/v6 or proprietary protocols) on the same substrate links.

Infrastructure providers (InPs) offer physical network resources to several mobile network operators. Operators specify network topologies and hardware resource demands to be deployed within the infrastructure of the InP. Operators are usually external customers of the InP. This does not exclude, however, that InP itself can also deploy networks on its own on top of its infrastructure, renting spare resources to other operators. The InP provides its physical resources to the operators, ensuring that all network requests of the operators are fulfilled. In this paper, fully-virtualized MEC networks are proposed. This means that both the network core, and also the network edge, i.e., MEC servers and base stations resources provide virtualization capabilities. Network virtualization is a useful technique in order to separate several internal networks and to increase manageability. Furthermore, it enables the InP to rent spare resources to other operators.

One important aspect in this area of research lies in the embedding of virtual network entities to the physical (or to be more general: the substrate) infrastructure. This is commonly known as the Virtual Network Embedding (VNE) problem [10]. Physical resources are limited and have to be shared between the virtual network entities that are assigned to these resources. This is depicted in Fig. 2: two network requests are assigned to a substrate network. The VNE problem is divided in two sub-problems: *Virtual Node Mapping* and *Virtual Link Mapping*: Virtual nodes are assigned to substrate nodes offering sufficient resources. Virtual links are either assigned to a single substrate link, or span a path of multiple links in the substrate network, where each link offers sufficient resources. This is shown in Fig. 2 for the virtual link demanding 100MBit/s bandwidth capacity. The VNE problem becomes $\mathcal{NP}$-hard when substrate nodes and links have finite resources [10].

C. Problem Formulation

In this subsection, a formal description for the general VNE problem as depicted in Fig. 2 is presented. This formal model will then be enhanced with respect to MEC specific properties.

A substrate network $S = (N, L)$ is modeled as a set of substrate nodes N and a set of links L mutually connecting some of the nodes. Similar to the substrate network, a VNR is modeled as a collection of virtual nodes N^i and links L^i. Substrate nodes and links offer resources R, assigned by $\text{cap} : N \cup L \rightarrow 2^R$. Virtual nodes and links *demand* these resources, formally described as $\text{dem}_i : N^i \cup L^i \rightarrow 2^R$. The objective of a VNE algorithm is to embed several Virtual Network Requests VNRs, denoting $\text{VNR}^i = (N^i, L^i)$ as being the i-th request. Virtual entities that are embedded onto a substrate entity *consume* substrate resources they demanded. Therefore, the VNE has to assure that a sufficient amount of resources is provided by a substrate entity before a virtual entity gets assigned to it. The embedding is modeled as a

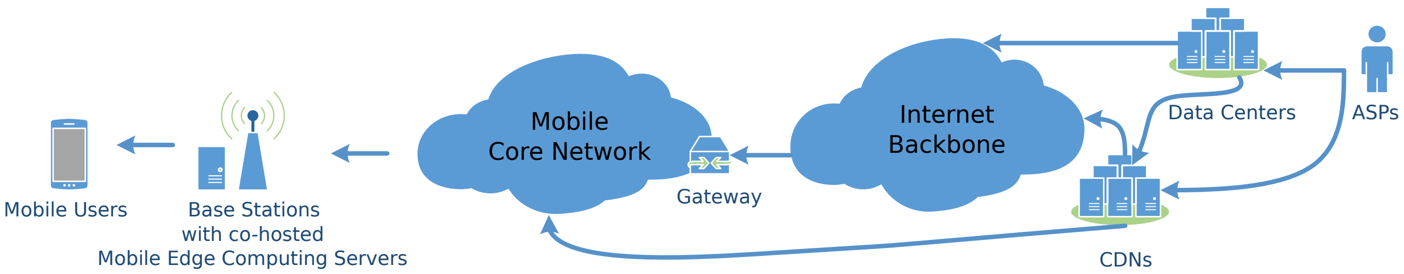


Fig. 1: Mobile Edge Computing: Deployment of MEC servers at the Edge of the Mobile Network

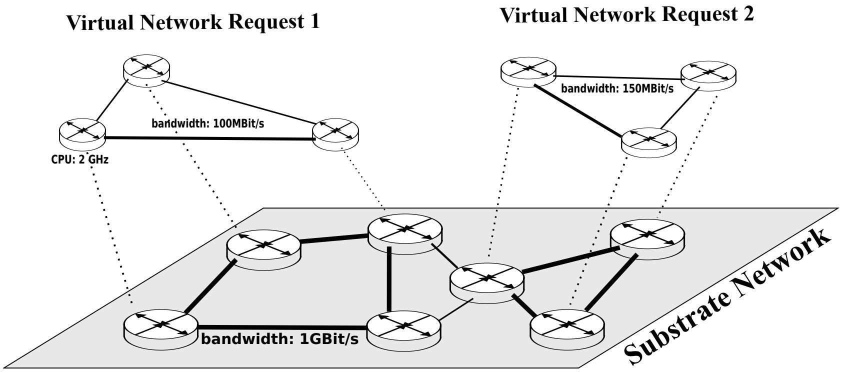


Fig. 2: Virtual Network Embedding

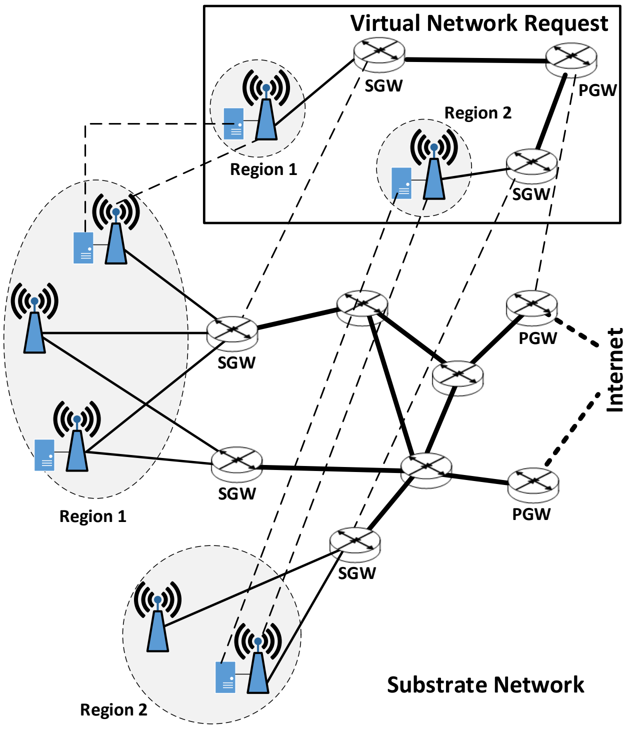


Fig. 3: Embedding of Mobile Edge Computing Networks

function $f_i : N^i \rightarrow N$ assigning nodes of VNR^i to the substrate network, and a function $g_i : L^i \rightarrow \mathrm{SN}' \subseteq SN$ assigning edges of the VNR to the substrate links (or a combination of links, i.e., paths). The notation used here is in line with the one introduced in [10].

The VNE problem in the mobile edge scenario is depicted in Fig. 3: For this scenario, a differentiation between edge and core nodes is needed. Therefore, the following new variables are introduced: N_{edge} and N_{core} for substrate edge/core nodes, L_{edge} and L_{core} for substrate edge/core links, and, respectively, N^i_{edge}, N^i_{core}, L^i_{edge}, and L^i_{core} for the virtual nodes and edges.

Edge nodes are nodes located at the edge of the network, i.e., nodes providing network access to the mobile devices (representing base stations) or computing and storage capabilities (representing MEC servers). Core nodes are all other nodes (network core devices), e.g., routers, *Serving Gateways (SGW)* (connecting the base stations to the core network) or *Packet Data Network Gateways (PGW)* (connecting the core network to the internet). A main characteristic of edge nodes representing base stations is their relatedness to a specific geographic region. Fig. 3 depicts the embedding of a VNR. In this case, the VNR demands base stations with affiliated MEC servers in two specific regions (region 1 and 2), thus not allowing to, e.g., only use the two base stations with MEC capabilities in region 1. Geographic constraints of edge nodes are one of the new challenges for VNE in the MEC scenario, and are reflected in the proposed VNE parameters and optimization objectives which are discussed in the next section.

III. Embedding Virtual Networks in Mobile Edge Computing Scenarios

This section introduces the VNE problem in the context of mobile edge computing. To this end, the VNE problem (1) is extended with respect to edge-specific parameters and (2) new optimization objectives are introduced. Furthermore, (3) general challenges for future VNE algorithms in mobile edge networks are discussed. To the best of our knowledge, these parameters, optimization objectives, and mobile edge related challenges have not been discussed in literature before.

A. New VNE parameters

VNE algorithms take different parameters of network resources into account for calculating network embeddings. As an example for such parameters, substrate resources have individual capacities, and virtual resources have respective requirements (also known as *demands*). Various kinds of resources are assigned to nodes and links: For example, in traditional VNE scenarios, CPU resources are assigned to nodes and bandwidth resources to links. Such a strict distinction in node and link parameters, however, is problematic, since some parameters influence each other. For example, CPU utilization of a node being part of a path between two nodes influences available bandwidth on that communication path.

Fischer et al. propose a classification of VNE parameters: *Primary/secondary* and *functional/non-functional* [10]. Primary parameters like CPU resources are directly assigned to a substrate resource. Secondary parameters, however, depend on other primary parameters and these side-effects have to be considered first. As an example, packet loss at a node depends on the primary parameters CPU resource and memory resource. Functional parameters like CPU-/bandwidth capacity specify low-level functionality. Non-functional parameters are high-level properties. For example, resilience and security are both non-functional parameters.

Being two well-known parameters common to most existing VNE approaches, CPU and bandwidth also apply in the context of MEC. Both core and edge nodes provide CPU capacities. On core nodes, CPU capacity is consumed for tasks like routing and mobility-specific tasks like authorization and billing. Edge nodes, i.e., MEC servers, provide CPU capacity for applications deployed by the ASPs. In the following, new VNE parameters are introduced that are applicative to the MEC scenario. Parameters are classified as being primary/secondary and functional/non-functional.

MEC Coverage (primary & functional) Base stations provide wireless link capacity, connecting mobile devices to the mobile network. The range of a typical macrocell base station covers several dozens of kilometers. Other types of base stations, small cells like microcells, cover much lower ranges. Relay nodes are base stations providing enhanced coverage at cell edges and hot-spot regions, covering the same region as the main base station.
Coverage refers to the geographical region covered by the base stations of the same mobile operator, i.e., the geographical region in which mobile devices are able to communicate with the operator's core network. Coverage depends on lower level characteristics such as physical location of the base station, transmission power and environmental influences. Increasing the transmission power of a base station increases the radius of the covered region, but also decreases average bandwidth available to the mobile users. Increasing coverage leads to interferences with other base stations if frequencies are reused. In the MEC scenario, coverage also influences availability of resources provided by MEC servers due to the fact that MEC servers are directly connected to base stations. Thus, increasing coverage also increases availability of MEC resources. However, since more mobile users are able to reach the MEC server, this leads to higher utilization of the MEC server and, thus, to higher latency. Therefore, there is a tradeoff between availability and utilization/latency of MEC resources.

MEC Server Storage (primary & functional) MEC servers provide disk storage for virtual machines running on top of them in order to, e.g., cache proximity-related data. This requirement is not considered by most existing VNE approaches.

Latency (secondary & functional) If the VNE algorithm embeds a latency-sensitive link to a path spanning several substrate links, the sum of all latency properties of these links may never exceed the demanded maximum latency of the corresponding virtual link. Latency is a critical factor for several mobile applications. Therefore, operators define an upper bound for reaching a MEC server from a certain base station. Depending on this value, the VNE is either able to choose between any nearby MEC server or is forced to use substrate resources of a MEC server right at the given base station. Communication latency is a secondary, functional parameter, since it depends on CPU utilization and refers to low-level functionality.

Regional Bandwidth Capacity (secondary & functional) Cell capacity refers to available bandwidth resources provided by one or more base stations within a region (up- and downlink). Bandwidth capacity is increased by operating multiple *geographically adjacent* base stations, each covering a subset of the same region, instead of just one macrocell. Furthermore, LTE Advanced allows the parallel deployment of small cells within the *same* region that is covered by a macrocell (enhanced Inter-cell interference coordination, eICIC), leading to improved bandwidth capacity. In the MEC scenario, regional bandwidth capacity is proportional to the amount of available MEC server resources that are accessible within a geographical region. Bandwidth capacity, being a secondary parameter, is limited by bandwidth resources of deployed base stations and by bandwidth resources of the links connecting the base stations to the core network.

Regional MEC Computing and Storage Capacity (secondary & functional) Similar to regional bandwidth capacity, CPU/storage capacity refers to CPU/storage resources available in a region. CPU/storage capacity compounds of the resources of MEC servers directly connected to the base stations deployed in that region and computing capacity of *logically adjacent* MEC servers, i.e., MEC servers that are either directly connected to base stations covering that region or well-connected through the mobile network to the base stations.

MEC Resources per Mobile Device (secondary & functional) Since all mobile devices in a single cell share resources corresponding to the above parameters CPU capacity, storage, bandwidth and latency, VNRs demand resources on a per-user basis, e.g., a certain minimum amount of bandwidth per user in order to achieve the intended quality of experience. As a consequence, the VNE has to lower the average number of users in certain cells, which is done by adjusting the coverage of adjacent cells (i.e., to distribute the users among multiple smaller cells). Vice versa, when resource requirements are smaller than available resources, the VNE is able to utilize a smaller number of base stations, omitting certain nodes completely, allowing to (temporarily) shut down these nodes for energy-saving purposes.

The next subsection discusses MEC-specific optimization objectives for the VNE problem. Due to the fact that many MEC-related parameters are secondary, i.e., they depend on other, primary parameters –, the interdependencies are emphasized.

B. New Optimization Objectives

VNE algorithms compute the *optimal* (or near-optimal) embedding of a set of VNRs. An embedding is optimized with respect to an optimization objective. As an example, such an optimization criterion is, in order to reduce cost for the IsP, to minimize the number of substrate resources that are utilized by the embedded virtual networks. This section discusses various VNE optimization objectives in the context of MEC.

Increase MEC Coverage One main objective for network operators is to provide cell coverage to their mobile users. Operators both aim at expanding geographical dimension of cell coverage, and the capacity of their cells: The more bandwidth resources are available in a region, the more users are able to connect to their networks (and pay for using their

services). Therefore, one objective for future VNE algorithms is to share resources provided by the base stations optimally between VNRs with respect to coverage and bandwidth. However, this is not limited to base stations: In MEC, this also applies to resources provided by MEC servers. Cells providing bandwidth resources also provide computing capabilites. These capabilities have to be adequately shared between network operators covering the same regions.
There is a tradeoff between power consumption and coverage, since more resources are needed to cover larger areas or to provide higher bandwidth capacities.

Reduce Latency Reducing communication latency is of major importance for today's mobile network operators. VNE algorithms are obliged to take this aspect into account by embedding VNRs in a way that minimizes latency at the edge and/or in the core network. Reducing latency at the edge, i.e., between MEC servers and base stations, leads to small response times between services hosted at the MEC servers and mobile users. To reduce overall network delay, delay between base stations and gateway nodes connecting the network core to the Internet (PGWs, cf. Fig. 3) has to be considered by future VNE approaches.

Provide QoS-compliant Embeddings Besides latency, future VNE algorithms have to consider flexible QoS- and QoE-considerations. VNRs either demand strict resource assignments which have to be reserved exclusively. As an example, a VNR requests 1km cell capacity in a specific area and 10GB of MEC server storage capacity. These resources are then exclusively reserved for the VNR and thus can not be shared with other VNRs. As an alternative, VNE algorithms should balance utilization throughout the substrate network resources in order to guarantee equivalent QoS for all VNRs. In this case, VNRs do not request such strict assignments and the VNE aims at providing well-balanced cell coverage and MEC server capacities for all VNRs.

Maximize Regional MEC Capacities Regional bandwidth/ computing/ storage capacity refers to available resources provided by one or more base stations or MEC servers within a region. Bandwidth is limited by resources of the base stations, but also by the links connecting the base station to the core network. Bandwidth capacity is increased by overlapping coverage of *geographically adjacent* base stations. Computing capacity is increased by utilizing *logically adjacent* MEC servers, i.e., MEC servers that are well-connected through the core network to a base station covering the region.

Provide Resilience / Fault Tolerance for MEC services Resilience at the network edge is provided by a VNE by allocating resources of additional, geographically adjacent base stations covering the same region. If another base station fails, this backup resource takes over. The VNE has to ensure that the backup resource does not introduce any interferences with other base stations. Backup MEC server capabilities are provided by allocating logically adjacent MEC server resources. The path from the base station to the backup MEC server should provide similar latency conditions as the old one.

Reduction of Power Consumption Only few VNE approaches aim at reducing power consumption of substrate resources [10]. However, since they focus on data center routers and servers, these approaches are only partly applicable in the context of MEC. In the MEC scenario, new power models and interdependencies have to be considered: Increasing transmission power of base stations extends coverage, but also influence other base stations [11]. Decreasing power consumption decreases coverage, impelling mobile devices to re-register to other base stations due to poor connection quality to the current BS. This leads to load shifting both at the edge of the mobile network and, as a consequence, also in the network core. Current energy-aware VNE approaches aim at switching off as many substrate entities as possible in order to decrease overall power consumption. However, in the MEC scenario, this not only results in decreased coverage, but also in reduction of MEC server computing and storage capacity. If the connection of the base station to the core network is shut down, the MEC server gets isolated. MEC servers connected to base stations that are already in use should be preferred instead of switching on MEC servers connected to inactive core resources.
Therefore, there is a tradeoff that should be taken into account by future VNE algorithms: Coverage of base stations, available bandwidth, and MEC server capacity in this region vs. energy efficiency. How sparse can network infrastructure be / how many cloudlets can be switched off, while still ensuring enough bandwidth and CPU capacities?

Provide Security at the Edge Due to security considerations, not all operators or ASPs want to share MEC servers running their applications with other operators. Therefore, virtual MEC servers of different providers are embedded onto different substrate MEC servers in order to reduce the risk of any malicious influence (e.g., denial of service attacks). The challenge for new VNE approaches is to find an embedding that considers security constraints by still ensuring latency/QoS demands.

C. Challenges in Mobile Edge Networks

This subsection outlines several general challenges for VNE algorithms in the context of large-scale, fully-virtualized mobile networks.

Scalability As mentioned before, the VNE problem is $\mathcal{NP}$-hard. Therefore, most VNE approaches are based on heuristics. This results in non-optimal solutions, but decreases the size of the problem and leads to significant improvements with respect to runtime. However, almost all algorithms rely on a single, central node calculating the VNE. Centralization hinders scalability for large-scale, dynamic networks. In fact, most algorithms were evaluated with substrate networks spanning only few dozens or hundreds of nodes. Such settings might be realistic in the context of middle sized networks like testbeds. However, it is far away from national-wide or transnational mobile networks. One might be puzzled by the fact, therefore, why only few approaches are distributed: In fact, almost all VNE algorithms are centralized and require full knowledge of the substrate network topology and substrate resources [10], [12]. In large-scale environments like real-life mobile networks, current VNE algorithms are stretched to their limits [13].

Mobility-Awareness of Embeddings Another VNE challenge in MEC networks is to provide mobility-awareness. Mobile users move between neighbored cells, resulting in additional handoff overhead and routing of traffic through different paths. Therefore, embedding communication paths of virtual base stations covering adjacent cells on similar core network paths reduces variations in latency (jitter). A mobility-aware VNE algorithm aims at providing both physical and logical proximity for virtual base stations covering adjacent regions.

Utilization-Awareness of Embeddings In mobile networks, both edge and core utilization fluctuates significantly depending on the time of day. Demand also varies due to big sports events, new year's Eve, etc. Davy et al. propose the usage of user mobility models for shared network resources, predicting aggregated movement patterns of mobile users. Furthermore, popularity prediction methods for video content discussed, predict distribution of video content based on their expected popularity [13]. Integrating such prediction methods is seen as promising step towards utilization-awareness VNE algorithms. Being static and centralized, most VNE algorithms cannot cope with dynamic and flexible VNR requirements in large-scale environments. Therefore, novel VNE algorithms should be both dynamic and distributed in order to provide flexibility and adaptability in large network scenarios.

IV. Related Work

This section discusses related work: Several papers related to mobile network virtualization are mentioned. To the best of our knowledge, there is neither related work on network virtualization in the context of edge computing networks nor on VNE algorithms in this context.

Virtualization-based isolation techniques in the context of mobile networks are proposed as an enabling technology for future, cost-efficient mobile networks, shared and operated by multiple network operators. Different network infrastructure sharing scenarios are discussed and sharing options are classified based on different business models [1].

A general virtualization-enabled network architecture is proposed in [6]. The advantages of shared, hereogeneous network infrastructures are emphasized and applications of network virtualization in this context is discussed. An approach towards the virtualization of LTE networks has been introduced by [8]. Authors discuss the advantages of virtualizing LTE infrastructures and elaborate on virtualization of the LTE air interface.

An extensive and up-to-date classification of current VNE parameters and objectives is given in [10]. Many VNE approaches have been proposed so far, focussing on embedding objectives like cost-optimization, resilience etc. While some objectives are related to the ones presented in this paper (e.g., resilience and security), none considers the special demands of MEC networks. Since InPs and network operators usually aim for a combination of multiple (possibly contrary) objectives (e.g., reducing power consumption to a certain extend by also providing a sufficient degree of network resilience), an in-depth evaluation of these approaches in combination with the novel VNE parameters and objectives presented in this paper is left for future work.

V. Conclusion

This paper proposes network virtualization as a key technology for future mobile edge computing networks. The Virtual Network Embedding problem is analyzed in this context and new challenges for future embedding algorithms are discussed. To the best of our knowledge, no VNE approach has been published so far considering the MEC-specific network parameters and optimization objectives presented here. Therefore, the authors hope that this position paper provides an initial step towards new VNE approaches. As a first step, the authors are currently implementing a delay-aware VNE algorithm that considers geographical constraints as well as latency considerations.

References

[1] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki, “Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence,” Communications Magazine, IEEE, vol. 49, no. 10, pp. 134–142, 2011.

[2] Ericcson, “Ericcson mobility report june 2013,” http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-june-2013.pdf [retrieved: 2014-01-28].

[3] Intel and Nokia Siemens Networks, “Increasing mobile operators’ value proposition with edge computing,” http://nsn.com/system/files/document/edgecomputingtechbrief_328909_002_0.pdf [retrieved: 2014-01-28].

[4] IBM Corporation, “Smarter wireless networks; add intelligence to the mobile network edge,” http://public.dhe.ibm.com/common/ssi/ecm/en/wsw14201usen/WSW14201USEN.PDF [retrieved: 2014-01-28].

[5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, 2009.

[6] M. Hoffmann and M. Staufer, “Network virtualization for future mobile networks: General architecture and applications,” in Communications Workshops (ICC), 2011 IEEE International Conference on. IEEE, 2011, p. 1–5.

[7] E. Kudoh and F. Adachi, “Power and frequency efficient virtual cellular network,” in Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, vol. 4. IEEE, 2003, pp. 2485–2489. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1208838

[8] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, “LTE mobile network virtualization: Exploiting multiplexing and multi-user diversity gain,” Mobile Networks and Applications, vol. 16, no. 4, pp. 424–432, 2011.

[9] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, “Impact of cloudlets on interactive mobile cloud applications,” 16th International Enterprise Distributed Object Computing Conference, pp. 123–132, Sep. 2012.

[10] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, “Virtual network embedding: A survey,” Communications Surveys & Tutorials, IEEE, vol. 15, no. 4, pp. 1888–1906, 2013.

[11] F. Richter, A. J. Fehske, and G. P. Fettweis, “Energy efficiency aspects of base station deployment strategies for cellular networks,” in Vehicular Technology Conference Fall. IEEE, 2009, p. 1–5.

[12] M. T. Beck, J. F. Botero, A. Fischer, H. De Meer, and X. Hesselbach, “A distributed, parallel, and generic virtual network embedding framework,” in IEEE Int’l Conf. on Communications (ICC). IEEE, 2013. [Online]. Available: http://www.net.fim.uni-passau.de/pdf/Beck2013a.pdf

[13] S. Davy, J. Famaey, J. Serrat, J. L. Gorricho, A. Miron, M. Dramitinos, P. M. Neves, S. Latre, and E. Goshen, “Challenges to support edge-as-a-service,” Communications Magazine, IEEE, vol. 52, no. 1, pp. 132–139, 2014.

The Significance of Imaginary Points in Linear Least Square Approximation

Mikael Fridenfalk
Department of Game Design
Uppsala University
Visby, Sweden
mikael.fridenfalk@speldesign.uu.se

***Abstract*—The linear least square method constitutes one of the most useful statistical methods in mathematics. This paper shows that in this method, real, versus imaginary points may act as minimizers, versus maximizers of the error. The use of imaginary points provides thereby an additional degree of freedom in the design of methods based on this statistical method.**

Keywords-complex number; curve fitting; imaginary number; least square method

I. Introduction

Previous research on complex numbers in conjunction with the linear least square method has been restricted to applications, such as constrained phases [2], stochastic processes [5], and complex monomial neural networks [1]. In this paper, a new application is presented, using real, versus imaginary points as error minimizers, versus maximizers. The initial motivation for the development of the method presented in this paper was to accelerate the backpropagation process in the evaluation of the weights of a large-scale feedforward neural network. An analytic solution was however found along the way with the potential to replace backpropagation in large-scale neural networks [4]. As a brief overview, in Section II, the theory behind the linear least square is reiterated, along with a curve-fitting example. In Section III, a new and more generalized method is proposed for linear least square fitting, including imaginary points, followed by experimental results in Section IV for the verification of the proposed method.

II. State of the Art

To begin with, the theory behind the standard method is reiterated. To reproduce a textbook example on the subject [3], given the inconsistent linear equation system:

$$x_1 + x_2 = 4 \tag{1}$$

$$2x_1 + x_2 = 8 \tag{2}$$

$$x_1 + 2x_2 = 5 \tag{3}$$

or alternatively expressed, using a matrix $\mathbf{A}$ of size $M \times N$ (with $M = 3$ and $N = 2$), and the vectors $\mathbf{x}$ and $\mathbf{b}$:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 2 \end{bmatrix} = [\ \mathbf{c}_1 \quad \mathbf{c}_2\ ] = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} \tag{5}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6}$$

$$\mathbf{b} = \begin{bmatrix} 4 \\ 8 \\ 5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{7}$$

where $\mathbf{c}_1$ and $\mathbf{c}_2$ are the column vectors of $\mathbf{A}$, and $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ the row vectors of $\mathbf{A}$. It is possible to schematically represent such system by Fig. 1, with:

$$\mathbf{b} = \mathbf{p} + \mathbf{q} \tag{8}$$

where $\mathbf{b}$ denotes in this example a vector outside a plane V, spanned by $\mathbf{c}_1$ and $\mathbf{c}_2$, and $\mathbf{p}$ denotes the orthogonal projection of $\mathbf{b}$ on V, as shown in Fig. 1. Due to orthogonal projection, the smallest distance from $\mathbf{b}$ to V is $\epsilon = |\mathbf{p} - \mathbf{b}|$, and therefore $\epsilon^2 = |\mathbf{p} - \mathbf{b}|^2$ is minimal. The least square error may in this example be expressed as:

$$\epsilon^2 = \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 \tag{9}$$

$$|\mathbf{p} - \mathbf{b}|^2 = (\mathbf{r}_1 \cdot \mathbf{x} - b_1)^2 + (\mathbf{r}_2 \cdot \mathbf{x} - b_2)^2 + (\mathbf{r}_3 \cdot \mathbf{x} - b_3)^2 \tag{10}$$

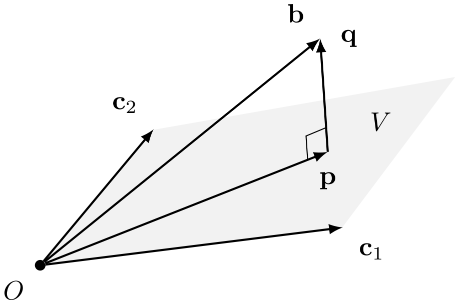


Figure 1. Orthogonal projection of $\mathbf{b}$ on plane V, spanned by $\mathbf{c}_1$ and $\mathbf{c}_2$.

or in the general case, given row m in $\mathbf{A}$ and element m in $\mathbf{b}$, with $\mathbf{p} = \mathbf{A}\mathbf{x}$, as:

$$\epsilon_m = |\mathbf{r}_m \cdot \mathbf{x} - b_m| \tag{11}$$

Given an arbitrary vector $\mathbf{y}$ in $\mathbb{R}^N$:

$$\mathbf{A}\mathbf{y} = y_1\mathbf{c}_1 + y_2\mathbf{c}_2 + \ldots + y_N\mathbf{c}_N \tag{12}$$

Since $\mathbf{A}\mathbf{y}$ will always lie in V, and is, therefore, orthogonal to $\mathbf{q}$, the linear least square method can be derived by the following equations:

$$(\mathbf{A}\mathbf{y}) \cdot \mathbf{q} = (\mathbf{A}\mathbf{y})^T\mathbf{q} = 0 \tag{13}$$

$$(\mathbf{A}\mathbf{y})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0 \tag{14}$$

$$\mathbf{y}^T\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0 \tag{15}$$

$$\mathbf{y}^T(\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b}) = 0 \tag{16}$$

Thus:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b} \tag{17}$$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{18}$$

The estimated solution $\mathbf{x}$, is for maximum evaluation speed, preferably solved by the direct solution of (17). As an example of a curve fitting application, given M points (a_m, b_m):

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix} \tag{19}$$

and a second degree polynomial:

$$b = x_1 + x_2 a + x_3 a^2 \tag{20}$$

$$\mathbf{A} = \begin{bmatrix} 1 & a_1 & a_1^2 \\ 1 & a_2 & a_2^2 \\ \vdots & \vdots & \vdots \\ 1 & a_N & a_N^2 \end{bmatrix} \tag{21}$$

Equations (19) and (21) with (17) give thus the least square solution $\mathbf{x}$ for the polynomial in (20), that best approximates the distribution of the sample points in (19).

III. Proposal

As an example of the extension of the standard method described above for least square curve fitting, we present here a similar method, but with the addition of imaginary points. The definition of an imaginary point is in this paper any row m in $\mathbf{A}$ and element m in $\mathbf{b}$, that has been multiplied with i, as in $\sqrt{-1}$. In the following example, $m = 2$ is designated as an imaginary point:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \cdot i \\ \vdots \\ a_M \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \cdot i \\ \vdots \\ b_M \end{bmatrix} \tag{22}$$

Given the polynomial equation in (20):

$$\mathbf{A} = \begin{bmatrix} 1 & a_1 & a_1^2 \\ i & a_2 \cdot i & a_2^2 \cdot i \\ \vdots & \vdots & \vdots \\ 1 & a_N & a_N^2 \end{bmatrix} \tag{23}$$

and the definitions of the square matrix $\mathbf{U} = \mathbf{A}^T\mathbf{A}$ and the vector $\mathbf{v} = \mathbf{A}^T\mathbf{b}$:

$$\mathbf{U} = \begin{bmatrix} 1 & i & \dots & 1 \\ a_1 & a_2 \cdot i & \dots & a_N \\ a_1^2 & a_2^2 \cdot i & \dots & a_N^2 \end{bmatrix} \begin{bmatrix} 1 & a_1 & a_1^2 \\ i & a_2 \cdot i & a_2^2 \cdot i \\ \vdots & \vdots & \vdots \\ 1 & a_N & a_N^2 \end{bmatrix} \tag{24}$$

$$\mathbf{v} = \begin{bmatrix} 1 & i & \dots & 1 \\ a_1 & a_2 \cdot i & \dots & a_N \\ a_1^2 & a_2^2 \cdot i & \dots & a_N^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \cdot i \\ \vdots \\ b_N \end{bmatrix} \tag{25}$$

Since $i^2 = -1$, both $\mathbf{U}$ and $\mathbf{v}$ will only consist of real numbers, yielding the least square equation:

$$\mathbf{U}\mathbf{x} = \mathbf{v} \tag{26}$$

Given the standard least square method, based on M sample points, the squared error may for any sample m be expressed as:

$$\epsilon_m^2 = (\mathbf{r}_m \mathbf{x} - b_m)^2 \tag{27}$$

or more explicitly:

$$\epsilon_m^2 = \left(\begin{bmatrix} a_{m1} & a_{m2} & \dots & a_{mN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} - b_m \right)^2 \tag{28}$$

Therefore, the total squared error is equal to:

$$\epsilon^2 = \sum_{m=1}^{M} (\mathbf{r}_m \mathbf{x} - b_m)^2 \tag{29}$$

The division of the M sample points into J real points (a_j, b_j), versus K imaginary points (a_k, b_k), yields the following equations for the squared errors:

$$\epsilon_j^2 = \left(\begin{bmatrix} 1 & a_j & a_j^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - b_j \right)^2 \tag{30}$$

$$\epsilon_k^2 = \left(\begin{bmatrix} i & a_k \cdot i & a_k^2 \cdot i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - b_k \cdot i \right)^2 \tag{31}$$

By extraction of i and the relation $i^2 = -1$:

$$\epsilon_k^2 = (\mathbf{r}_k \mathbf{x} \cdot i - b_k \cdot i)^2 \tag{32}$$

$$\epsilon_k^2 = i^2 (\mathbf{r}_k \mathbf{x} - b_k)^2 \tag{33}$$

$$\epsilon_k^2 = -(\mathbf{r}_k \mathbf{x} - b_k)^2 \tag{34}$$

$$\epsilon^2 = \sum_{j=1}^{J} (\mathbf{r}_j \mathbf{x} - b_j)^2 - \sum_{k=1}^{K} (\mathbf{r}_k \mathbf{x} - b_k)^2 \tag{35}$$

Thus, imaginary points seem to reverse the direction or “polarity” of the least square method. However, since this method is based on projection, and the parameter that is minimized is $|\epsilon|$, large error components may reverse the expected polarities of sample points.

IV. Experimental Results

An equation solver was developed in C++ for the solution of linear equation systems of the form $\mathbf{U}\mathbf{x} = \mathbf{v}$. To optimize the evaluation speed of $\mathbf{U}$ and $\mathbf{v}$, a specialized matrix multiplication method was implemented, by the reversal of the signs of the products that correspond to imaginary points. Using this equation solver, Figs. 2-9 show the results of curve fitting of a third degree polynomial, based on real (•), versus imaginary points (×).

To comment these figures, Fig. 2 presents a curve fitting experiment using a third degree polynomial, with four real points, and an imaginary point, placed slightly above the line formed by the real points, yielding due to symmetry, a parabolic curve. As shown here, while the distance to the real points is minimized, the distance to the imaginary point is maximized. Figure 3 presents the same case as in previous figure, but with an imaginary point placed on the same line as the one formed by the real points, yielding as expected a flat line. Figure 4 presents the same case as in previous figure, but here with an imaginary point placed slightly below the line formed by the real points. Figure 5 presents the inversion of real versus imaginary points with respect to previous figure. Since the least square method minimizes $|\epsilon|$, even if the system is reversed (with $\epsilon^2 < 0$, where ϵ is an imaginary number), as expected by the derived theory, the result remains the same. Figure 6 presents a least square curve fitting of a third degree polynomial based on seven real points. Figures 7-8 present the same case as in Fig. 6, except for the replacement of a real point with an imaginary. Regarding Fig. 9, in our experiments, a borderline point such as this showed to nominally reverse the polarity of an imaginary point. A borderline point seems thus to be able to affect the error by a significant amount.

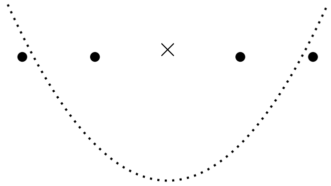


Figure 2. Curve fitting using a third degree polynomial, with four real points (●), and an imaginary point (×).

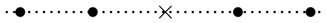


Figure 3. Same as previous figure, but with an imaginary point placed on the same line as the one formed by the real points.

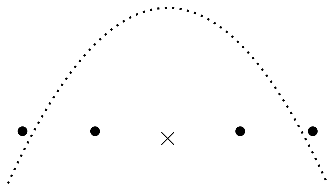


Figure 4. Same as previous figure, but here with an imaginary point placed slightly below the line formed by the real points.

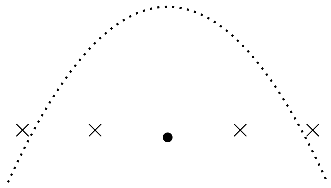


Figure 5. Inversion of real versus imaginary points.

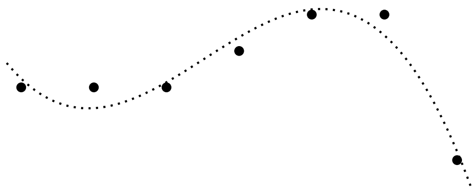


Figure 6. Least square curve fitting of a third degree polynomial based on seven real points.

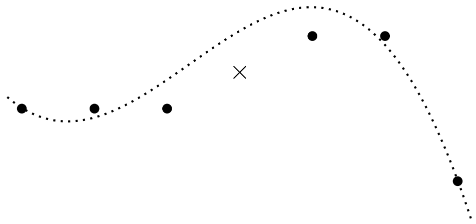


Figure 7. Same as previous figure, except for the replacement of a real point with an imaginary.

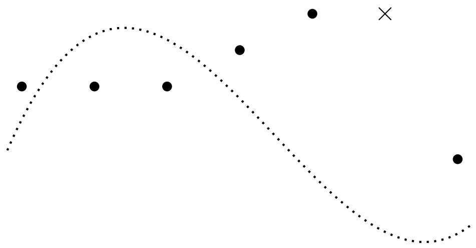


Figure 8. Same as previous figure.

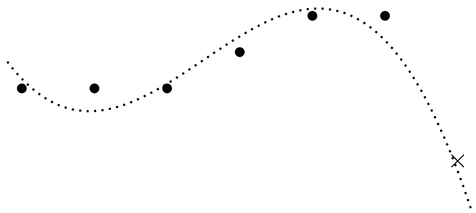


Figure 9. A demonstration of the effect of a borderline point.

V. Conclusion

The linear least square method is associated with the evaluation of coefficients of linear equation systems, minimizing errors. This paper shows that pure imaginary points (as defined in this paper) in a such system, tend nominally to act as squared error maximizers instead of minimizers. According to experimental results, the new method needs to be used with caution, since crossover between real and imaginary errors affects the behavior of the system.

References

[1] M. F. Amin, R. Savitha, M. I. Amin, and K. Murase, "Orthogonal Least Squares Based Complex-Valued Functional Link Network", Neural Networks, Elsevier, vol. 32, 2012, pp. 257-266.

[2] M. Bydder, "Solution of a Complex Least Squares Problem with Constrained Phase", Linear Algebra Appl., vol. 433, no. 10-11, 2010, pp. 1719-1721.

[3] C. H. Edwards and D. E. Penney, Elementary Linear Algebra, Prentice Hall, 1988.

[4] M. Fridenfalk, "The Development and Analysis of Analytic Method as Alternative for Backpropagation in Large-Scale Multilayer Neural Networks", accepted for publication in the proceedings of The Eighth International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP 2014, Rome, Italy, August, 2014.

[5] K. S. Miller, "Complex Linear Least Squares", SIAM Review, Society for Industrial and Applied Mathematics, vol. 15, no. 4, 1973, pp. 706-726.

Modeling and Visualization of Cataract Ontologies

- A prototypical application in ophthalmic hospitals -

Klaus Peter Scherer, Constantin Rieder, Christian Henninger, Markus Germann
Applied Computer Science
Karlsruhe Institute of Technology,
Eggenstein-Leopoldshafen, Germany
klaus-peter.scherer@kit.edu

Joachim Baumeister, Jochen Reutelshöfer
denkbares GmbH
Würzburg, Germany
joachim.baumeister@denkbares.com

***Abstract*—For handling the complex ophthalmic knowledge concerning surgical cataract interventions on human eyes knowledge domains are modeled as ontologies to guarantee a consistent knowledge at each moment. The different ophthalmic knowledge domains are developed with special wiki models (Semantic Wiki KnowWe). Based on the concepts and the relations between these concepts, the knowledge is represented by a semantic network. For user specific comfortable handling of the information, different visualization methods are designed, realized and compared. The visualization aims to satisfy the needs of ophthalmic experts (clinical surgeons) as well the knowledge engineers. So, different approaches of visualization are shown and evaluated by the clinical partners of the project.**

Keywords-knowledge base; computer aided assistance; knowledge representations; visualization; wiki representation; surgical intervention; ontology modeling ;semantic net.

I. Introduction

Concerning the complex surgical interventions on human eyes [1], it is very helpful for the clinical experts to get a support by a computer based knowledge system. Such a support system must guarantee a well-structured consistent information and a comfortable access to this information [2] [3]. The knowledge has to be formalized also for a logical reasoning process of the system. The final aim is to support the decision process of the ophthalmic surgeons by the visualized semantics. Different graphical visualization methods enhance the decision support of the doctors and enhance so the quality of the surgical process.

In Section II, a short medical background for the cataract surgical interventions is illustrated. The complex situation leads to a knowledge based approach, which is performed by the ontologies in Section III. Section IV shows the most important concepts and relations, realized for the cataract situation. In Section V, the same ontologies are represented by different visualizations, which the surgeon can choose and switch, dependent on his demand.

II. Medical background

In the biomechanical system of the human eye, the intraocular lens is the most important component for the refraction process to focus the rays to the retina. Parallel to the human aging process there is no possibility to prevent a fix dark cloudy lens with medical treatments. Cataract surgical interventions on human eyes are the single method to replace the old dark cloudy human lens by a new clear artificial lens [1]. This is performed about 700.000 times per year only in Germany. On the market, a lot of lens systems with different haptics, optics, materials and power refraction values complicate the selection of the most appropriate system (Fig. 1). A special configuration and selection of a patient related lens system can guaranteed by a computer-aided decision support system.

III. Concept of ontology support

For the ophthalmic surgical assistance different knowledge domains are used as ontologies, realized as a network of concepts and special relations between these concepts [4]. In order to build a semantic net and provide the users more expressive types of knowledge, domain-specific relation types between the so called concepts are created in the ontology network:

"subconcept": A refinement of the given concept, used to arrange the concepts in a hierarchical order.

"has to": Connection between complications, which may occur due the operation and their necessary treatments.

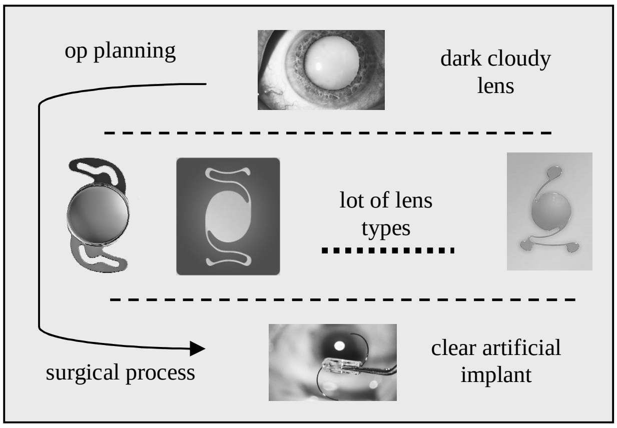


Figure 1. Planning a special lens implantation.

"can": A relation used to identify possible reactions to the given state of the patient.
"cave": This relation is used to connect concepts that should be urgently considered.
The resulting ontology is formalized in the RDF vocabulary description language.

IV. OPHTHALMIC ONTOLOGIES

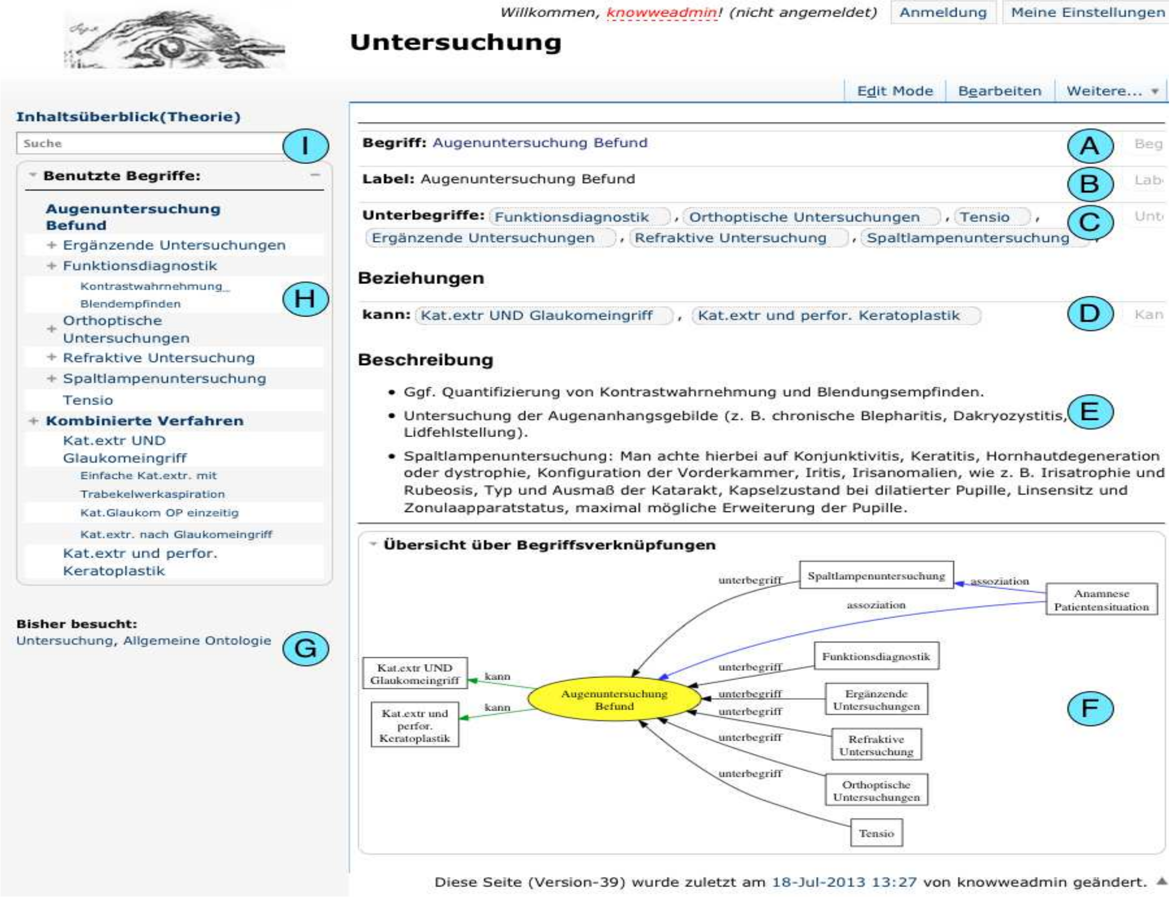
A. Knowledge concepts and relations

Fig. 2 shows a developed wiki-based knowledge concept, describing the domain "Augenuntersuchung Befund" (eye examination results) in its special structure. In an analogue way each concept is structured under the following two aspects. A custom concept definition markup defines a new concept of the ontology (A). A list of subconcepts defines the hierarchical structure of this given concept ("Unterkonzepte") (C). Furthermore, relations of the selected concept within the semantic network can be defined (D). The informal description of the concepts is described in standard wiki syntax (E) [6].

B. Realisation by special wikis

In the left panel of Fig. 2, a hierarchical collection of concepts is shown (H). It resembles a selection of the domain concepts from the ontology that recently were within the focus of the user, i.e., that have been used for editing or appeared on the visited documents. For the editing of the formalized parts of the content, i.e., the comma-separated lists of sub-concepts or other kinds of relations, the system enables drag-and-drop editing. Any concept within the left panel can be dragged onto a list of the document content and will be appended to it in the source text of the document. When a desired concept is currently not present in the left panel, it can be looked up using the search slot above it. The auto-completion functionality allows selecting the concept and adds it to the collection of concepts. In this way, the whole semantic network can easily changed by using drag-and-drop methods, while freedom and simplicity of document editing is retained [8].

.



Untersuchung

Inhaltsüberblick(Theorie)

Benutzte Begriffe:

Augenuntersuchung Befund
+ Ergänzende Untersuchungen
+ Funktionsdiagnostik
Kontrastwahrnehmung_
Blendempfinden
+ Orthoptische Untersuchungen
+ Refraktive Untersuchung
+ Spaltlampenuntersuchung
Tensio
+ Kombinierte Verfahren
Kat.extr UND Glaukomeingriff
Einfache Kat.extr. mit Trabekelwerkaspiration
Kat.Glaukom OP einzeitig
Kat.extr. nach Glaukomeingriff
Kat.extr und perfor. Keratoplastik

Bisher besucht:
Untersuchung, Allgemeine Ontologie

Begriff: Augenuntersuchung Befund

Label: Augenuntersuchung Befund

Unterbegriffe: Funktionsdiagnostik , Orthoptische Untersuchungen , Tensio , Ergänzende Untersuchungen , Refraktive Untersuchung , Spaltlampenuntersuchung ,

Beziehungen

kann: Kat.extr UND Glaukomeingriff , Kat.extr und perfor. Keratoplastik

Beschreibung

- Ggf. Quantifizierung von Kontrastwahrnehmung und Blendungsempfinden.
- Untersuchung der Augenanhangsgebilde (z. B. chronische Blepharitis, Dakryozystitis, Lidfehlstellung).
- Spaltlampenuntersuchung: Man achte hierbei auf Konjunktivitis, Keratitis, Hornhautdegeneration oder dystrophie, Konfiguration der Vorderkammer, Iritis, Irisanomalien, wie z. B. Irisatrophie und Rubeosis, Typ und Ausmaß der Katarakt, Kapselzustand bei dilatierter Pupille, Linsensitz und Zonulaapparatstatus, maximal mögliche Erweiterung der Pupille.

Übersicht über Begriffsverknüpfungen

Diese Seite (Version-39) wurde zuletzt am 18-Jul-2013 13:27 von knowweadmin geändert.

Figure 2. The wiki document of the concept "Augenuntersuchung" with subconcepts and relations.

V. Visualisation concepts

Additionally to the wiki-based description of knowledge concepts and the relations, a good performed visualization enhances the fast understanding and leads to a better overview about the refinement structure of the semantic network and the relations within the net. The visualization methods are oriented at the different user interesting cases. As user requests by knowledge acquisition following results are found:

- Obtain an overview of the knowledge base by reducing complexity by using visualization methods.

- Obtain an overview of the processes and dependencies between procedure steps of ophthalmic surgery.

- Browsing through the entire knowledge base to identify interesting spots.

- Retrieve detailed information on special relations between concepts and procedure steps on demand.

- Help the user to find quickly the category of a concept. Concerning these demands, the following visualizations are modeled and realized.

A. Hierarchical forest Visualization

The hierarchical forest visualization is a combination of the process history and the classical graph representation with hierarchical refinement (Fig. 3). This representation is very comfortable to combine time dependant concepts with the temporal relation ”before” or “after” and the “consists of” relation of a concept in the vertical direction. Based on this combined workflow-refinement concept, the user can find out the time scale, where special concepts are integrated. Furthermore a class-subclass structure is represented from each basic concept in the upper temporal process chairs.

B. Circle Pack Visualization

As seen before the hierarchical view of concepts is well represented by a tree structure. However, the view becomes confusing very quickly by presenting the entire content of a large knowledge base. The tree diagram becomes too large when too many nodes and branches must be placed on a single page. Addressing those disadvantages, the Circle Pack visualization (Fig. 4) provides a useful alternative by representing hierarchical relations through containment. It is possible to see an overview of the overall structure and the position of a certain concept. Concepts are displayed as circles. Child-concepts are located inside their parents.

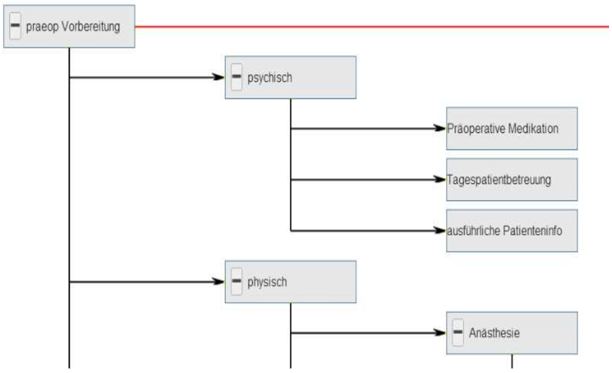


Figure 3. Hierarchical Forest visualization.

.

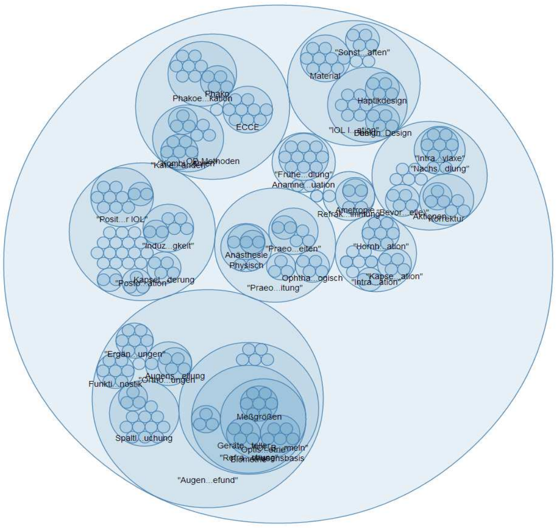


Figure 4. Circle Pack visualization.

C. Reingold Tilford tree

The Reingold Tilford tree (Fig. 5) has advantages by concentrating the representation of many concepts with their relations on a very small space.

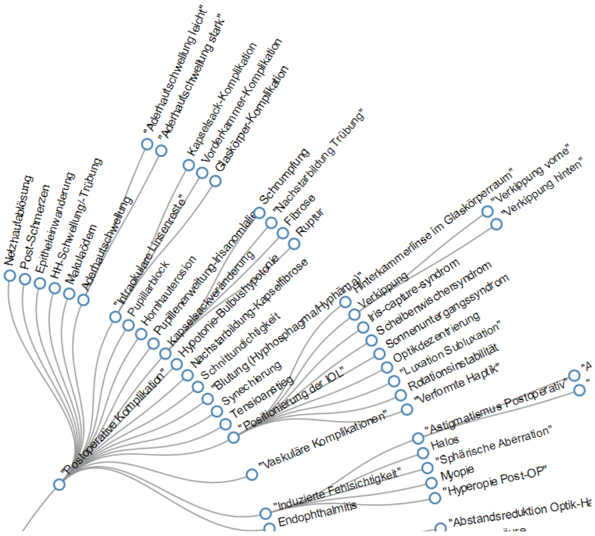


Figure 5. Reingold Tilford tree with radial orientation.

The semantic refinement, based on the subgraphes is performed by a radial orientation. The basic nodes are in the centre. The different sublevels are represented by concentrated circles around the superclass. The level of refinement is shown very directly. The semantic topology is the same as for the other representations, but the graphical visualization is different.

D. *Collapsle Tree*

A collapsible tree (Fig. 6) is a classical representation of hierarchical graphs, well known from the structure of explorer data files. This concept is accepted and very easy understandable. The hierarchical structure is based on the concepts and the refinement method by subconcepts. So, the information, embedded in concepts, can be refined to a very special subclass with specific features (attributes). This fact allows a very comfortable navigation on each semantic level. By selecting interactively a concept, the following subtree is expanded and the specialized subconcepts are graphically represented. Otherwise, concepts, not interesting at the moment can be retracted. The semantic relation is “consists of” with the inverse relation “belongs to”.

All visualizations try to implement the well-known visualization mantra by Shneiderman: Overview First, Zoom and Filter, Then Details-on-Demand [9]. The implementation was realized with the JavaScript library jsPlumb and with d3js.

VI. CONCLUSIONS

The developed knowledge based assistance system is suitable to support the surgeon’s decision for complex cataract operations. Especially the model of the knowledge continuum based on ontologies is responsible and necessary for a correct consistent enhancement of the knowledge domains. The different visualization methods are useful for the development of the ontology as well as for the application in user cases. Situation dependent and user dependent, classical hierarchical tree representations or semantic networks with their refinement possibilities are modeled and realized. They are used from the clinical experts (surgeons) in use of decision support and for tutoring system. The user has also the possibility to switch from one visualization model to another and can select the currently best for him. So, the user has a comfortable access to the information he needs. Because of that the ontology visualization models and their realization are accepted in the clinical process.

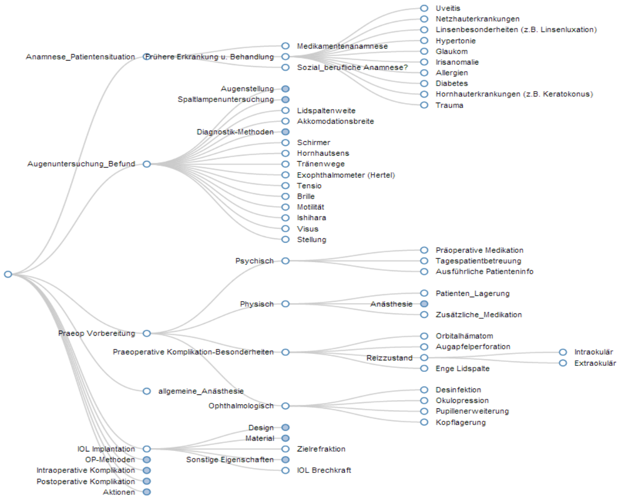


Figure 6. Collapsible Tree visualization.

ACKNOWLEDGMENT

The research and development of the proposed cataract surgical assistance system is funded by the national economical government of Germany. It is proposed to install a commercial clinical system based on the results of the presented prototype at clinical partners. The authors also thank the project executing organisation AiF in Berlin, which is responsible for the allocation of the budget to the research centre and the commercial company.

REFERENCES

[1] A. J. Augustin, „Augenheilkunde“, Springer Verlag, Berlin, Heidelberg, New York, ISBN 3-540-65947-1, 2001.

[2] R. Studer, R. Benjamins, and D. Fensel, “Knowledge engineering” in Principles and methods, Data and Knowledge engineering, vol. 25, 1988, pp. 161-197.

[3] V. R. Benjamins, D. Fensel, P. Gomez, and A. Perez, “Knowledge management through ontologies”, Proceedings on the Int. conf. on Praktical aspects of Knowledge management (PAKM 98), Basel, Schweiz, 1998.

[4] K. P. Scherer, “Hypothesis Generation in the context of an ophthalmic application”, Intern. Conf. on Applied Computer science, Genua, Italy, Oct. 2009, pp. 130-134.

[5] J. Baumeister, J. Reutelshoefer, and F. Puppe, “Engineering Intelligent Systems on the Knowledge Formalization Continuum”, International Journal of Applied Mathematics and Computer Science (AMCS), vol. 21, 2011.

[6] J. Baumeister, J. Reutelshöfer, and F. Puppe, “KnowWE: A Semantic Wiki for Knowledge Engineering”, Applied Intelligence, vol. 35, 2011, pp. 323-344.

[7] M. Musen, “Automated generation of Model-Based Knowledge Acquisition Tools”, Pitman Publishing London, 1989.

[8] M. Molina, and G. Blasco, “Using electronic documents for knowledge acquisition and model maintenance” in Knowledge Based intelligent Information and Engineering Systems, vol. 2774 of LNCS, Springer Berlin, Heidelberg, 2003, pp. 1357-1364.

[9] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations”, in *Proceedings of the IEEE Symposium on Visual Languages*, 1996, pp. 336 - 343.

Towards a Decision Support System for Automated Selection of Optimal Neural Network Instance for Research and Engineering

Rok Tavčar
and Jože Dedič

Cosylab, d.d.
Ljubljana, Slovenia
Email: rok.tavcar@cosylab.com

Drago Bokal

Faculty of Natural Sciences and Mathematics
University of Maribor
Maribor, Slovenia
Email: drago.bokal@uni-mb.si

Andrej Žemva

Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia.
Email: a.zemva@ieee.org

***Abstract*— The success of any advanced computing method (ACM) depends as much on its excellence as it does on a) whether it is optimally deployed and b) if it matches the problem at hand. Neural Networks, which are ACMs of remarkable potential, receive severe penalties in both of the latter aspects, due to reasons, put forth and addressed herein. This paper presents a theoretical foundation for an inference engine decision space and a taxonomic framework for a knowledge base, which are part of our proposed knowledge-driven decision support system (DSS) for optimal matching of a neural network (NN) setup against the given learning task. Such DSS supports solving a multiple criteria optimization problem, considering specific design constraints of the given NN-based machine learning application.**

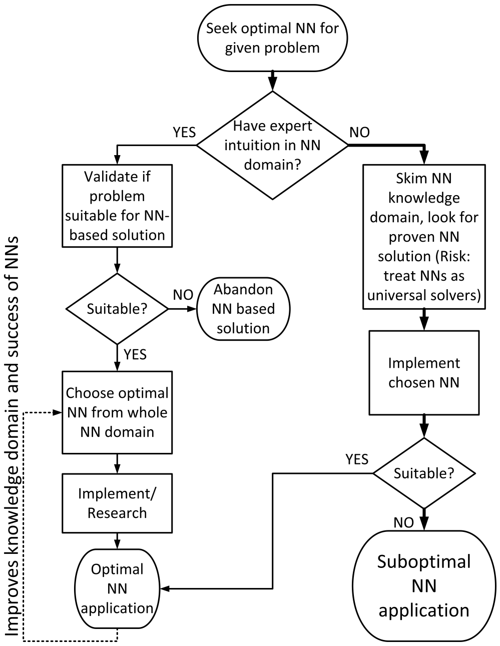
Keywords — Neural Networks; Decision Support System; Knowledge base; Taxonomy; Multiple-Criteria Optimization Problem.

I. Introduction

The compelling notion, that NNs are universal approximators [1], leads quickly to believe, that any NN will do well on any presented machine learning task. However, the universal approximation theorem only guarantees the existence of an approximation, but not that it can be learned, nor that it would be efficient. Practice shows that every given problem requires a carefully crafted NN design and that advanced NN concepts, tailored to specific types of tasks are necessary to attain best results. This factor, among others, has led to the existence of a large number of conceptually varying NN architectures and learning algorithms [2].

For best results, any researcher or practitioner of today needs to understand a vast domain of knowledge in order to find a NN solution most suitable to their task. Due to domain vastness, researchers often limit themselves to NN domains they are familiar with, preventing new knowledge from propagating efficiently among all who would benefit from it. Specifically, we thus face a twofold handicap for progress of NN research: a) practitioners use suboptimal NN setups for real-world applications [2], inhibiting broader NN acceptance in the industry and b) researchers delve into local extrema of research (e.g., through jumping on the bandwagon of imminent peers [3]), pushing frontiers of NN research in suboptimal directions.

Figure 1 shows a simple flowchart view of the current typical approach to selection of NNs for chosen learning task. It can be seen, that the lack of systematic approach to NN selection often yields suboptimal results. This has a negative effect on a wider acceptance of NNs in the industry. A key prerequisite in current NN design is expert intuition, which can be attained either through significant experience with NN implementation and applications in practice, or through access to expert intuition in an environment of experienced NN users. When expert intuition is present in early stages of design, the subsequent efforts give good results (Figure 1, left); when not, design efforts too often lead to suboptimal results (Figure 1, right).



Seek optimal NN for given problem
Have expert intuition in NN domain?
YES
NO
Validate if problem suitable for NN-based solution
Skim NN knowledge domain, look for proven NN solution (Risk: treat NNs as universal solvers)
Suitable?
NO
Abandon NN based solution
YES
Implement chosen NN
Choose optimal NN from whole NN domain
YES
Suitable?
Implement/ Research
NO
Optimal NN application
Suboptimal NN application
Improves knowledge domain and success of NNs

Figure 1. Typical design flow of selection and implementation of NNs for a given learning task.

The NN community needs a streamlined way of enabling existing and potential NN users to make optimal technological choices efficiently and systematically. Having today's foremost

NN research applied in the industry can foster wider acceptance of NNs into practice and improve NN research. In 2006, Taylor and Smith [4] created an important taxonomy-based evaluation of NNs, which aids in validating whether a given problem is solvable with a NN at all. The next concern, which they point out and we hereby address, is to choose the right NN architecture and its concrete implementation for the problem. Our goal is to provide a decision support system (DSS) for industry practitioners and researchers to systematically find the right NN for their application or research interest. This paper proposes a solution that enables (1) a systematical overview of the complete NN knowledge domain to (2) compare NN instances through their capabilities in a (3) quickly interpretative way using a framework that is (4) adaptable in terms of NN properties, even classification dimensions.

The state of the art in NN design methodology can be split into two groups. The first group focuses on choosing the optimal NN *macrostructure* (e.g., Support Vector Machine versus Recurrent Neural Network). In the second group, there are guidelines and (semi)automated methods, that help find the optimal *microstructure* (e.g., number of hidden layers and neurons) of a selected macrostructure. The first group of approaches consists of guidelines and overview literature [5][6][7]. The problem (and virtue) with this set of methodologies is, that they require understanding of a vast set of NN concepts, before the designer is able to make an optimal choice. Dreyfus [5] states, for example: ”No recipes will be provided here. It is our firm belief that no significant application can be developed without a basic understanding of the principles and methodology of model design and training.” Of course, we agree with this position. However, it can be observed in practice, that there is a lack of systematic approach in choosing the macrostructure. As a consequence, Feedforward NN (FFNN) [1], learned with Backpropagation (BP) [21], is still chosen in the majority of applications, which we consider a negative trend [8]. The second group of approaches is necessary for the fine microstructural tuning of a chosen macrostructure (also usually demonstrated on FFNN with BP). These approaches are either given as a set of rules and recipes, or as an automated optimization tool. The most systematic approaches rely on the Design of Experiments (DoE) method, involving Taguchi principles [9][10][11]. Such methods systemize and automate the selection of, e.g., number of hidden layers or neurons, through experimenting with different setups. Similar methods are constructive and pruning algorithms, that add or remove neurons from an initial architecture [12][13]. Also, related are evolutionary strategies, which employ genetic operators for similar purposes [14][15][16].

Our proposed approach fits between these two groups and improves the results of both group’s goals. It exhibits the main qualities of the second group (ability to automate the decision process) and applies them to the problematic of the first group (i.e., choosing the macrostructure), which is a crucial step in NN design, because the effect of any design actions depends greatly on early decisions. The aim of our proposed DSS is to improve the performance of NN-based based applications on a large scale, through enabling designers to perform optimal early design decisions. Figure 2 illustrates how our proposed DSS improves the NN design process by enabling users to systematically find optimal NN instances for their application.

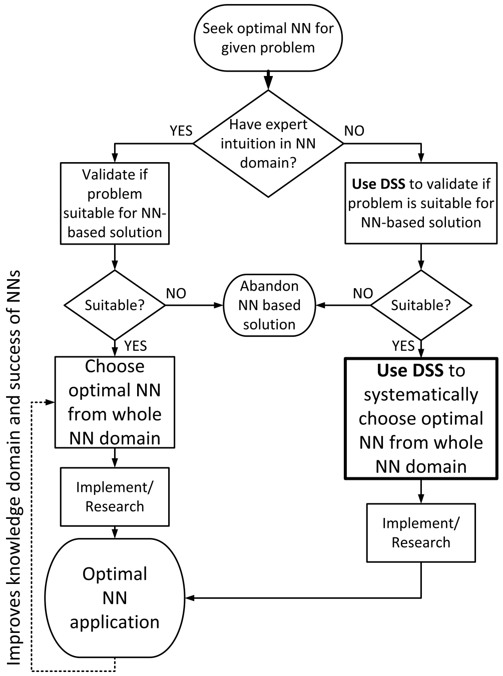

Seek optimal NN for given problem
Have expert intuition in NN domain?
YES
NO
Validate if problem suitable for NN-based solution
Use DSS to validate if problem is suitable for NN-based solution
Suitable?
NO
Abandon NN based solution
NO
Suitable?
YES
YES
Choose optimal NN from whole NN domain
Use DSS to systematically choose optimal NN from whole NN domain
Implement/ Research
Implement/ Research
Optimal NN application
Improves knowledge domain and success of NNs

Figure 2. Our proposed DSS improves the NN design process by enabling users to systematically find optimal NN instances for their application.

At the heart of our proposed DSS is the taxonomic framework that facilitates a qualitative measure between NN instances. However, directly comparing NN instances from literature in detail is prohibitively problematic due to bias or lack of method in the description process [17]. In contrast to related taxonomic efforts, our taxonomy must thus provide a significant level of abstraction, allowing both a complete field overview and sufficient depth to aid qualitative comparison, while providing the flexibility for future adaptations of the proposed classification.

The Andrews-Diederich-Tickle (ADT) taxonomy [18] enables two NNs to be compared pairwise through ADT^5 criteria (defined by Andrews et al. [18] and refined by Tickle et al. [19]), but this taxonomy lacks orthogonality since some of its taxonomical categories (dimensions) are interdependent. Other taxonomies classify NNs purely through topology [20] or realization method [21]. And more recently, researchers create taxonomies that assist in choosing the best solution for the task [3][22] within a limited application area and solve locally what our work solves globally. Our generically specified ranking between feasible solutions permits us to deliver rule-of-thumb guidance that provides an excellent starting point for further in-depth analysis based on, e.g., ADT^5 criteria.

After review of DSS theory in existing literature, we decide to design our proposed DSS as a Knowledge-Driven DSS [23], as it fits our application best. Therefore, our Knowledge-Driven DSS will comprise the following components:

1. Knowledge base
2. User interface
3. Inference engine model
4. Communications component

Corresponding to the above components, our proposed system comprises the following: NN Knowledge base (Section III), 3D visualization of data and qualitative relations (Section IV-A), inference engine and qualitative relations in data (Section IV-B), interaction with 3D environment and entry of objective parameters (Section V), respectively.

II. Inference engine decision space

The main and fundamental result of our work is the conceptualization and theoretical foundation for the inference engine decision space (this Section) and knowledge base framework (Section III), that enables a global overview of the complete NN domain. The decision space, presented hereby, serves also as a coherent terminology and context for our knowledge base framework. Mathematical structure of interrelations must be well defined, to facilitate an effective inference engine, used in solving multiple-objective optimization problems [24]. The decision space is defined via the following descriptors of the NN knowledge domain:

- Set of NN instances $\mathcal{I}$: contains the subset of elements of the NN knowledge domain, which are neural networks. From the whole neural network knowledge domain (NNs, research initiatives, research groups, research goals, application areas, etc.), we gather concrete NN implementations and form the set of NN instances.
- NN classifier $\zeta : \mathcal{I} \to \mathcal{P}$: provides a classification of each member of $\mathcal{I}$ into a particular set of groups $\mathcal{P}$.
- Property $\mathcal{P}$: the co-domain of a classifier ζ, with the latter considered as a function.
- Property value $p_i \in \mathcal{P}$: a specific group of some classifier. It is given a name, which is then identified as this property value.
- NN framework $\mathcal{F}$: ordered list of classifiers relevant for a given user's interest.
- NN universe $\mathcal{U}$: defined by a framework $\mathcal{F}$, it is an $|\mathcal{F}|$-dimensional space, which is Cartesian product of the properties defined by the classifiers in $\mathcal{F}$.
- NN instance $\mathcal{I}_i \in \mathcal{I}$: $\mathcal{I}_i = (p_1, \ldots, p_f)$; an f-tuple of property values, each coming from its corresponding NN property.
- NN category $\mathcal{C}_p \subseteq \mathcal{P}$: subset of a specific property, containing a set of values (classifier groups) of this property. Possibly a singleton.
- NN landscape $\mathcal{L} = C_1 \times C_2 \times C_3 \times \ldots \times C_f$ with at least one C_i being equal to the whole property P_i. Subspace of a NN universe.
- NN type $\mathcal{T} = C_1 \times C_2 \times C_3 \times \ldots \times C_f$: Cartesian product of categories. If all categories in the cartesian product are singletons, the NN type is also a NN instance.
- NN comparator δ: innate comparative quality, defining a partial order $>_\delta$ on the set of NN instances $\mathcal{I}$, by which some pairs of NN instances can be compared. In our proposed DSS, NN comparators are chosen by defining the NN selection criteria (see section III). NN comparators are represented as colored arrows between NN types, with the color specifying different NN instance selection criteria and the thickness of the arrow proportional to number of evidence papers supporting the comparison.
- NN selection criteria ∇: a set of possibly competing NN comparators used for comparison of NN instances.
- Pareto front $\mathcal{R}$ of given NN selection criteria ∇: a set of (discrete) NN instances $j \in \mathcal{R}$ such that whenever some NN instance $i \in \mathcal{I}$ is better than j with respect to some NN comparator $\delta \in \nabla$, i.e., $j >_\delta i$, then there is some other comparator $\delta' \in \nabla$, such that $i >_{\delta'} j$, i.e., i is better than j w.r.t. δ'. In other words, a NN instance belongs to the Pareto front of ∇, if it cannot be improved over without harming at least one of the NN selection criteria in ∇.

What signifies our approach is the decision to abandon the aim for back-to-back comparison of specific NN implementations via rigid criteria (which would limit us to NN research subdomains) and employ a flexible DSS, enabling self-organization of data and allowing the evolution of the framework, together with the evolution of knowledge base contents.

III. Taxonomy for Knowledge Base

With the inference engine decision space theoretically defined in Section II, we proceed to determine the principal dimensions for classification of NN instances. As no single source provides a definitive field overview, we as first step systematically create a taxonomic blueprint for our knowledge base. We define the NN classifiers ζ as operators for sorting of NN instances into main taxonomic branches:

ζ_1 Implementation Platform
ζ_2 NN Architecture
ζ_3 Learning Paradigm
ζ_4 Learning Algorithm
ζ_5 Learning Task

Using our defined NN classifiers, we proceed to build the taxonomy. For its core, we extract the classification used in the book Neural Networks: A Comprehensive Foundation [4], which offers a wide overview of main concepts in NN domain. To build upon this core, we add the overviews of evolutionary methods [25], Spiking Neural Networks [26] and a recent 20-years overview of hardware-friendly neural networks [27]. A principal quality of our system lies in our choice of high abstraction when defining the taxonomy; e.g., while there exist numerous flavors of the BP algorithm, our taxonomy does not differentiate between them. Only by obscuring a such detail, we can achieve a domain-wide overview. Still, as the field of NNs is very diverse, an ultimate taxonomy requires broader community collaboration and finally, consensus; both of which exceed the scope of this work.

We find that our chosen NN classifiers map NN instances into NN properties (i.e., sets of NN categories $\mathcal{C}$, possibly singletons) $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4$ and $\mathcal{P}_5$, respectively:

$\mathcal{P}_1$ (**ζ_1: Implementation Platform)** takes values from:

- General Purpose ($\mathcal{C}_1^1$): Software Simulation on general purpose computer of Von Neumann Architecture (CPU), Digital Signal Processor (DSP) Graphical Processing Unit (GPU), Supercomputer (SCP)
- Dedicated Hardware ($\mathcal{C}_2^1$): Field Programmable Gate Array (FPGA),Neural Hardware / Neural Processing Unit (NPU), Analog Implementation (ANLG), Application Specific Integrated Circuit (ASIC)

$\mathcal{P}_2$ (**ζ_2: NN Architecture)** takes values from:

- Feedforward Neural Network (FFNN)
- Second Generation NNs ($\mathcal{C}_2^2$): Recurrent Neural Network (RNN),Long Short-Term Memory (LSTM)
- Spiking Neural Network (SNN)
- Cellular Neural Network (CNN)
- Self-organizing Map (SOM)
- Reservoir Networks (RSVN) ($\mathcal{C}_6^2$): Echo-state Network (ESN), Liquid-state Machine (LSM)
- Convolutional NN (CONN)
- Deep Belief Network (DBN)
- Hybrid (HYB)

$\mathcal{P}_3$ (**ζ_3: Learning Paradigm)** takes values from:

- Supervised Learning (SUP)
- Reinforcement Learning (REINF)
- Unsupervised Learning (UNSUP)
- Genetic Learning (GENL)

$\mathcal{P}_4$ (**ζ_4: Learning Algorithm)** takes values from:

- Error Correction ($\mathcal{C}_1^4$, ECR): Backpropagation (BP), Extended Kalman Filter (EKF), Stochastic Gradient Descent (SGD),
- Hebbian Learning (HBL)
- Competitive Learning (CPL)
- Evolutionary ($\mathcal{C}_4^4$, EVOL): Evolution of Architecture (EVLARCH), Evolution of Weights (EVLWT), Evolution of Learning Algorithm (EVLALG)
- Reservoir Computing (RSV)
- Hybrid (HYB)

$\mathcal{P}_5$ (**ζ_5: Learning Task)** takes values from:

- Pattern Association ($\mathcal{C}_1^5$): Autoassociation (PASCAUT), Heteroassociation (PASCHET)
- Pattern Recognition ($\mathcal{C}_2^5$, PREC): Natural Language Processing (NLP), Principal Component Analysis (PCA), Speech (SPC), Dimensionality Reduction (DRED), Spatio-temporal (SPT)
- Control ($\mathcal{C}_3^5$, CTL): Indirect (CTLIND), Direct (CTLDIR)
- Function Approximation ($\mathcal{C}_5^5$, FAPPROX): System Identification (SYSID), Inverse System (INVSYS)
- Classification (CSF)
- Regression (RGR)

Property $\mathcal{P}_2$ thus comprises 11 NN property values, gathered in 9 categories, of which $\mathcal{C}_2^2$ and $\mathcal{C}_6^2$ each contain two property values; $\mathcal{C}_2^2$ contains p_2 and p_3 and $\mathcal{C}_6^2$ contains p_7 and p_8. Property value indices run free from category indices.

The presented property values and categories can be further manipulated and refined. However, to enable efficient domain overview, a significant level of abstraction is required. For further detailed inspection, more specialized taxonomies can be used (see Section I). For example, the Backpropagation learning algorithm has a multitude of variants [17], but for a comprehensive overview, abstraction is crucial.

A. Qualitative comparison through NN selection criteria ∇

With the taxonomic backbone defined, we can proceed with classification of NN instances from processed literature through property values $\mathcal{P}$, using our set of NN classifiers ζ. This comparative dimension, well-defined but very permitting, is a core facility of our knowledge base and the heart of our DSS' inference engine. Therefore, we also extract from literature sources the qualitative comparison information between NN instances w.r.t. the following set of chosen NN selection criteria ∇:

δ_1 **Low cost of ownership** (feasibility, practicality, low hardware cost, low development complexity, presence of user community)

δ_2 **Capability** (effectiveness, convergence speed, generalization performance, benchmark success, high learning rate, low error)

δ_3 **Real-time requirement** (speed of execution, on-line vs. off-line learning, pre-learned vs. adaptive learning)

δ_4 **Design maturity**(proven solution vs. emerging technology)

While estimates for all NN criteria can be extracted from literature or provided by a domain expert, design maturity could also be automatically calculated as a measure of occurrence frequency in literature.

B. 5-letter notation and knowledge base formation

In the 5-dimensional NN universe, defined by our NN framework $\mathcal{F}$, that we define in Section III through selecting our set of NN classifiers $\zeta_{1,\ldots,5}$, each NN instance is described via five NN properties $\mathcal{P}_{1,\ldots,5}$. Therefore, each element in the database compares two NN instances or NN landscapes in terms of five parameters. To construct our formal notation, we build upon the idea of 3-letter notation used in the theory of scheduling problems [28] and adapt it to a 5-letter notation for describing NN instances. Our resulting formal representation of relation(s) between two NN instances is as follows:

$$(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \mathcal{P}_5) >_{\delta_{i\ldots n}} (\mathcal{P}_1', \mathcal{P}_2', \mathcal{P}_3', \mathcal{P}_4', \mathcal{P}_5'), \tag{1}$$

where each NN property $\mathcal{P}_{1\ldots5}$ can be a comma-separated list of elements (NN property values), n is the total number of selection criteria and where $i, i \in \{1 \ldots n\}$ denotes the group of indices of NN qualifiers, by which the 'greater' NN instance is superior to the 'lesser' NN instance.

In our knowledge database, the following example statement extracted from a scientific source [27]: "FPGA is a superior implementation platform to ASIC in terms of flexibility and cost for implementations of FFNN or RNN with supervised or reinforcement learning, using stochastic learning algorithms." is formally denoted as follows:

$$\begin{aligned}&(\mathcal{P}_1[4], \mathcal{P}_2[3,4], \mathcal{P}_3[1,2], \mathcal{P}_4[3], \mathcal{P}_5[x]) >_{\delta_2}\\&(\mathcal{P}_1[6], \mathcal{P}_2[3,4], \mathcal{P}_3[1,2], \mathcal{P}_4[3], \mathcal{P}_5[x]),\end{aligned} \tag{2}$$

Or:

$$\begin{aligned}&(FPGA, \{FFNN, RNN\}, \{SUP, REINF\},\\ &SGD, x)\\ &>_{\delta_{cost,flexibility}}\\ &(ASIC, \{FFNN, RNN\}, \{SUP, REINF\},\\ &SGD, x)\end{aligned} \tag{3}$$

This example also illustrates the case where the paper does not specify all property values (in this example, the learning task $\mathcal{P}_5[x]$), the statement is incomplete and it may mean either that the relation is indifferent to that property, or that there is no information present about that property's role in the relationship. After reviewing selected literature (e.g., [27][29][25][30]–[36]), we get a number of such specific statements that comprise our knowledge base seed information, which serves as basis for development of our inference engine and visualization scheme.

IV. Result: knowledge-driven DSS with inference engine and visualization tool

The proposed inference engine, together with knowledge base visualization, are the final results of our efforts presented in this paper. Both modules operate on the data in the knowledge database in a read-only fashion. In the following subsections, we present our scheme for exploratory visualization of our multidimensional knowledge database and describe our interactive inference engine.

A. Visualization scheme

Every point in the NN universe's graphical representation corresponds to one NN instance. The most valuable information in our knowledge database is the qualitative comparison between NN instances. This is shown in Figure 3, illustrating the graphical representation of Statement (3) from Section III-B. We have found, that using three dimensions for the visualization is optimal, because it allows users to navigate the environment interactively and to recognize interdependencies, even after switching between the chosen set of three dimensions. The 3D visualization can only represent three dimensions at a time and the user can explore the NN knowledge domain using any dimension set.

Figure 4 shows the 3D representation our NN universe $\mathcal{U}$, containing points from our prototype knowledge base. This view allows users to examine the NN knowledge domain in a full 3D environment, visually exploring (through zoom and rotation of view around any axis) the comparative relations between NN instances. Axes correspond to NN properties $\mathcal{P}$; each dot corresponds to a single NN instance $\mathcal{I}_i$; arrows represent qualitative comparators $\delta_{1...4}$ between two NN instances; arrow thickness and dot size indicate the quantity of source papers (database entries) for the shown information; call-out-type labels are references to source literature. Each of the selection criteria is assigned its own arrow color (red, magenta, blue and black for δ_1, δ_2, δ_3 and δ_4, respectively). Coloring of NN instances aids in visual comparison (blue is better, green is worse). Using the inference engine (Section IV-B), the visualization can be actively augmented according to the user's decision input.

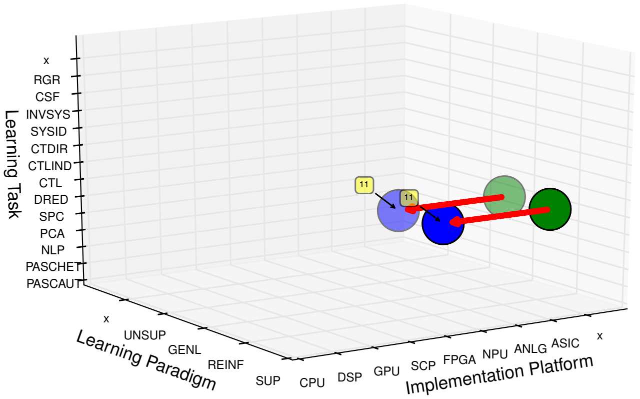


Figure 3. Example graphical representation of qualitative relation between two NN instances. The figure represents Statement (3) from Section III-B.

B. Inference Engine

Our inference engine approaches our NN instance selection process as a multiple-objective optimization problem [24] and applies a Pareto front method [37], using our Pareto front $\mathcal{R}$, as defined in Section II, to find the suitable, multiple, non-dominant solutions. After the user specifies their boundary conditions and sets weights of the NN selection criteria ∇ through the graphical user interface, the DSS automatically identifies the discrete-equivalent of Pareto front $\mathcal{R}$ and the user can directly locate and examine the source literature, relating the NN instances in $\mathcal{R}$. Rating of alternatives is based on a weighted pairwise comparison matrix [38], resulting in levels within a discrete-space equivalent of Pareto front, which guide the user towards NN instances, specified as superior with respect to their criteria. The user can iteratively and interactively further fine-tune the selection of best candidates via weights of their criteria ∇, to determine the optimal NN instance for their problem, until the final choice is made. Information, inferred by the inference engine, is also used as input into the visualization tool, to augment the database visualization by superimposing relationships, marking Pareto points and their scores, hiding a subset of NN instances, etc. (see Figure 5). Both the visualization tool and the inference engine can be extended with additional inference and visualization functions. Section V gives further insight into the typical application of the inference engine, through step-by-step explanation.

V. Practical example of DSS use

The two major user groups that can gain remarkable benefits from using our proposed DSS, are **Industry Practitioner** and **Academic Researcher**. Both user groups share the main interest of finding the optimal NN instance for their scenario, but have a different angle: a) the industry practitioner's goal is to find the **best fitting, well proven** NN implementation

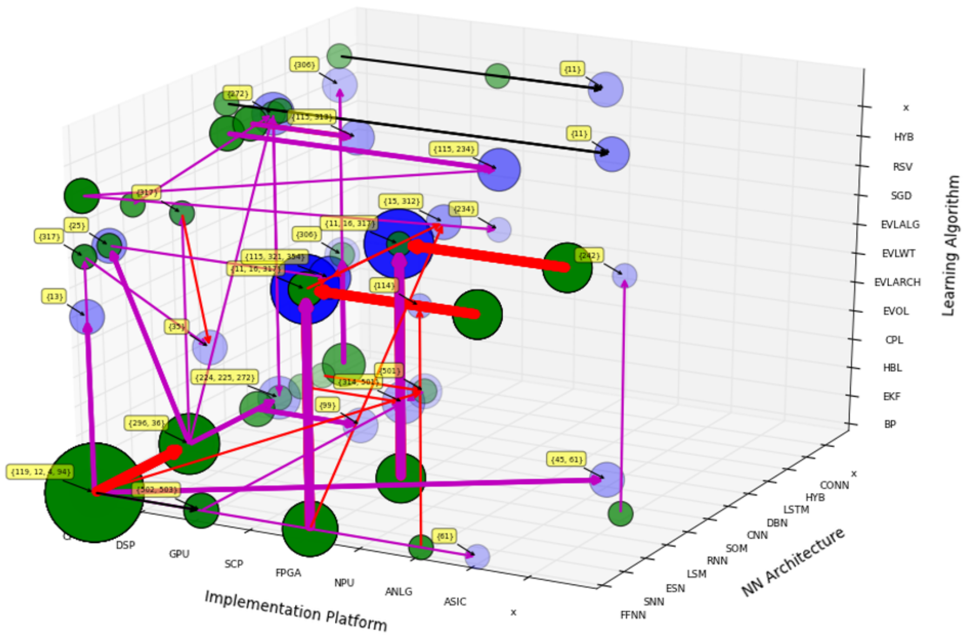


Figure 4. The 3D representation our NN universe $\mathcal{U}$, containing points from our prototype knowledge base. See Section III for axis label definitions.

for their **application** (with set boundary conditions on task type, implementation platform, etc.), and b) the academic researcher's aim is to find an active research area or synergies between domains, to systematically selects the most meaningful **research direction**. In this section, we illustrate step-by-step a typical use case for the industry practitioner. Let our example demand a **highly accurate** and **real-time capable** NN instance for image-based **object recognition** using **supervised learning**. The following steps illustrate how this ground truth is used with our DSS as decision input and how the inference engine results are interpreted and used:

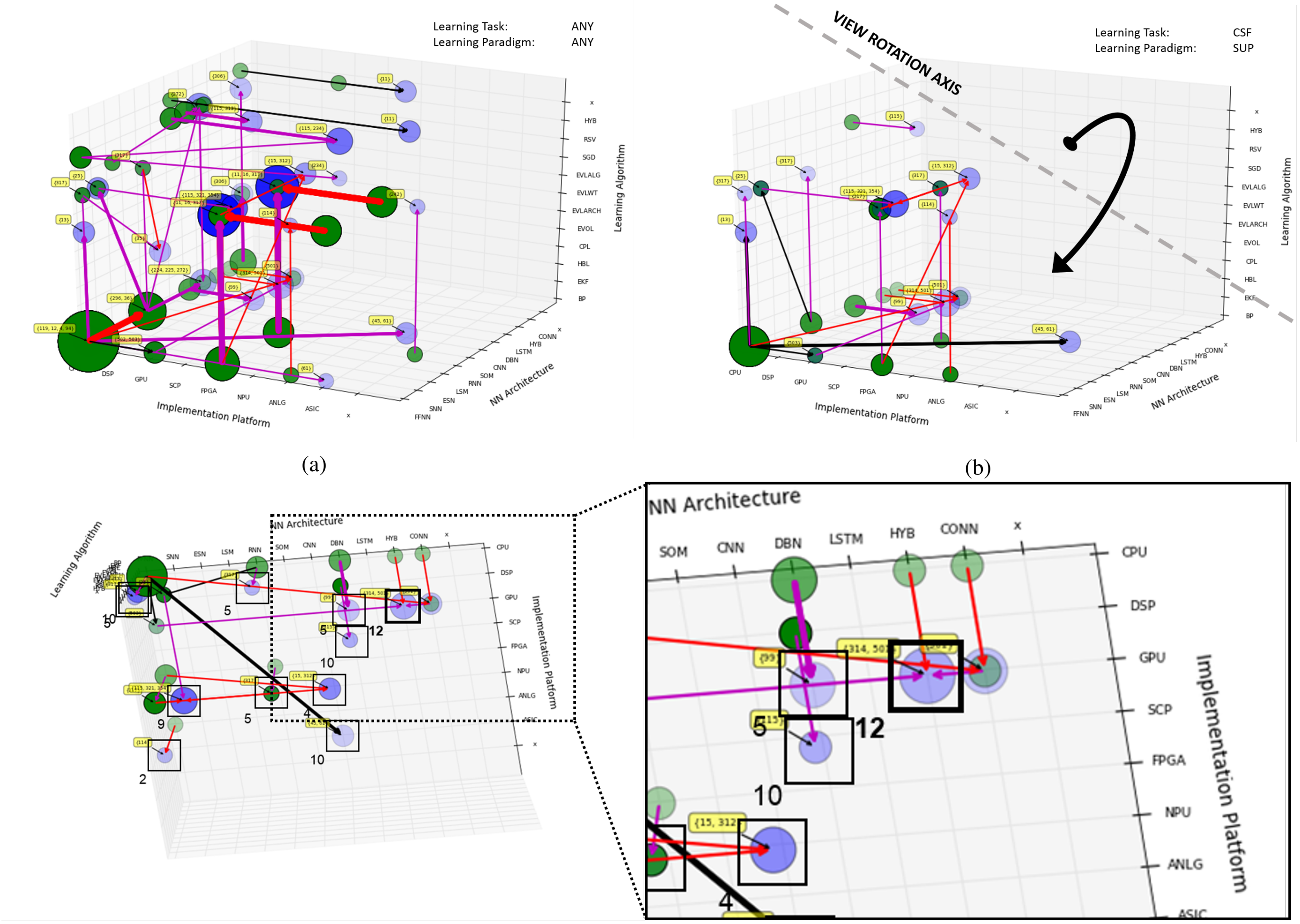
1) Enter task requirements into DSS: the practitioner enters their set of boundary conditions by selecting the NN instance properties, that are defined by the application. In our example, these are the learning task and learning paradigm. The DSS considers these two properties as pivots, therefore, only the **remaining** three properties (dimensions) are shown in the 3D visualization tool (Figure 5a). After the axes are determined, the user specifies pivot axis values (learning task = classification, CSF and learning paradigm = supervised, SUP). The effect of this is shown in Figure 5b, where only those NN instances are shown, whose pivot property values are as specified by the user. Thus, this step narrows down the search down to three dimensions and defines the NN landscape, which optimal solutions can be chosen from. If more than two pivot axes are specified by the use case, the NN landscape is 2- or 1-dimensional, further focusing the search.

2) Set weights for selection criteria ∇: once the 3D NN landscape is defined in the previous step, the user specifies weights for each of ∇ within the range from -5 to 5. In our example, the selected weights for $\delta_{1...4}$ are 2, 5, 5 and 3, respectively (see Section III-A for list of criteria).

3) Examine Pareto front $\mathcal{R}$: based on weighted criteria, the inference engine extracts NN instances, that belong to the discrete Pareto front $\mathcal{R}$. These are NN instances, for which there is no NN instance superior w.r.t. any of the selection criteria (no arrows leaving the NN instance). These points are highlighted by the DSS via black squares. For better viewing of the points in $\mathcal{R}$, the user can interact with the 3D view by rotation around any axis. This is seen in 5c (left), showing the $\mathcal{R}$ points in an updated view, obtained by rotating 5b around the 'view rotation axis' in the indicated direction. In the lower left corner of each $\mathcal{R}$-marking is the NN instance's score (closeup view in Figure 5c, right), calculated by the inference engine using the weighted pairwise comparison matrix.

4) Analyze top alternative in Pareto front: the user chooses the highest-ranking NN instances in the Pareto frontier and analyzes their corresponding source literature, indicated by call-outs (see Figure 3). Our example gives the highest score of 12 to NN instance, described in database entries 314 and 502 (Figure 5c). From corresponding source papers [32] and [33], the practitioner learns, that a) FFNNs can be used as convolution NNs, b) GPU implementation in [33] has better flexibility than previously known implementations, c) GPU implementation of CONN has better real-time capabilities than CPU implementation, d) Hybrid between pure CONN and FFNN has better recognition performance than any of these two used alone, e) hybrid implementation in [33] has won an impressive series of image classification competitions, etc.

In conclusion, based on the industry practitioner's input criteria, the DSS recommends, that a GPU-based hybrid CONV-FFNN NN should be investigated as best choice for the given use case. This simple case illustrates how a user can, using our DSS in a few simple steps, rapidly traverse an immensely diverse knowledge base, in order to choose an optimal direction for further investigation, and finally, concrete implementation.



(a)

(b)

(c)

Figure 5. Rapid assessment of complete knowledge domain. Typical step-by-step application of the inference engine, together with the visualization scheme, used for finding an optimal NN instance, based on user input parameters.

VI. Conclusion and Future Work

In this work, we have identified the need for an abstract-level overview of the NN knowledge domain and alleviate the barriers, which an industry practitioner or researcher meet, when selecting the right NN instance or research direction for their specific scenario. We devised a theoretical foundation for a decision support system, comprising a knowledge database and inference engine, that can automate the decision process of choosing the best NN architecture for the task at hand. We also presented a prototype implementation and a proof-of-concept through step-by-step use of our DSS. This illustrated its potential in aiding users to exploit the whole knowledge of NN research domain and improve NN results in research and industry, through choosing optimal approaches to machine learning problems.

In our future work, we will study how the inference engine could be expanded to automatically find promising combinations of NN properties, based on current highest-scoring NN instances within the database. This will enable our system to automatically highlight synergies between existing approaches. Future work also includes making the knowledge base and its visualized interaction accessible online. Moderated, collaborative editing of the knowledge base among researchers is also considered. Once the knowledge base reaches critical mass, researchers will be motivated to contribute their own work, or populate it with entries where they notice a lack of coverage. An editorial group could, on a per need basis, revise the taxonomy when novel NN properties or categories emerge. The proposed 5-letter notation enables automatic parsing of the literature, keeping the knowledge database up-to-date at all times and solving this problem once and for all. An automatically-generated dynamic survey paper could always be kept up-to-date and available in printed form for a quick overview of recent developments.

Acknowledgment

Operation part financed by the European Union, European Social Fund. The authors would like to thank the reviewers for their valuable comments that contributed greatly in improving this paper.

References

[1] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural networks, vol. 2, no. 5, 1989, pp. 359–366.

[2] B. M. Wilamowski, “Neural network architectures and learning algorithms,” Industrial Electronics Magazine, IEEE, vol. 3, no. 4, 2009, pp. 56–63.

[3] H. Jacobsson, “Rule extraction from recurrent neural networks: A taxonomy and review,” Neural Computation, vol. 17, no. 6, 2005, pp. 1223–1263.

[4] B. Taylor and J. Smith, “Validation of neural networks via taxonomic evaluation,” in Methods and Procedures for the Verification and Validation of Artificial Neural Networks. Springer US, 2006, pp. 51–95.

[5] G. Dreyfus, “Modeling with neural networks: Principles and model design methodology,” in Neural Networks. Springer Berlin Heidelberg, 2005, pp. 85–201.

[6] A. Omondi and J. C. Rajapakse, FPGA Implementations of Neural Networks. Springer Netherlands, 2006.

[7] Y. Huang, “Advances in artificial neural networks–methodological development and application,” Algorithms, vol. 2, no. 3, 2009, pp. 973–1007.

[8] C. Moraga, “Design of neural networks,” in Knowledge-Based Intelligent Information and Engineering Systems. Springer, 2007, pp. 26–33.

[9] J. Ortiz-Rodríguez, M. Martínez-Blanco, and H. Vega-Carrillo, “Robust design of artificial neural networks applying the taguchi methodology and doe,” in Electronics, Robotics and Automotive Mechanics Conference, 2006, vol. 2. IEEE, 2006, pp. 131–136.

[10] E. Inohira and H. Yokoi, “An optimal design method for artificial neural networks by using the design of experiments.” JACIII, vol. 11, no. 6, 2007, pp. 593–599.

[11] J. F. Khaw, B. Lim, and L. E. Lim, “Optimal design of neural networks using the taguchi method,” Neurocomputing, vol. 7, no. 3, 1995, pp. 225 – 245.

[12] J.-f. Qiao, Y. Zhang, and H.-g. Han, “Fast unit pruning algorithm for feedforward neural network design,” Applied Mathematics and Computation, vol. 205, no. 2, 2008, pp. 622–627.

[13] H. Han and J. Qiao, “A self-organizing fuzzy neural network based on a growing-and-pruning algorithm,” Fuzzy Systems, IEEE Transactions on, vol. 18, no. 6, 2010, pp. 1129–1143.

[14] S. G. Mendivil, O. Castillo, and P. Melin, “Optimization of artificial neural network architectures for time series prediction using parallel genetic algorithms,” in Soft Computing for Hybrid Intelligent Systems. Springer, 2008, pp. 387–399.

[15] Z.-j. Zheng and S.-q. Zheng, “Study on a mutation operator in evolving neural networks,” Journal of Software, vol. 13, no. 4, 2002, pp. 726–731.

[16] G. G. Yen, “Multi-objective evolutionary algorithm for radial basis function neural network design,” in Multi-Objective Machine Learning. Springer, 2006, pp. 221–239.

[17] M.-T. Vakil-Baghmisheh and N. Pavesic, “A fast simplified fuzzy artmap network,” Neural Processing Letters, vol. 17, no. 3, 2003/06/01 2003, pp. 273–316.

[18] R. Andrews, J. Diederich, and A. B. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” Knowledge-Based Systems, vol. 8, no. 6, 1995, pp. 373–389.

[19] A. Tickle, R. Andrews, M. Golea, and J. Diederich, “The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks,” Neural Networks, IEEE Transactions on, vol. 9, no. 6, 1998, pp. 1057–1068.

[20] E. Fiesler, “Neural network classification and formalization,” Computer Standards & Interfaces, vol. 16, no. 3, 1994, pp. 231–239.

[21] R. Rojas, “Neutral networks: A systematic introduction,” Springer, 1996.

[22] H. R. Maier, A. Jain, G. C. Dandy, and K. P. Sudheer, “Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions,” Environmental Modelling & Software, vol. 25, no. 8, 2010, pp. 891–909.

[23] D. J. Power, Decision support systems: concepts and resources for managers. Greenwood Publishing Group, 2002.

[24] N. Xiong and M. L. Ortiz, “Principles and state-of-the-art of engineering optimization techniques,” in ADVCOMP 2013, The Seventh International Conference on Advanced Engineering Computing and Applications in Sciences, 2013, pp. 36–42.

[25] S. Ding, H. Li, C. Su, J. Yu, and F. Jin, “Evolutionary artificial neural networks: a review,” Artificial Intelligence Review, 2013, pp. 1–10.

[26] S. Ghosh-Dastidar and H. Adeli, “Spiking neural networks,” International Journal of Neural Systems, vol. 19, no. 04, 2009, pp. 295–308.

[27] J. Misra and I. Saha, “Artificial neural networks in hardware: A survey of two decades of progress,” Neurocomputing, vol. 74, no. 1-3, 2010, pp. 239–255.

[28] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. Rinnooy Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” Annals of Discrete Mathematics. v5, 1977, pp. 287–326.

[29] L. Fortuna, P. Arena, D. Balya, and A. Zarandy, “Cellular neural networks: a paradigm for nonlinear spatio-temporal processing,” Circuits and Systems Magazine, IEEE, vol. 1, no. 4, 2001, pp. 6–21.

[30] J. Schmidhuber, D. Wierstra, M. Gagliolo, and F. Gomez, “Training recurrent networks by evolino,” Neural Computation, vol. 19, no. 3, 2007, pp. 757–779.

[31] G. Andrienko et al., “Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns,” Computer Graphics Forum, vol. 29, no. 3, 2010, pp. 913–922.

[32] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “”the german traffic sign recognition benchmark: a multi-class classification competition”,” in Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011, pp. 1453–1460.

[33] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” Neural Networks, vol. 32, no. 0, 2012, pp. 333–338.

[34] R. Raina, A. Madhavan, and A. Y. Ng, “Large-scale deep unsupervised learning using graphics processors,” in ICML, vol. 9, 2009, pp. 873–880.

[35] D. Coyle, “Neural network based auto association and time-series prediction for biosignal processing in brain-computer interfaces,” Computational Intelligence Magazine, IEEE, vol. 4, no. 4, 2009, pp. 47–59.

[36] R. K. Al Seyab and Y. Cao, “Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation,” Journal of Process Control, vol. 18, no. 6, 2008, pp. 568–581.

[37] M. Farshbaf and M.-R. Feizi-Derakhshi, “Multi-objective optimization of graph partitioning using genetic algorithms,” in Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP ’09. Third International Conference on, Oct 2009, pp. 1–6.

[38] S. Ghodsypour and C. O’Brien, “A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming,” International Journal of Production Economics, vol. 56–57, no. 0, 1998, pp. 199 – 212, production Economics: The Link Between Technology And Management.

A Memory Controller for the DIMM Tree Architecture

Young-Jong Jang, Young-Kyu Kim, Taewoong Ahn, and Byungin Moon
School of Electronics Engineering
Kyungpook National University
Daegu, Republic of Korea
e-mail: youngjong25@ee.knu.ac.kr, kyk79@ee.knu.ac.kr, myannet11@gmail.com, bihmoon@knu.ac.kr

***Abstract*—The dual in-line memory module (DIMM) tree architecture was proposed to solve signal integrity and data access latency problems of many-DIMM system. Although the DIMM tree demands a memory controller specific to it, there has been little research on the memory controller for the DIMM tree. For this reason, this paper proposes a new memory controller architecture for the DIMM tree. This architecture was modeled using DRAMSim2 for its verification and analysis, and the experimental results show that the proposed DIMM tree memory controller works properly and efficiently.**

Keywords-DIMM tree; many-DIMM system; DIMM to DIMM transfer; memory controller

I. INTRODUCTION

In the traditional computer systems, there are two memory access methods: one is the multi-drop bus based memory access and the other is the point-to-point link-based memory access. The former method is mainly used in the traditional computer systems [1], but it is not appropriate for implementation of large-capacity main memory systems due to its signal integrity issues [2]. In the latter method, DIMM modules are connected through daisy chains. The fully buffered DIMM (FB-DIMM) is an example that uses point-to-point links [3]. In this method, the control signals are buffered and repeated at each DIMM. This method has the disadvantage that as the number of connected DIMM modules increases, the transmission time of control signals also increases. In order to resolve these problems, the DIMM tree architecture has been proposed. The DIMM tree architecture overcomes the transmission delay time and signal integrity issues by a tree structure connection of DIMM modules [4].

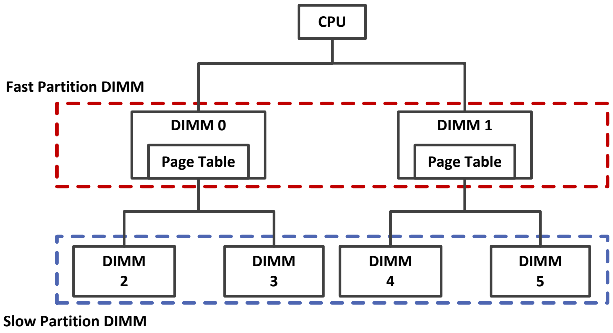


Figure 1. Partitioned DIMM tree structure.

The structure and operation of the DIMM tree architecture were presented for the implementation of the DIMM tree architecture, and the concept of the partitioned DIMM tree architecture and the direct DIMM-to-DIMM transfer was established in order to efficiently manage large-capacity memory of the DIMM tree architecture [5]. The memory controller for the DIMM tree architecture is also very important for the implementation of the DIMM tree architecture. However, there was little concrete discussion on the memory controller for the DIMM tree architecture. Thus, this paper proposes a hardware architecture of a memory controller for the management of the DIMM tree architecture.

The rest of the paper is organized as follows. Section 2 describes the background of the DIMM tree architecture. Section 3 presents the hardware architecture of the proposed DIMM tree memory controller. In Section 4, we present the experimental environments, and analyze the results of the experiments. Finally, Section 5 concludes the paper.

II. DIMM TREE ARCHITECTURE

Therdsteerasukdi et al. [4] presented the DIMM tree architecture of better scalability by connecting DIMM modules in a tree structure to solve signal integrity and access latency problems. Then, they improved their research by proposing the partitioned DIMM tree and the direct DIMM-to-DIMM transfer for efficient memory management.

A. *Partitioned DIMM Tree*

The partitioned DIMM tree architecture consists of a fast partition and a slow partition, as in shown Fig. 1 [5]. The fast partition corresponds to the DIMMs at a level closer to the CPU. Thus, it takes one DIMM access latency to access the fast partition. The slow partition contains the remaining DIMMs. The fast partition is used as a cache of the slow partition, which works as the main memory. The relation between the fast partition and the slow partition is similar to that between main memory and hard disk in a virtual memory system, so data transfer between the fast partition and slow partition is performed by page units. In addition, the page fault handler to process page faults is necessary in the memory controller [5][6].

B. *Page Fault Handler*

The partitioned DIMM tree architecture uses a small space in the fast partition for a fast partition page table to manage page translation between the fast and slow partitions. When memory access is requested from the CPU, the fast partition page table in the fast partition is checked to find the fast partition page corresponding to the requested page of the slow partition. When a page fault occurs, the page fault handler updates the requested page to the fast partition from the slow partition, and writes back the replaced page to the slow partition from the fast partition. These update and write back are processed with the proposed direct DIMM-to-DIMM transfer [5]. This direct DIMM-to-DIMM transfer reduces the overhead of page fault processing.

III. PROPOSED DIMM TREE MEMORY CONTROLLER

In this section, a hardware architecture of a memory controller is proposed for the DIMM tree architecture described in Section 2. The proposed hardware architecture of the DIMM tree memory controller is shown in Fig. 2. The organization and operation of the DIMM tree memory controller is as follows.

When a read or write request is entered from the CPU to the bus, and the request is loaded into the Transaction Buffer, and then four fast partition page table entries, which are accessed using the least significant bits of the requested slow partition page number as the index, as shown in Fig. 3, are loaded into the Page Table Buffer. The Hit/Miss Control Unit is used to check whether a page fault occurs or not for the request. This module compares the tag of the slow partition address with each of the slow partition page number tags of four page table entries loaded on the Page Table Buffer. When one of the four page table entry tags matches the tag of the slow partition address and the valid bit of the corresponding page table entry is 1, the state becomes ‘hit’. Otherwise the state is ‘miss’ meaning that a page fault occurs. In case of ‘hit’, the hit/miss signal of the Hit/Miss Control Unit is set to 1, and the Page Table Buffer outputs the way number of the ‘hit’ page table entry.

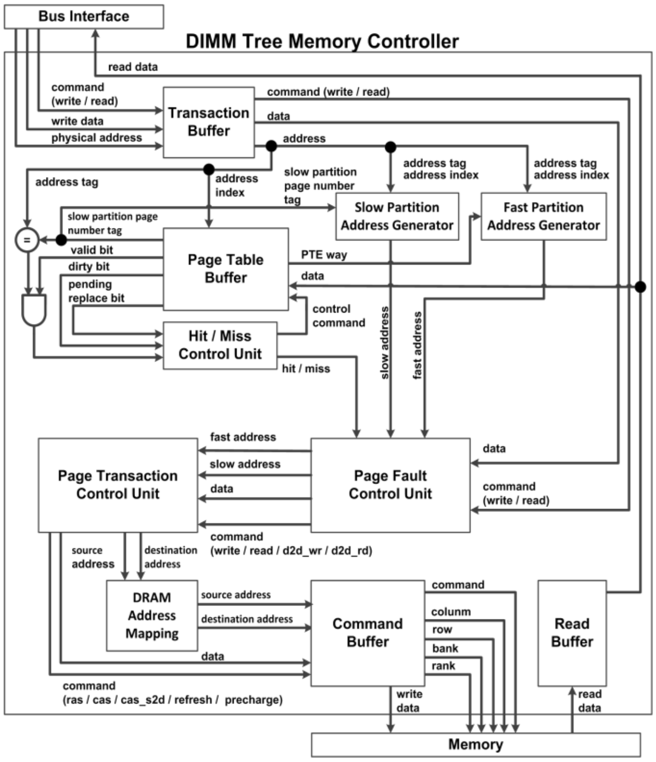


Figure 2. Architecture of the proposed DIMM tree memory controller.

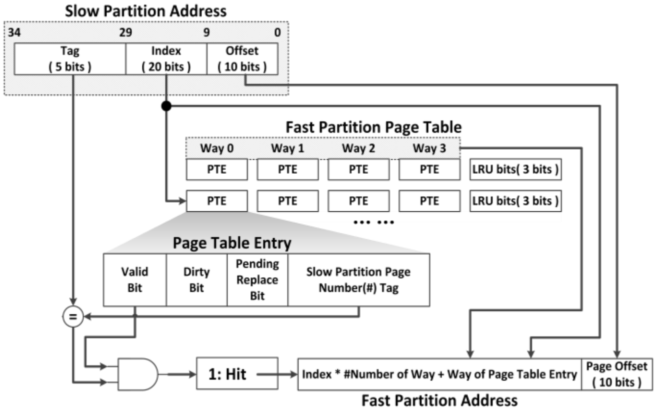


Figure 3. Address translation from the slow partition to fast partition.

The Page Fault Control Unit generates the DIMM-to-DIMM transfer command. When the hit/miss signal from the Hit/Miss Control Unit is “1”, the Page Fault Control Unit directly transfer the request command in the Transaction Buffer to the Page Transaction Control Unit. However, when a page fault occurs, Some DIMM-to-DIMM commands are inserted. When the page-faulted request command is a write, the Page Fault Control Unit inserts the DIMM-to-DIMM update command (D2D_RD) to transfer the faulted page from the slow partition to the fast partition. After the write request command, the DIMM-to-DIMM writeback command (D2D_WR) is issued to write the corresponding page of the fast partition to the slow partition. When the request command is a read, the Page Fault Control Unit inserts the DIMM-to-DIMM update command (D2D_RD) for getting the faulted page of the slow partition to the fast partition before issuing the read request command to read data from the fast partition. These generated command streams are transferred to the Page Transaction Control Unit.

The Page Transaction Control Unit processes the commands from the Page Fault Control Unit, and makes DRAM control commands such as row active (RAS), column read (CAS), CAS source to destination (CAS_S2D), REFRESH, and PRECHARGE. The CAS_S2D is a special command for the DIMM tree architecture [5], but the others are common commands for controlling DRAM. When the command from the Page Fault Control Unit is either a write or read, the Page Transaction Control Unit generates RAS and CAS commands. Since the CPU can only access the fast partition, the RAS and CAS from the Page Transaction Control Unit are associated with the fast partition. On the other hand, if the command from the Page Fault Control Unit is either D2D_WR or D2D_RD, the Page Transaction Control Unit generates RAS and CAS_S2D commands. The D2D_WR and D2D_RD have two addresses, one is a source address and the other is a destination address. The source address and destination address are determined depending on the type of the command. If the command is D2D_WR, the source address is a fast partition address, and the destination address is a slow partition address. It is vice versa in case of the D2D_RD command.

TABLE I. SIMPLESCALAR PARAMETERS

Simulation Tool	SIM-cache
Processor ISA	PISA instruction
L1 D-cache	256 sets, 32 bytes cache line, 4-way set-associative, LRU policy
L1 I-cache	
L2 D-cache	1024 sets, 64 bytes cache line, 4-way set-associative, LRU policy
L2 I-cache	
I-TLB	16 sets, 4096 bytes cache line, 4-way set-associative, LRU policy
D-TLB	32 sets, 4096 bytes cache line, 4-way set-associative, LRU policy

TABLE II. DRAMSIM2 PARAMETERS

DRAM type	DDR3-1600[a]
DRAM Data Width	8 bytes
Row buffer policy	close page policy
Bank per rank	8
Row count	16384
Column count	1024
DIMM size	1 GB
Page size	1 KB

a. Micron, x8 DDR3-1600 DIMM [12]

The command generated by the Page Transaction Control Unit, is transferred to the Command Buffer. The Command Buffer sends the DIMM tree commands (RAS, CAS and CAS_S2D) and addresses (rank, bank, row and column) to DIMM modules, which process those commands.

IV. EXPERIMENTS

For the verification of the proposed DIMM tree memory controller hardware architecture, we modeled the proposed hardware architecture by DRAMSim2 [7] and generated six workloads each of which has 500 thousand requests for the simulation.

A. Workload

For the performance verification of the proposed DIMM tree memory controller hardware architecture, we need workload of large size with a wide range of memory addresses. For this reason, we extracted the memory transaction traces by running bzip2, gromacs, hmmer, lbm, mcf and milc in the SPEC CPU 2006 benchmark [8] in the SimpleScalar simulator [9]. Bzip2, gromacs and hmmer use a wide range of memory addresses, but have high locality. Lbm, mcf and milc have high locality and use a small range of memory addresses [10]. Experiments were carried out using the SimpleScalar with the parameters describes in Table Ⅰ. Also, the PIN [11] was used to attach the time stamp to the memory transaction traces extracted from the SimpleScalar.

B. Experimental Environments

The DRAMSim2 was used for modeling the proposed DIMM tree memory controller. It is assumed that capacity of each DIMM is 1GB and the page size is 1 KB. The DIMM tree architecture has a tree structure with the depth of 2 and the degree of 4, as shown in Fig. 4. The fast partition has the size of 4 GB, and the slow partition is composed of 16 GB. The block diagram of the proposed DIMM tree memory controller model is shown in Fig. 5. The detailed parameters applied to DRAMSim2 is shown in Table Ⅱ.

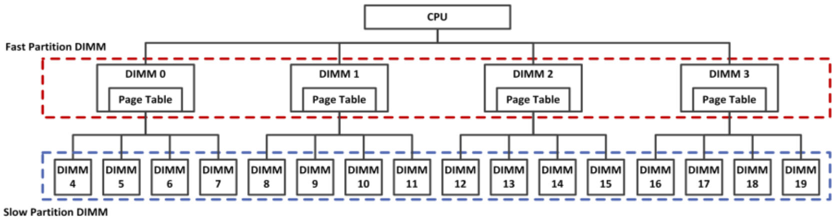


Figure 4. DIMM tree structure used in experiments.

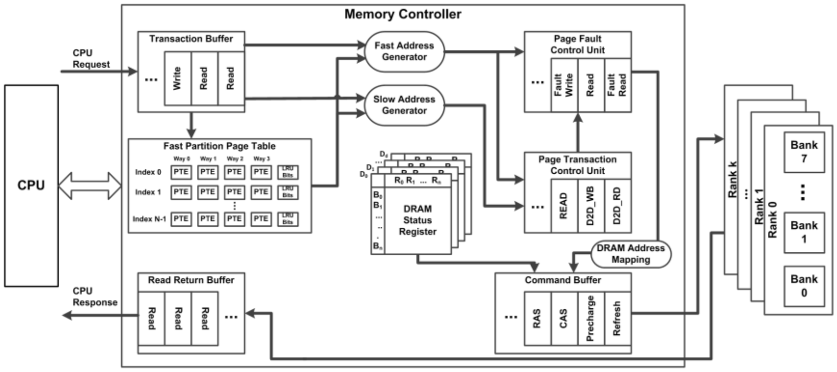


Figure 5. Simulation environment of the proposed DIMM tree memory controller.

C. Results

The generated workloads, each of which has a total of 500 thousand requests, are used for the CPU Requests of the DIMM tree memory controller model of DRAMSim2. Through simulations with workloads, it was verified that the proposed DIMM tree memory controller operates properly. In addition, performance analysis of the proposed DIMM tree memory controller was carried out as follows.

Fig. 6 shows hit rate per 1,000 traces when a total of 10 thousand traces of each workload are processed. Lbm, mcf and milc are memory-intensive workload, thus have high hit rate. After 5 thousands of traces have been processed, the hit rates of lbm, mcf and milc was maintained at about 73%. On the other hand, hit rates of bzip2, gromacs and hmmer were continuously increased. As shown in Fig. 7, the hit rates of gromacs and hmmer increased rapidly after 50 thousands of traces were processed. This is mainly because that the number of compulsory misses, which occur intensively at the early time of simulation, decreases after running 50 thousand traces. Workloads such as bzip2, gromacs and hmmer have a wide address range, so they need a longer time to fill the fast partition pages. The number of cycles taken to run all the traces is shown in Table Ⅲ. The performance for the workload hmmer is measured best because the hit rate for hmmer is best.

TABLE III. CLOCK CYCLES TAKEN TO PROCESS EACH WORKLOAD

Workloads	10000 traces (cycles)	100,000 traces (cycles)	500,000 traces (cycles)
bzip2	1,263,919	12,720,015	39,481,566
gromacs	538,966	3,409,885	11,668,567
hmmer	508,589	1,999,400	6,265,317
lbm	522,217	5,168,498	25,814,382
mcf	535,822	5,276,435	26,298,294
milc	535,397	5,279,990	26329965

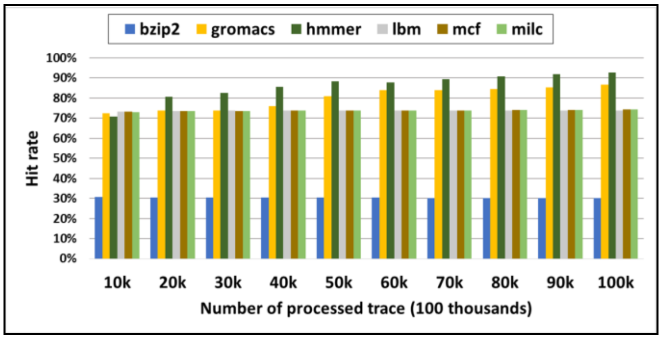


Figure 7. Hit rate per 10,000 traces when running 100,000 traces of each workload.

V. CONCLUSIONS AND FUTURE WORK

This paper proposed a hardware architecture of the DIMM tree memory controller. The proposed architecture was modeled using the DRAMSim2, Workloads were generated using the SPEC CPU 2006 to verify the functionality and performance. The experimental results show that the proposed DIMM tree memory controller works properly. This paper presented a DIMM tree memory controller which can be used as a reference model in DIMM tree based large memory systems. The study of this paper assumes that he DIMM tree memory controller operates on the single-processor environment. However, recently, the multiprocessor environment is used widely. So, our future work is to run multiple loads in parallel on the DIMM tree memory controller for multiprocessor systems.

ACKNOWLEDGMENT

This investigation was financially supported by Semiconductor Industry Collaborative Project between Kyungpook National University and Samsung Electronics Co. Ltd. This study was supported by the BK21 Plus project funded by the Ministry of Education, Korea (21A20131600011). This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the C-ITRC (Convergence In-formation Technology Research Center) support program (NIPA-2014-H0401-14-1004) supervised by the NIPA (National IT Industry Promotion Agency).

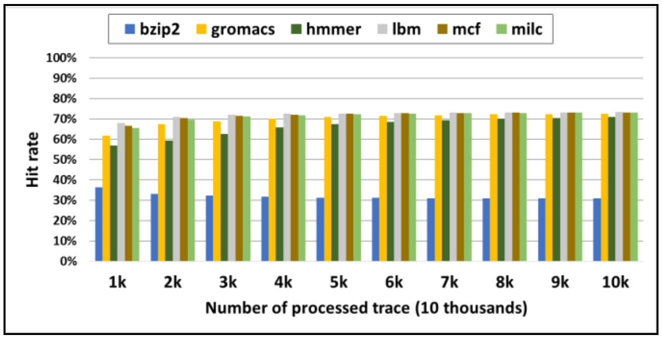


Figure 6. Hit rate per 1,000 traces when running 10,000 traces of each workload.

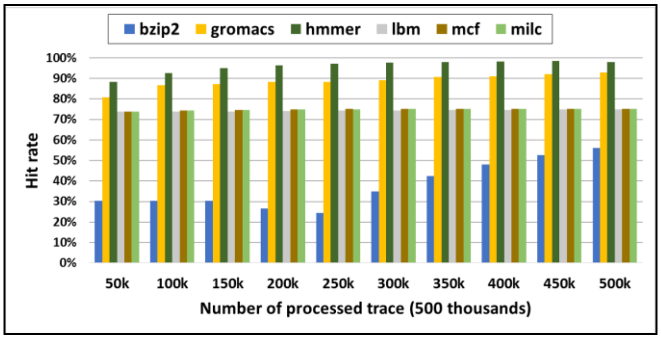


Figure 8. Hit rate per 50,000 traces when running 500,000 traces of each workload.

REFERENCES

[1] J. H. Kim, et al., “Challenges and Solutions for Next Generation Main Memory Systems,” Proc. IEEE 18th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS 09), Oct. 2009, pp. 93-96, doi: 10.1109/EPEPS.2009.5338468.

[2] B. Jacob, S. Ng, and D. Wang, “Memory Systems: Cache, Dram, Disk,” Morgan Kaufmann, 2008, pp. 377-391.

[3] B. Ganesh, A. Jaleel, D. Wang, and B. Jacob, “Fully-Buffered DIMM Memory Architectures: Understanding Mechanisms, Overheads and Scaling,” Proc. IEEE 13th International Symposium on High Performance Computer Architecture (HPCA 07), Feb. 2007, pp. 109-120, doi: 10.1109/HPCA.2007.346190.

[4] K. Therdsteerasukdi, et al., “The DIMM Tree Architecture: A High Bandwidth and Scalable Memory System,” Proc. IEEE 29th International Conference on Computer Design (ICCD 11), Oct. 2011, pp. 388–395, doi: 10.1109/ICCD.2011.6081428.

[5] K. Therdsteerasukdi, et al., “Utilizing Radio-Frequency Interconnect for a Many-DIMM DRAM System,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 2, no. 2, June 2012, pp. 210–227, doi: 10.1109/JETCAS.2012.2193843.

[6] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, “Scalable High Performance Main Memory System Using Phase-Change Memory Technology,” Proc. 36th annual international symposium on Computer architecture (ISCA 09), June 2009, pp. 24–33, doi: 10.1145/1555754.1555760.

[7] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, “DRAMsim2: A Cycle Accurate Memory System Simulator,” IEEE Computer Architecture Letters, vol. 10, no. 1, June 2011, pp. 16-19, doi: 10.1109/L-CA.2011.4.

[8] J. L. Henning, “Spec CPU2006 Benchmark Descriptions,” ACM SIGARCH Computer Architecture News, vol. 34. no. 4, Aug. 2006, pp. 1–17, doi: 10.1145/1186736.1186737.

[9] T. Austin, L. Eric, and D. Ernst, “SimpleScalar: An Infrastructure for Computer System Modeling,” IEEE Computer, vol. 35, no. 2, Feb. 2002, pp. 59-67, doi: 10.1109/2.982917.

[10] X. Mingli, D. Tong, Y. Feng, K. Huang, and X. Cheng, “Page policy control with memory partitioning for DRAM performance and power efficiency,” IEEE International Symposium on Low Power Electronics and Design (ISLPED 13), Sept. 2013, pp. 298-303, doi: 10.1109/ISLPED.2013.6629312.

[11] V. Reddi, A. M.Settle, D. A. Connors, and R. S. Cohn. “PIN: A Binary Instrumentation Tool for Computer Architecture Research and Education,” Proc. ACM Workshop on Computer Architecture Education (WCAE 04): held in conjunction with the 31st International Symposium on Computer Architecture, June. 2004, p. 22, doi: 10.1145/1275571.1275600.

[12] Micron. 1 Gb: x4,x8,x16 DDR3 SDRAM Features 2006.

Classification of Pattern using Support Vector Machines: An Application for Automatic Speech Recognition

Gracieth Batista*, Washington Silva† and Orlando Filho†
*Student of Electrical Engineering
Federal Institute of Maranhão, São Luis, Maranhão, Brazil 65053-155
Email: gracieth.cavalcanti@hotmail.com
†Department of Electrical and Electronics
Federal Institute of Maranhão
Email: washington.wlss@ifma.edu.br and orlando.rocha@ifma.edu.br

***Abstract*—This paper proposes the implementation of a Support Vector Machine (SVM) for automatic recognition of numerical speech commands. Besides the pre-processing of the speech signal with mel-ceptral coefficients. Also, this paper is used to Discrete Cosine Transform (DCT) to generate a two-dimensional matrix used as input to SVM algorithm for generating the pattern of words to be recognized. The Support Vector Machines represent a new approach to pattern classification. SVM is used to recognize speech patterns from the mean and variance of the speech signal input through the two-dimensional array aforementioned, the algorithm trains and tests those data showing the best response. Finally, the experimental results are presented for the speech recognition applied to Brazilian Portuguese language process.**

***Keywords*–Support Vector Machines; Classification; Pattern Recognition; Statistical Learning Theory; Application in Speech Recognition.**

I. Introduction

A. Digital Processing of the Speech Signal

Digital speech processing is a specialty in full expansion. There are numerous applications of this research area, we can refer to automatic speech recognition for purposes of interpretation of commands by machines or robots, automatic speech recognition for the purpose of biometric authentication, recognition of pathology in the mechanism of speech production for biometric and or medicinal purposes. The speech processing systems are divided basically into three sub-areas: speech coding, speech synthesis and automatic speech recognition. Regardless of the specific purpose, the initial stages of a system for processing digital speech is sampling followed by segmentation of words or phonemes [1] for short-term analysis by Fourier transform [2] or by spectral analysis [2]. The speech signal processing first involves obtaining a parametric representation based on a certain model and then applying a transformation to represent the signal in a more convenient form for recognition. The last step in the process is the extraction of important characteristics for a given application. This step can be performed either by human listeners or automatically by machines [2]. Among the techniques that have been developed for segmentation of speech, those based on Hidden Markov Models (HMM) [2] are quite traditional. Hybrid methods based on Artificial Neural Networks (ANN) [3] and criteria, such as average energy, selection of voiced phonemes and non voiced, Mel Frequency Cepstral Coefficients (MFCCs) [2], spectral metrics [2], and others, are also used. Speech coding systems include those cases in which the purpose is to obtain a parametric representation of the speech signal, based on the analysis of the frequency, average power and other characteristics of the spectrum of the signals. The techniques of encoding the speech signal are used both for transmission and for compact storage of speech signals. One of the main applications of speech coding is to transmit the speech signal efficiently [4]. Systems for automatic speech recognition or Speech Recognition Systems (SRS) are focused on the recognition of the human voice by intelligent machines.

B. Methodology Proposed

This article uses as a recognition default locutions from Brazillian Portuguese of the digits $'0','1','2','3','4','5','6','7','8','9'$. The speech signal is sampled and encoded in mel-cepstral coefficients and coefficients of Discrete Cosine Transform (DCT) [2] in order to parameterize the signal with a reduced number of parameters. Then, it generates two dimensional matrices referring to the mean and variance of each digit. The elements of these matrices representing two-dimensional temporal patterns will be classified by Support Vector Machines (SVMs) [3]. The innovation of this work is in the reduced number of parameters lies in the SVM classifier and in the reduction of computational load caused by this reduction of parameters.

C. SVM (Support Vector Machine)

Based on Statistical Learning Theory, SVM classifier is another category of feed-forward, whose outputs of neurons from a layer feed neurons from the next layer where feedback doesn’t occur [3]. This technique originally developed for binary classification, seeks to build hyperplanes as decision surfaces, in such a way so that the separation between classes is maximum, assuming that the patterns are linearly separable. As for non-linearly separable patterns, the SVM seeks an appropriate mapping function to make the mapped set linearly separable. Due to its efficiency in working with high-dimensional data, it is cited in the literature as a highly robust technique [5]. The results of applying this technique are comparable and often superior to those obtained by other learning algorithms, such as ANN.

1) Theory of Statistical Learning: The Theory of Statistical Learning aims to establish mathematical conditions that allow the selection of a classifier with good performance for the data set available for training and testing. In other words, this theory seeks to find a good classifier with good generalization regarding the entire data set. But, this classifier abstains from particular cases, which defines the capability to correctly predict the class of new data from the same domain in which the learning occured. **Machines Learning (ML)** [3] employs an inference principle called induction, in which general conclusions are obtained from a particular set of examples. A model of supervised learning based on Theory of Statistical Learning is given in Figure 1 [3].

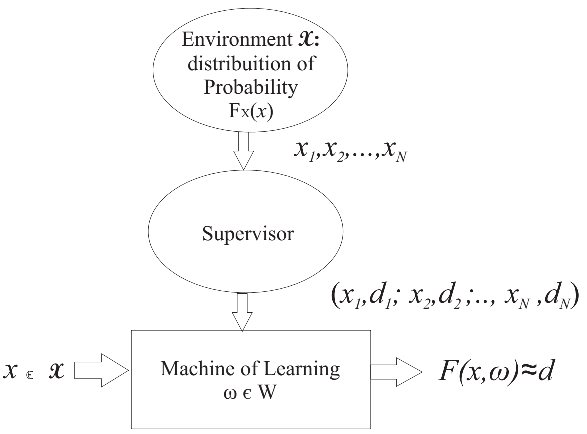


Figure 1. Flowchart of a model of supervised learning.

Environment: It is stationary. It provides an input vector x with a function of distribution of cumulative probability fixed, but unknown $F_x(x)$.

Supervisor: It shows a desired response d for each input vector x is provided by the environment accordance to a conditional cumulative distribution function $F_x(x|d)$ which is also fixed but unknown. The desired response due to input vector x is related by (1):

$$d = f(x, v) \tag{1}$$

where v is noise that allows the supervisor to be noisy. The kind of learning discussed in this work is supervised, but not noisy.

2) Functional of Risk: The desired performance of a classifier f is that it gets the smallest mistake during training, with the error being measured by the number of incorrect predictions of f. Therefore, its defined as Empirical Risk $Remp(f)$ the extent of loss between the desired response and the actual response. In (2), it is shown the definition of the Empirical Risk.

$$Remp(w) = \frac{1}{N}\sum_{i=1}^{N} c(f_i(x_i, y_i)) \tag{2}$$

where $y_i = F(x_i, w_i)$, w_i is a vector of adjustable weights, c is the cost function related to the prediction $f(x_i)$, with desired output $f(y_i)$, where one type of cost function is the "loss 0/1" defined by (3). The process of search by an equation $f(x)$ that represents a smaller value of $Remp(f)$ is called Empirical Risk Minimization.

$$c(f_i(x_i, y_i) = \begin{cases} 1, & \text{if} \quad y_i f(x_i) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Assuming that the patterns used for training (x_i, y_i) are generated by an independent and identically distributed distribution (iid) of probability $P(x, y)$. The probability of incorrect Classification from classifier f is called Functional Risk, which quantifies the capability of generalization, according to (4) [6].

$$R(f) = \int c(f(x_i, y_i))dP(x_i, y_i) \tag{4}$$

During the training process, $Remp(f)$ can be easily obtained, while $R(f)$ cannot, since probability P is unknown. Given a set of training data (xi, yi) with $x_i \in \Re^N$ and $y_i \in \{\pm 1\}, i = 1, 2, ..., n,$ i = 1, 2, ..., n, the input vector x_i and y_i is the output related to class x_i, then the goal is to estimate a function $f : \Re^N \rightarrow \{\pm 1\}$ and if no restriction is imposed on the class of functions in which one chooses to estimate f, it may happen that the function obtains a good performance in the training set, but not having the same performance in unknown patterns. This phenomenon is called the error "*overfitting*". Thus, the minimization of the empirical risk does not guarantee a good generalization capability, and being a great classifier is desired f^* such that $R(f^*) = min_{f \in F} R(f)$, where F is the set of possible functions f . The Theory of Statistical Learning provides ways to limit the class of functions (hyperplanes), in order to exclude bad models, that is, those leading to the error of overfitting, implementing a function with an adequate capacity to correctly classify the set of training data. Restrictions on Risk Functional use the concept of VC dimension [7].

3) SVM (Mathematical Modeling): Classifiers that separate the data through a hyperplane are called Linear and SVM fits this definition, therefore, we must pay attention to all that there is to train and classify, for as a SVM must also deal with non-linearly separable sets, this will resort to techniques. In the application of Techniques of Statistics Learning (TSL), the classifier must be chosen the classifier with the lowest possible empirical risk and which also satisfies the constraint of belonging to a family F with a small VC dimension. Also, to determine the separability of the optimal hyperplane, as it was assumed that the training set is linearly separable. The equation of a decision surface folows below:

$$\omega^T x + b = 0 \tag{5}$$

where x is an input vector, ω is a vector of adjustable weight (maximum separation possible between true and false examples) and b is a *bias*. And from this consideration follows a sequence of calculations in order to find the hyperplane with higher separability between classes. Under these conditions, the surface found is called optimal. In Figure 2, the geometry of an optimal hyperplane for two-dimensional space is illustrated.

For the case of a non-linear set, SVM creates another feature space from the original space, and the concepts and

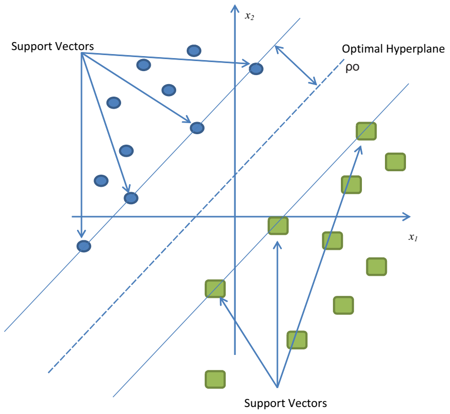


Figure 2. Optimal hyperplane for linearly separable patterns.

calculations of linear optimal hyperplane are applied in this new space [3].

4) SVM for multiple classes: The SVM is a dichotomic algorithm, that is, for pattern classfication based on two classes [3]. However, it is possible to obtain a classifier for multiple classes using the SVM algorithm. Scholkopf et al. proposed a classifier model of type "one vs. all" [8]. Clarkson and Brown have proposed a classifier model of the "one vs. one" [9]. However, both models are indeed classifiers of only two classes: Class +1 and Class -1 [3]. On system "one vs. all", one machine for each group is used, in which each group is trained separately from the rest of the set. In the system "one vs. one", only three machines are used, in which a group is classified against another; then, this one is rated against another group and so on, until the whole set is trained.

5) Functions Kernel: The decision surface of the SVM, which in the feature space is always linear, usually is nonlinear in the input space. As seen earlier, the idea of SVM depends on two mathematical operations:

1) Nonlinear mapping of an input vector into a feature space of high dimensionality, which is hidden from the entry and exit;
2) Build an optimal hyperplane to separate the features discovered in the first step. To design the optimal hyperplane, a kernel function is needed, or a core of the inner product. A Kernel function is a function that receives two points x_i and x_j of the input space and calculates the scalar product of the data in the feature space, given by (6).

$$k(x_i, x_j) = \Phi^T(x_i) \cdot \Phi(x_j) \qquad (6)$$

To ensure the convexity of the optimization problem and introduce the Kernel mapping in which the calculation of scalar products is possible, a kernel function that follows the conditions set by Mercers Theorem [10][11]must be used. The kernels that satisfy Mercers conditions are characterized for giving origin to semi-definite positive matrices k, in which each element k_{ij} is defined by $k_{ij} = k(x_i, x_j)$, $\forall i, j = 1, 2, ..., n$. Once the mapping is performed by a SVM kernel function, and not directly by $\Phi(x)$, it is not always possible to know exactly which mapping is actually performed, because the kernel functions perform an implicit mapping. Table I shows the main features commonly used as kernel functions. The expansion of the inner product core $K(x_i, x_j)$ in (6) makes it possible to find a decision surface that is non-linear in the input space, but whose image in the feature space is linear [3].

TABLE I. APPLICATION OF SVM

Name of Kernel	Function
Polinomial	$\left(x^T x_i + 1\right)^p$
RBF Kernel	$exp\left(-\frac{1}{2\sigma^2}\lVert x - x_i \rVert^2\right)$
Perceptron	$tanh\left(\beta_0 x^T x_i + \beta_1\right)$

The Kernel functions used have the following restrictions:

- In the Polynomial kernel, the parameter p is first specified by the user;
- In the kernel RBF, the parameter σ^2 is common to all cores;
- In the perceptron, Mercer's theorem is satisfied only for some values of β_0, β_1;

6) Automatic Systems for Speech Recognition with SVM: Hidden Markov Models (HMMs) have become the most employed technique for Automatic Speech Recognition (ASR). However, the HMM-based ASR systems may reach their limit of performance. Hybrid systems based on a combination of artificial intelligence techniques provide significant improvements of performance. However, the progress in this paradigm has been hindered by their training computational requireents, which were excessive when these systems were proposed. Recently, several methods of Speech Recognition have been proposed using mel-frequency cepstral coefficients and Neural Networks Classifiers [12][13][14], Sparse Systems for Speech Recognition [15], Hybrid Robust Voice Activity Detection System [16], Wolof Speech Recognition with Limited vocabulary Based HMM and Toolkit [17], Real-Time Robust Speech Recognition using Compact Support Vector Machines [6].

Thus, the SVM has many functions; it is a binary algorithm, based in the Theory of Statistical Learning and in the Functional of Risk. And, finally, it has many functions for classification, such as in the case of multiple classes.

II. SYSTEM OF SPEECH RECOGNITION

A. *Pre-processing of Speech Signal*

Initially, after the segmentation of the speech is passed through the process of windowing, the speech signal is sampled and segmented into frames and is encoded in a set of melcepstral parameters. The number of parameters obtained is determined by the order of mel-cepstral coefficients. The obtained coefficients are then encoded by Discrete Cosine Transform (DCT) [2] in a two dimensional matrix that will represent the speech signal that to be recognized. The process of windowing in a given signal, aims to select a small portion of this signal, which will be analysed and named frame. A short-term Fourier analysis performed on these frames is called signal analysis frame by frame. The length of the frame T_f is defined as the length of time upon which a parameter set is valid. The term frame is used to determine the length of time between successive calculations of parameters. Normally, for speech processing, the time frame is between 10ms and 30ms [18].

B. Generation of two-dimensional DCT-temporal matrix

After being properly parameterized in mel-cepstral coefficients, the signal is encoded by DCT performed in a sequence of T observation vectors of mel-cepstral coefficients on the time axis. The coding by DCT is given by the equation following:

$$C_k(n,T) = \frac{1}{N}\sum_{t=1}^{T} mfcc_k(t) cos\frac{(2t+1)n\pi}{2T} \tag{7}$$

where $k, 1 \leq k \leq K$, refers to the $k-$th line (number of Mel frequency cepstral coefficients) of $t-$th segment of the matrix n, $1 \leq n \leq N$ component refers to the $n-$th column (order of DCT), $mfcc_k(t)$ represents the mel-cepstral coefficients. Thus, one obtains the two-dimensional matrix that encode the long term variations of the spectral envelope of the speech signal [19]. This procedure is performed for each spoken word. Thus, there is a two-dimensional matrix $C_k(n,T) \equiv C_{kn}$ for each input signal. The matrix elements are obtained as the following:

1) For a given model of spoken word ***P*** (digit), ten examples of this model are pronounced. Each example is properly divided into T frames distributed along the time axis. Thus, we have: $P_0^0, P_1^0, ..., P_9^0, P_0^1, P_1^1 ..., P_9^1, P_0^2, P_1^2, ..., P_9^2,..., P_m^j$, where j=0,1,2,...,9 is the number of patterns to be recognized and m=1,2,3,...,10, is the number of samples to generate each pattern.
2) Each frame of a given example of model ***P*** generates a total of K mel-frequency cepstral coefficients, and then, significant characteristics are obtained within each frame over this time. The DCT of order N is then calculated for each mel-cepstral coefficient of the same order within the frame, that is, c_1 in the frame t_1, c_1 in the frame $t_2, ...,$ c_1 in the frame t_T, c_2 in the frame t_1, c_2 in the frame $t_2, ...,$ c_2 in the frame t_T, and so on, generating elements $\{c_{11}, c_{12}, c_{13}, ..., c_{1N}\}$, $\{c_{21}, c_{22}, c_{23}, ..., c_{2N}\}$, $\{c_{K1}, c_{K2}, c_{K3}, ..., c_{KN}\}$ in the matrix given in (7). Thus, a two-dimensional temporal array DCT is generated for each m example of model ***P***, represented by C_{kn}^{jm}. Finally, arrays of mean CM_{kn}^j (8) e variance CV_{kn}^j (9) are generated. The parameters of CM_{kn}^j and CV_{kn}^j are used as datas of input in SVM algorithm.

$$CM_{kn}^j = \frac{1}{M}\sum_{m=0}^{M-1} C_{kn}^{jm} \tag{8}$$

$$CV_{kn}^j = \frac{1}{M-1}\sum_{m=0}^{M-1}\left[C_{kn}^{jm} - \left(\frac{1}{M}\sum_{m=0}^{M-1} C_{kn}^{jm}\right)\right]^2 \tag{9}$$

C. Generation of machines

In the technical literature about SVMs, the standards are called classes. The mean and variance matrices are transformed in two column vectors, CMe (vector with means) and $CVar$ (vector with variances).

$$\begin{aligned} CMe_i^j \;\; = \;\; & \langle CM_{11}^0, CM_{12}^0, ..., CM_{1N}^0, CM_{21}^0, CM_{22}^0, ..., CM_{2N}^0, \\ & CM_{KN}^J \rangle \end{aligned} \tag{10}$$

$$\begin{aligned} CVar_i^j \;\; = \;\; & \langle CV_{11}^0, CV_{12}^0, ..., CV_{1N}^0, CV_{21}^0, CV_{22}^0, ..., CV_{2N}^0, \\ & CV_{KN}^J \rangle \end{aligned} \tag{11}$$

For example, in the case of a matrix CM_{22}^j, that is, where K=2 e N=2, the matrices CMe and $CVar$ take the following form:

$$\begin{aligned} CMe_i^j \;\; = \;\; & \langle CM_{11}^0, CM_{12}^0, CM_{21}^0, CM_{22}^0, CM_{11}^1, \\ & CM_{12}^1, CM_{21}^1, CM_{22}^1, ..., CM_{22}^J \rangle \end{aligned} \tag{12}$$

$$\begin{aligned} CVar_i^j \;\; = \;\; & \langle CV_{11}^0, CV_{12}^0, CV_{21}^0, CV_{22}^0, CV_{11}^1, CV_{12}^1, \\ & CV_{21}^1, CV_{22}^1, ..., CV_{22}^J \rangle \end{aligned} \tag{13}$$

Each class in this example is represented by 4 elements in the vector of mean and 4 elements in vector of variance according to (12) and (13), that is, the first 4 elements of the vector of mean and of the vector of variance refer into class 0, the following 4 elements of each vector to the class 1, and so on. Figure 3 shows data of the peers of mean and variance of the speech signals from the examples of (12) and (13).

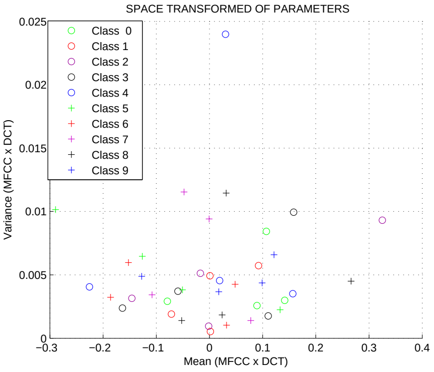


Figure 3. Classes and their different points.

The set of functions mapping of type input-output is given by (14):

$$\Omega = f\left([CMe_i^j; CVar_i^j], w\right) \tag{14}$$

where Ω is the real response produced by the learning machine associated with the entry pairs of means and variances, and w is a set of free parameters, called weights for weighting, selected from the parameter space related to patterns. Figure 4 shows a general model of the supervised learning from the examples, having three components:

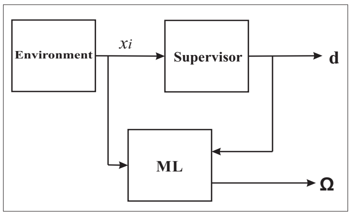


Figure 4. Model of Learning.

The **Environment** is the fixed input system; this yields x_i (points that come from the pairs of coordinates $(CMe, CVar)$) from the response of the DCT matrix of speech signals. The **Supervisor** returns a value of the desired output d_i for each input vector x_i in accordance with a conditional distribution function $F\left(d_i|x_i\right)$), also set. **Machine of Learning (ML)**, is an algorithm capable of implementing a set of functions $f\left([CMe_i^j; CVar_i^j], w\right)$, where $\omega \in W$, where W is a set of parameters belonging to the set of desired responses. In this context, the learning problem can be interpreted as a **problem of approximation**, which involves finding a function $f\left([CMe_i^j; CVar_i^j], w\right)$ that generates the best approximation to the Ω output of the supervisor. The selection is based on a set of independent training examples I and identically distributed (iid), generated according to:

$$F(x, d) = F(x)F(d|x) : (x_i, d_i) \qquad (15)$$

where (x_i, d_i) are peers with desired input and output with $d_i \in R^n$ and $i = 1, ..., I$.

III. Experimental Results

A. Training

After performing the pre-processing of the speech signal coding and generation of temporal matrices CM_{kn}^j and CV_{kn}^j, the models were trained by SVM machines CM_{22}^j and CV_{22}^j, that is, *K*=2 and *N*=2, as shown in Figure 5, for CM_{33}^j and CV_{33}^j, that is, *K*=3 and *N*=3, as shown in Figure 6 , and CM_{44}^j e CV_{44}^j,i.e., *K*=4 e *N*=4, as shown in Figure 7. The best results for matrices with *K*=2, *N*=2 , *K*=3 and *N*=3 were generated by polynomial function of order 3. However, the best results for matrices with *K*=4 e *N*=4 were generated by *Kernelcachelimit* function, because as each class is represented by 16 points and there are 10 classes to be classified (separated), there are 160 points separated and the *Polynomial* function obeys an order P as shown in Table I and $1 \leq P \leq 3$, $P \in Z$ resulting in a very limited hyperplane relative to the curvature that the function line can make with a limit P equal to 3. The *Kernelcachelimit* function provides a value that specifies the size of the cache memory of the kernel matrix, while the algorithm maintains a matrix with up to 5000×5000 of double precision floating-point numbers in memory.

In Bresolin [20], the use of SVM with wavelet digital voice recognition in Brazilian Portuguese, obtained an average of 97.76% using 26 MFCC's in the pre-processing of voice and SVM machine's with the following characteristics: MLP as Kernel functions, ten machines (one for each class) and "one vs. all" as method of multiple classes. In comparison to this work, the results of this remain more effective, because the amount of MFCC's is smaller and, also, the input of parameters in the machines are lower. Consequently, the computational load is lower.

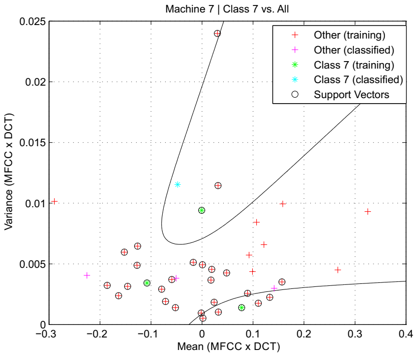


Figure 5. Machine generated for class 7 from matrices CM_{22}^7 and CV_{22}^7.

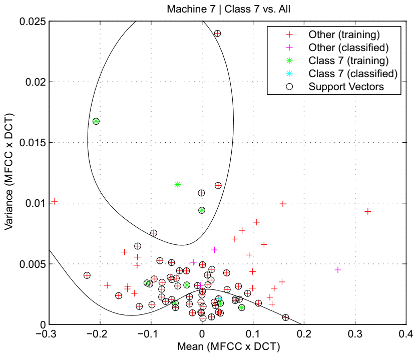


Figure 6. Machine generated for class 7 from matrices CM_{33}^7 and CV_{33}^7.

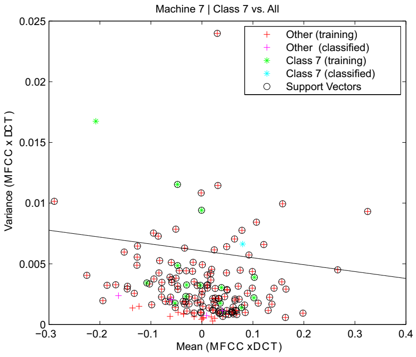


Figure 7. Machine generated for class 7 from matrices CM_{44}^7 and CV_{44}^7.

B. Test

With the result of the best function from training, the tests were made from voice banks where the speakers are independent and classified with the best function of training: *Polynomial* of order 3, except the matrices with *K*=4 e *N*=4 were tested (classified) with the same function of the training: *Kernelcachelimit*. The speakers 1 and 2 are male and the speaker 3 is female. The Tables II, III and IV show the rates of successes.

TABLE II. TEST PERFORMED FROM MATRICES CM_{22}^{j} AND CV_{22}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	10	10	10	10
Class 1	10	10	7	10
Class 2	8	5	7	5
Class 3	8	5	5	5
Class 4	10	5	5	5
Class 5	10	3	5	7
Class 6	8	5	7	5
Class 7	10	7	5	5
Class 8	10	10	10	10
Class 9	10	3	0	0
TOTAL	94	63	61	62

TABLE III. TEST PERFORMED FROM MATRICES CM_{33}^{j} AND CV_{33}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	10	9	8	9
Class 1	10	10	9	10
Class 2	8	8	7	7
Class 3	10	9	7	10
Class 4	10	3	4	2
Class 5	10	3	4	3
Class 6	10	6	7	7
Class 7	8	9	8	9
Class 8	10	9	10	7
Class 9	10	6	7	7
TOTAL	96	72	71	71

TABLE IV. TEST PERFORMED FROM MATRICES CM_{44}^{j} AND CV_{44}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	8	8	8	8
Class 1	10	10	10	10
Class 2	10	8	9	8
Class 3	10	9	7	8
Class 4	10	6	7	8
Class 5	10	8	9	7
Class 6	8	8	7	8
Class 7	10	10	10	10
Class 8	10	10	10	10
Class 9	10	6	6	5
TOTAL	96	82	83	82

IV. CONCLUSION

Analysing the methodology and applications of SVM, one realises that it is a technique with excellent response time of computational execution. Despite being a dichotomic method of classification, this also has possible means to work with a larger number of classes of different data types to be separated. In the standards classification proposed in this work, the SVM presented problems to correctly classify points very close among to each other, because of the form generalization of one versus all. However, as it has a very wide scope in relation to the classification functions during the learning process of the machines, the SVM ends up compensating for the problem of generalization with the use of more points for classification. That is, the greater the number of points to represent the class the higher the amount of hits. In general, the patterns were classified very well, except with the digit ‘9’. The digits ‘1’ and ‘8’ obtained the highest classifications. The use of mean and variance chosen as characteristics of the data to be generated patterns was the most appropriate way to find a better separability between points and therefore a better classification.

ACKNOWLEDGMENT

The authors thank the Scientific Initiation Program of the Federal Institute of Maranhao for financial support through grant aid, and availability of Digital Systems Laboratory, and the Research Group for Electronic Instrumentation Technology Applied to the IFMA.

REFERENCES

[1] P. Fantinato, Segmentacao de Voz baseada na Analise Fractal e na Transformada Wavelet. Prentice Hall, Outubro 2008.

[2] L. Rabiner and R. Schafer, Digital Processing of Speech Signals. Prentice Hall, 1978.

[3] S. Haykin, Redes Neurais:Principio e pratica. Bookman, 2002.

[4] A. Bresolin, Reconhecimento de voz atraves de unidades menores do que a palavra, utilizando Wavelet Packet e SVM, em uma nova Estrutura Hierarquica de Decisao. Tese de Doutorado, Natal 2008.

[5] C. Ding and I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics, 2001.

[6] R. Urena, A. Moral, C. Moreno, M. Ramon, and F. Maria, “Real-time robust automatic speech recognition using compact support vector machines.” IEEE Transactions on Audio, Speech, and Language Processing, May 2012, pp. 1347–1362.

[7] V. Vapnik and A. Chervonenkis, On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Dokl, 1968.

[8] B. Scholkopf, O. Simard, A. Smola, and V. Vapnik, Prior knowledge in support vector kernels. The MIT Press, 1998.

[9] P. C. Clarkson and P. Moreno, “Acoustics, speech and signal processing.” IEEE International Conference, March 1999, pp. 585–588.

[10] J. Mercer, Functions of positive and negative type, and their connections with theory of integral equations. Transactions of the London Philosophical Society, 1909.

[11] C. De-Gang, Y. Heng, and E. Tsang, Generalized Mercer theorem and its application to feature space related to indefinite kernels. International Conference Machine Learning and Cybernetics, 2008.

[12] D. Hanchate, M. Nalawade, M. Pawar, V. Pohale, and P. Maurya, “vocal digit recognition using artificial neural network”.” 2nd International Conference on Coumputer Engineering and Technology, April 2010, pp. 88–91.

[13] R. Aggarwal and M. Dave, “application of genetically optimized neural networks for hindi speech recognition system”.” World Congress on Information and Communication Technologies (WICT), December 2011, pp. 512–517.

[14] S. Azam, Z. Mansor, M. Mughal, and S. Moshin, “urdu spoken digits recognition using classifield mfcc and backpropagation neural network”.” 4th International Conference on Computer Graphics, Imaging and Visualization (CGIV), August 2007, pp. 414–418.

[15] M. Mohammed, E. Bijov, C. Xavier, A. Yasif, and V. Supriya, “robust automatic speech recognition systems:hmm vesus sparse”.” Third International Conference on Intelligent Systems modelling and Simulation, February 2012, pp. 339–342.

[16] C. Ganesh, H. Kumar, and P. Vanathi, “performance analysis of hybrid robust automatic speech recognition system”.” IEEE International Conference on Signal Processing, Computing and Control (ISPCC), March 2012, pp. 1–4.

[17] J. Tamgo, E. Barnard, C. Lishou, and M. Richome, “wolof speech recognition model of digits and limited-vocabulary based on hmm and toolkit”.” 14th International Conference on Computer Modelling and Simulation (UKSim), March 2012, pp. 389–395.

[18] J. Picone, “Signal modeling techniques in speech recognition.” IEEE Transactions on Computer, April 1991, pp. 1215–1247.

[19] P. Fissore and E. Rivera, “Using word temporal structure in hmm speech recongnition.” ICASSP 97, April 1997, pp. 975–978.

[20] A. Brasolin, A. Neto, and P. Alsin, “”digit recognition using wavelet and svm in brazilian portuguese”.” ICASSP 2008, April 2008, pp. 1–4.

Searching Source Code Using Code Patterns

Ken Nakayama
Tsuda College
2-1-1 Tsuda-machi, Kodaira-shi, Tokyo, Japan
e-mail: ken@tsuda.ac.jp

Eko Sakai
Otani University
Koyama-Kamifusacho, Kita-ku, Kyoto, Japan
e-mail: echo@res.otani.ac.jp

***Abstract*— Understanding source code is crucial for a software engineer. To efficiently grasp the semantics of source code, an experienced engineer recognizes semantic chunks and relations (code patterns) in source code as clues. If a rich repository of searchable code patterns, together with human understandable meanings, is available, comprehension of unfamiliar source code would be easier. However, explicitly defining a source code query even for a simple code pattern can be prohibitively complex for human. In this paper, a tool for search-by-example through abstract syntax tree is presented. A programmer gives sets of desired and undesired nodes, then the system presents some candidate nodes resembling desired ones. This kind of implicit definition by examples is suitable for constructing and revising a repository socially. The method is supervised incremental learning of decision trees. The proposed system uses a set of primitive attributes to reflect domain-specific knowledge safely and easily.**

Keywords—source code search; search by example; abstract syntax tree

I. Introduction

Observing common programming habit, an engineer seems to locate and loosely combine apparent semantic chunks of various granularity to understand source code. We call such an apparent semantic chunk a code pattern. Although formal correctness of this coarse understanding is not guaranteed, this often successfully outlines the structure of the source code, and thus it becomes good guidance to the engineer. Rich knowledge of code patterns is one of the primary sources of the strength of an experienced software engineer.

If code patterns in source code can be explicitly given by the author or other person who understands the code before, or automatically searched by a tool, understanding unfamiliar source code would be easier. Unfortunately, neither of them is realized. As for the former, comment lines embedded in source code are too ambiguous for automatic processing. As for the latter, defining syntactic search pattern for a code pattern would be too complicated for human, since a code pattern may appear with many minor variations.

We believe that software engineering community should share and socially evolve the "dictionary for code understanding" in the form of queries through source code. This is like a searchable code snippet. Contributors are expected to add and refine these searchable snippets together with description in natural language and example codes.

In this paper, a define-by-example framework for a code pattern is presented based on [1]. In the framework, each instance of a code pattern is represented as an anchored abstract syntax tree (AST). An anchored AST is a tuple

$$(t_1, t_2, \cdots, t_n; T) \qquad (1)$$

where $(t_1, t_2, \cdots, t_n)$ is a tuple of AST nodes, or anchors, and T is the enclosing AST, say a `class`. A tuple of anchors provides a syntactically clear (i.e., tightly coupled with source code) chunk for semantic annotaion, while T indicates the maximum extent (context) of the interest in which possible relationships among $(t_1, t_2, \cdots, t_n)$ are searched for. The number of anchors n may be arbitrarily chosen by a user. A prototype system is presented, too.

In the next section, related work is presented. In Section III, a motivating example of the code pattern is presented. In Section IV, anchored abstract syntax tree is propsed as a means of representing instances of a code pattern, then a system which infers a candidate definition of a code pattern from user specified code pattern examples is presented. Section V outlines the algorithm for inferring a code pattern definition. Some experience with our prototype system is in Section VI, the conclusion in Section VII.

II. Related Work

Comprehension of computer programs by human programmer is one of the most important activities in software engineering. There are a wide variety of researches in the literature. Dynamic analyses [2] try to understand runnable programs, while static approaches analyse source codes without running them. The essential difficulty of program comprehension exists in the broad gap of complexity between human and software. When coding a program, abstraction mechanisms such as libraries, code fragments (code snippets), or design pattern help a programmer. However, when reading a code in practice, a programmer can rely only on comments in source code and documents. Feature (concept) location systems may suggest possible locations of interest in source code, but a programmer still have to read the code line by line. Design pattern detection systems may report the meaning of the located code, but their targets are small number of well-known design patterns.

When searching through source code, conventional string search tools are not suitable, because they do not assume the syntax of a programming language. Some tools search source code based on its structure to overcome this limitation. For example, `sourcerer`[3] is a search engine for open source code. `sourcerer` allows a user to ask about implementation and program structure. They compare some ranking methods for the search result ordering.

A brief history of software reuse is presented in [4]. Signature matching [5] tries to search a function using types of its parameters and the return value as a query. There are extensions, such as, [6] using formal semantics, [7] using a specification language, [8] using contract-based specification matching, or [9] using a type system for specification description. All of these approaches are based on "static" information which is extracted from source code or specification without running it.

Another way of charactrising a function is specifying a set of expected (input, output) pairs as a query. Each candidate function is run against these inputs to check whether the output becomes the same with the specified ones. This is "dynamic" characteristics of a function. Researches based on this method include [10], [11], and [12]. Generally, however, signature or type matching alone seldom gives satisfactory search result.

While above approaches aim at reusing functions or methods, other techniques such as [13] or [14] are intended for reusing code fragments.

III. Code Pattern

As an example of code pattern, we will compare some implementations of memoization in Java language. Memoization [15] is a technique to avoid redundant computation of a referentially transparent function, at the cost of storing and recalling already computed results. Referential transparency requires a function to always return the same result for the same given arguments. Memoization is effective when computationally expensive function is called repeatedly. In the design pattern, the flyweight pattern is essentially similar with this technique.

The code in Fig. 1 illustrates a typical memoization. The method `get(P p)` returns the value of function `calc(P p)` either by actually calculating it or by recalling it from already calculated table. For this purpose, this method has a table `Map<P, V> values` which is used to keep already calculated values. If `get(P p)` is called with `p` for the first time, the value `calc(P p)` is added into the table with `p` as its key. If `get(P p)` is called again, this method returns the result by retrieving it from the table without calculating it again.

Although memoization is a quite simple technique, it may be implemented in various ways like in Fig. 2. Some characterizations of memoization should be given to a system, but it is practically impossible for a user to define a search pattern which matches all of these code patterns selectively.

```
abstract class Memoize<P, V> {

    Map<P, V> values = new HashMap<P, V>();

    public V get(P p) {
        if (! values.containsKey(p)) {
            values.put(p, calc(p));
        }
        return values.get(p);
    }

    public abstract V calc(P p);

}
```

Fig. 1: Example code of memoization in Java language.

```
if (! values.containsKey(p)) {      // Code A
  values.put(p, calc(p));
}
return values.get(p);
```

```
if (tab.containsKey(this.d))        // Code B
    return tab.get(this.d);
else {
    Node newNode = new Node(this.d,
            new P(this.d - 1, this.t),
            new P(this.d + 1, this.t));
    tab.put(this.d, newNode);
    return newNode;
}
```

```
if (memo.get(n) == null) {          // Code C
    memo.put(n,
            memoizedFibonacci(n - 1).
            add(memoizedFibonacci(n - 2)));
}
return memo.get(n);
```

```
Byte cb = (Byte) cache.get(nInt);   // Code D
if (cb == null) {
    byte b = runAlgorithm(n);
    cache.put(nInt, new Byte(b));
    return b;
} else {
    return cb.byteValue();
}
```

```
if (r[x][z] == -1) {                // Code E
    r[x][z] = compute(x, z);
}
return r[x][z];
```

Fig. 2: Examples of the memoization code pattern.

To cope with this difficulty, we have adopted a decompose-and-conquer strategy. Let us focus on the conditional expression of `if` statement as an example. The followings are from Fig. 2 and other codes:

```
if (! values.containsKey(p)) {...
if (tab.containsKey(this.d)) ...
if (built.containsKey(m)) ...
if (value == null) {...
if (memo.get(n) == null) {...
if (solved.get(newVal) != null) {...
if (cb == null) {...
if (rem[x][z] == -1) {...
if (c[i][j] == -1) {...
```

Some common patterns are observed. If the table is implemented as a `Collection` object, a predicate method `containsKey()` is used. If the result type is `int`, the value `-1` seems to represent the absence of the data in the table, while `null` is used for reference types (e.g., objects) after trying to get the value using `get()`.

We expect the code pattern would be something like the followings:

```
_X_.containsKey(_Y_)
_X_.get(_Y_) == null
_X_ = _Y_.get(_Z_); ...; _X_ == null
_X_[_Y_][_Z_] == -1
```

allowing `boolean` negation such as `!` or `!=`. Note that this pseudo notation is only for explanation here.

```
IfStatement     // root ASTNode of this part
  // conditional expression of the if-statement
  // ASTNode for condition expression
  EXPRESSION
    PrefixExpression
      > (Expression) type binding: boolean
      OPERATOR: '!'
      OPERAND
        MethodInvocation
          > (Expression) type binding: boolean
          > method binding:
                  Map<P,V>.containsKey(Object)
          EXPRESSION
            SimpleName
              > (Expression) type binding:
                      java.util.Map<P,V>
              > variable binding:
                      Memoize<P,V>.values
              IDENTIFIER: 'values'
          TYPE_ARGUMENTS (0)
          NAME
            SimpleName
              > (Expression) type binding:
                      boolean
              > method binding:
                  Map<P,V>.containsKey(Object)
              IDENTIFIER: 'containsKey'
          ARGUMENTS (1)
            SimpleName
              > (Expression) type binding: P
              > variable binding: p
              IDENTIFIER: 'p'
  THEN_STATEMENT        // then-part
    Block               // then-part is a block
      STATEMENTS (1)
  ELSE_STATEMENT: null  // no else-part
```

Fig. 3: AST for `if` statement of Fig. 1 (generated using [17] and adapted).

Once this code patter is defined, this can be used as a building block of other code patterns. Of course, these syntactic characteristics do not guarantee that this conditional expression is a constituent element of memoization. Although they just suggest it, the existence of multiple clues gradually increases the confidence of it.

IV. Code Pattern by Example: User Interaction

A. Code Pattern as Anchored AST

Source code that conforms to the syntax specification of the language can be represented as AST. Fig. 3 is an AST which corresponds to the `if` statement in Fig. 1. This AST is generated by Java development tools (JDT)[16] in Eclipse IDE. The root node `IfStatement` represents the whole `if` statement, which has three branches, namely, `EXPRESSION` (condition expression), `THEN_STATEMENT` (then-part), and `ELSE_STATEMENT` (else-part). In this example, `ELSE_STATEMENT` is `null` since the `if` statement has no else-part. Fig. 3 shows just an overview the AST for simplicity.

We represent an instance of a code pattern as an anchored AST. If a user wants to express a code pattern for the conditional expression in the above example, anchored AST's $(t_1; T)$ where t_1 is conditional expression such as

```
  values.containsKey(p)
```

or

```
  cb == null.
```

As for the latter one, if a user wants to give explicit hint for the constraint on variable `cb`, constraining AST node

```
  Byte cb = (Byte) cache.get(nInt);
```

may be specified as an additional anchor t_2, resulting an anchored AST $(t_1, t_2; T)$.

B. User Interaction Model

A user interactively defines a new code pattern by giving positive or negative examples of a code pattern. Each example is an anchored AST. Negative example refers to an anchored AST which is not an instance of the code pattern. Typically, a negative example is obtained when the system mistakenly locate a false positive code pattern. In such a case, a user teaches that it is a negative example.

We will give the overview of this user interaction using our prototype system. The prototype system has been implemented as a plug-in of Eclipse IDE [18]. A plug-in can take advantage of functions for manipulating Java source code provided by Eclipse JDT [16]. Fig. 4 is the screen shot of the prototype system. In the window, a special pane for anchored AST registration and classification is placed on the right-hand-side.

Part (a) through (d) in Fig. 5 are screen snapshots when a user is defining an single anchored AST $(t_1; T)$. A user selects a code region corresponding to the intended AST node (part (a)), then add this either as a positive (part (b)) or as a negative (part (c)) example by clicking appropriate button. Added AST node is displayed in the pane with a character '`P`' or '`N`' indicating positive or negative, respectively.

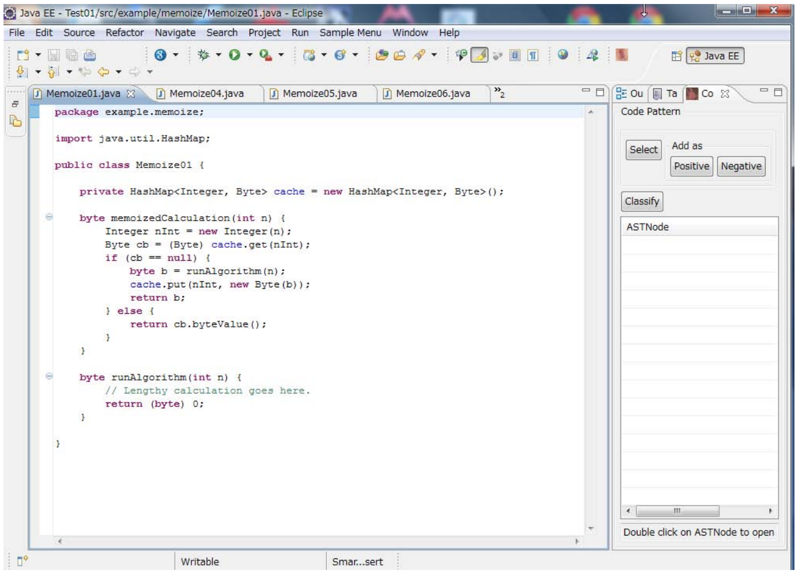
For each code pattern, the system keeps positive example set S_+ and negative example set S_-. A user can add arbitrary number of positive examples to S_+. Once a user thinks that he has put enough positive/negative examples, the system infers a hypothetical definition of the intended code pattern from S_+ and S_-, the system infers the definition of a code pattern as a supervised discrimination learning on AST.

Then, the sytem searches through source code for a tuple of AST nodes that match the hypothetical definition. If such a tuple is found, it is presented to the user. The user judges whether the found tuple is an instance of the intended code pattern or not, and add it as either positive or negative example to the system. Updated S_+ and S_- are used for the next search. A user continues this interaction until he is satisfied with the inferred code pattern.

V. Classifying Attribute Vectors

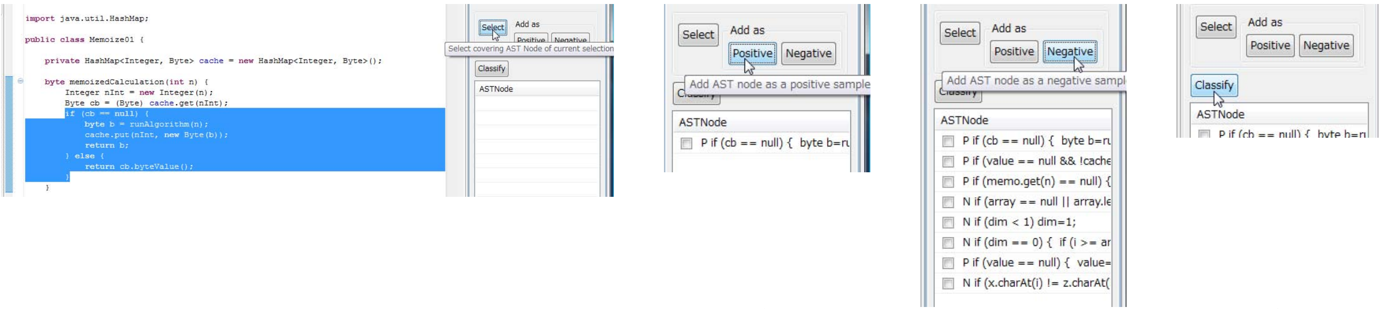
A. Projecting Anchored AST onto Attribute Vector

The problem to solve is finding a common pattern in S_+ and not in S_-. Since anchored AST has complex structure to compare directly, we have decided to project an anchored AST onto a vector of attributes. Let us illustrate an attribute vector with a simple example. Suppose that we are interested in a


Java EE - Test01/src/example/memoize/Memoize01.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Sample Menu Window Help
Java EE
Memoize01.java
Memoize04.java
Memoize05.java
Memoize06.java
Ou
Ta
Co
Code Pattern
Select
Add as
Positive
Negative
Classify
ASTNode
Double click on ASTNode to open
Writable
Smar...sert

```
package example.memoize;

import java.util.HashMap;

public class Memoize01 {

    private HashMap<Integer, Byte> cache = new HashMap<Integer, Byte>();

    byte memoizedCalculation(int n) {
        Integer nInt = new Integer(n);
        Byte cb = (Byte) cache.get(nInt);
        if (cb == null) {
            byte b = runAlgorithm(n);
            cache.put(nInt, new Byte(b));
            return b;
        } else {
            return cb.byteValue();
        }
    }

    byte runAlgorithm(int n) {
        // Lengthy calculation goes here.
        return (byte) 0;
    }

}
```

Fig. 4: A screen shot of the prototype system.



Select covering AST Node of current selection

Add AST node as a positive sample

Add AST node as a negative sampl

□ P if (cb == null) { byte b=ru
□ P if (value == null && !cache
□ P if (memo.get(n) == null) {
□ N if (array == null || array.le
□ N if (dim < 1) dim=1;
□ N if (dim == 0) { if (i >= ar
□ P if (value == null) { value=
□ N if (x.charAt(i) != z.charAt(

(a) (b) (c) (d)

Fig. 5: (a) Select an AST node on source code, add it as a (b) positive or (c) negative example, repeat this to register some examples, then (d) classfy to compose a decision tree.

code pattern represented as anchored AST's with two anchors. That is,

$$S_+,\ S_- \subseteq U^{(2)},\ S_+ \cap S_+ = \emptyset \qquad (2)$$

where $U^{(2)} = \{(t_1, t_2; T)\}$ denotes the set of all anchored AST's with two anchors. And suppose that we have choosen a vector of four attribute functions $A = (a_1, a_2, a_3, a_4)$ where

$$a_i : U^{(2)} \mapsto \text{(various attribute value)}\ (i = 1, \cdots, 4) \qquad (3)$$

as shown in Table I. Then, if t_1 is the `if`-statement

TABLE I: ATTRIBUTE FUNCTIONS (EXAMPLE).

$a_1(t_1, t_2; T)$	`true` if t_1 is an `if`-statement; `false` otherwise
$a_2(t_1, t_2; T)$	`true` if t_2 is an `if`-statement; `false` otherwise
$a_3(t_1, t_2; T)$	Method name if t_2 is a method invocation; `null` otherwise
$a_4(t_1, t_2; T)$	`true` if t_2 is under t_1 in AST T; `false` otherwise

```
if (! values.containsKey(p)) {
    values.put(p, calc(p));
}
```

and t_2 is "`put(p, calc(p));`" in t_1, $A(t_1, t_2; T)$ would be an attribute vector $(\texttt{true}, \texttt{false}, \texttt{"put"}, \texttt{true})$.

B. Generating Attribute Vector

An attribute function vector A is generated by recursively combining predefined primitive attribute functions. Table II is an excerpt of the list of primitive attribute functions used in the current prototype system. There are primitive functions too that have multiple inputs not shown in this table.

`ASTNode` is a class type which represents AST node in JDT. Current implementation of primitive attribute function returns `null` when its imput(s) are illegal or nonsense. For instance, `paIfThenStatement` returns the AST node of then-part if the input is `if`-statement and then-part exists, but it returns `null` otherwise.

An attribute function vector A for an anchored AST with one anchor $U^{(1)}$ is shown in Fig. 6. Line `0` is the given AST node t_1. After that line, generated attributes follow. For example, line `1` means that this at-

TABLE II: PRIMITIVE ATTRIBUTE FUNCTIONS (EXAMPLE).

Function name	Input type	Output type
paAstNodeType	(ASTNode)	Class
paIfExpression	(ASTNode)	Expression
paIfThenStatement	(ASTNode)	Statement
paBlockStatements	(ASTNode)	List<Statement>
⋮	⋮	⋮

tribute is defined as application of primitive attribute function `paAstNodeType()` to the value of line `0`, that is t , and the type of value is `java.lang.Class` . Similarly, line `2` uses another primitive attribute function `paIfExpression()` , which returns another AST node of class `org.eclipse.jdt.core.dom.Expression` (subclass of ASTNode). If line `0` is not an instance of `if`-statement, the value would be `null`. The value type of line `2` is obtaind as an attribute at line `11`.

By applying generated attribute function vector A to an anchored AST, an attribute value vector is obtained. Fig. 7 shows an example of attribute value vector. This is logically single vector (wrapped to fit the page width). `NA` denotes the value which is not used for constructing a decision tree. For example, if a value is an instance of `ASTNode`, it is used as an input of other attributes, but is not directly used as an attribute for classification.

C. Discrimination by Constructing Decision Tree

From attribute value vectors projected from S_+ and S_-, a decision tree is constructed for classification. A decision tree is a tree-structured cascade of decisions in which each node represents a decision on a certain attribute at a time. Current prototype system uses J48 in [19]. Fig. 8 is an example of constructed decision tree. `N0`... are nodes, and `N0->N1`... are transitions labeled with a condition. For example, At the root node `N0`, primitive attribute functions `paIfExpression()`, `paInfixExpressionRightOperand()`, and `paAstNodeType()` are applied to the given `ASTNode` (`param0`) in this order. `paIfExpression()` assumes the given `ASTNode` is an `if`-statement, and returns its condition expression. Then, `paInfixExpressionRightOperand()` assumes the condition is infix expression, and returns its right-hand-side operand. Finally, `paAstNodeType()` extracts its node type. Depending on the type, one of `N0->N1`, `N0->N2`, ..., `N0->N5` is chosen. If the decision flow reaches `N1`, that indicates that the given tuple is classified as `positive`.

VI. Experience with Prototype System

A. Searching a Tuple Through Source Code

A decision tree can classify a given tuple of `ASTNode`. Current prototype system does not have efficient search mechanism yet. It uses brute-force exhaustive search by enumerating all combination of `ASTnode`'s in the given source code. This works when applied to small tuple and source code, improvement is necessary.

B. Current Set of Primitive Attribute Functions

The primitive attribute functions currently implemented in our prototype system are in two types, namely, (1) function which gives some value describing a feature of the given `ASTnode`, (2) function which traverses to another `ASTnode` from the given `ASTnode`.

Since most of type (2) follow parent-to-child relationship in AST, structurally similar code pattern can be defined. However, anchored AST's that have different structures to each other are recognized as different ones. For instance, `x = 3;` and `{ x = 3; }` are semantically equivalent, but the existence of an extra block prevents the intended recognition. In the same way, an operator such as equivalence operator == is commutative (if there is no side effect), but current system cannot generalize `x == null` and `null == x`. Rather, the system just enumerate these two patterns as distinct ones.

Another example is `ASTnode` traverse by following data flow or control flow. When the AST node at `if`-statement is given as an anchor in the next code, the variable `cb` should be treated as if it is `cache.get(nInt)`.

```
Byte cb = (Byte) cache.get(nInt);
if (cb == null) {
```

This can be realized if a primitive function exists which traverse AST following data flow backward.

To improve such situations, more sophisticated primitive functions should be introduced.

VII. Conclusion

A framework for incremental definition of a code pattern has been presened. In the framework, a code pattern is represented as an anchored AST by a user. An anchored AST is then projected onto an attribute vector for classification using a decision tree. An attribute function vector is generated by recursively combining primitive attribute functions. A prototype system is implemented as a plug-in of Eclipse IDE. Inferred decision tree if good for exact exact search.

Future improvements include enriching primitive attribute functions, reducing redundant or nonsense attributes.

References

[1] K. Nakayama and E. Sakai, "Source code navigation using code patterns," IEICE technical report. Life intelligence and office information systems, vol. IEICE-113, no. 479, mar 2014, pp. 23–28, (In Japanese).

[2] B. Cornelissen, A. Zaidman, A. Van Deursen, L. Moonen, and R. Koschke, "A systematic survey of program comprehension through dynamic analysis," Software Engineering, IEEE Transactions on, vol. 35, no. 5, 2009, pp. 684–702.

[3] S. Bajracharya, T. Ngo, E. Linstead, Y. Dou, P. Rigor, P. Baldi et al., "Sourcerer: a search engine for open source code supporting structure-based search," in Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications. ACM, 2006, pp. 681–682.

[4] S. P. Reiss, "Semantics-based code search," in Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on. IEEE, 2009, pp. 243–253.

[5] A. M. Zaremski and J. M. Wing, "Signature matching: A key to reuse," vol. 18, no. 5. ACM Press, 1993, pp. 182–190.

[6] E. J. Rollins and J. M. Wing, "Specifications as search keys for software libraries." in ICLP. Citeseer, 1991, pp. 173–187.

```
0: Given parameter: value type class org.eclipse.jdt.core.dom.ASTNode
1: Generated attribute: paAstNodeType(0)-->class java.lang.Class
2: Generated attribute: paIfExpression(0)-->class org.eclipse.jdt.core.dom.Expression
3: Generated attribute: paIfThenStatement(0)-->class org.eclipse.jdt.core.dom.Statement
4: Generated attribute: paIfElseStatement(0)-->class org.eclipse.jdt.core.dom.Statement
5: Generated attribute: paInfixExpressionOperator(0)-->
                    class org.eclipse.jdt.core.dom.InfixExpression$Operator
6: Generated attribute: paInfixExpressionLeftOperand(0)-->class org.eclipse.jdt.core.dom.Expression
7: Generated attribute: paInfixExpressionRightOperand(0)-->class org.eclipse.jdt.core.dom.Expression
8: Generated attribute: paMethodInvocationBoundMethod(0)-->class java.lang.String
9: Generated attribute: paMethodInvocationArguments(0)-->interface java.util.List
10: Generated attribute: paBlockStatements(0)-->interface java.util.List
11: Generated attribute: paAstNodeType(2)-->class java.lang.Class
  (some lines omitted)
20: Generated attribute: paIfExpression(7)-->class org.eclipse.jdt.core.dom.Expression
21: Generated attribute: paIfThenStatement(2)-->class org.eclipse.jdt.core.dom.Statement
  (the rest are omitted)
```

Fig. 6: Combination of primitive attribute functions (example).

```
NA,class org.eclipse.jdt.core.dom.IfStatement,NA,NA,NA,null,null,null,null,null,null,
class org.eclipse.jdt.core.dom.InfixExpression,class org.eclipse.jdt.core.dom.Block,
class org.eclipse.jdt.core.dom.Block,null,null,null,null,null,null,null,null,null,null,null,
null,null,null,null,null,null,"==",null,null,null,null,NA,null,null,null,null,NA,null,null, ...
```

Fig. 7: Attribute values for decision tree construction (example).

```
J48() result:
digraph J48Tree {
N0 [label="paAstNodeType(paInfixExpressionRightOperand(paIfExpression(param0)))" ]
N0->N1 [label="= class org.eclipse.jdt.core.dom.NullLiteral"]
N1 [label="positive (3.0)" ]
N0->N2 [label="= class org.eclipse.jdt.core.dom.NumberLiteral"]
N2 [label="negative (2.0)" ]
N0->N3 [label="= class org.eclipse.jdt.core.dom.MethodInvocation"]
N3 [label="negative (1.0)" ]
N0->N4 [label="= class org.eclipse.jdt.core.dom.InfixExpression"]
N4 [label="negative (1.0)" ]
N0->N5 [label="= class org.eclipse.jdt.core.dom.PrefixExpression"]
N5 [label="positive (1.0)" ]
}
```

Fig. 8: Attribute values for decision tree construction (example).

[7] D. Hemer and P. Lindsay, "Supporting component-based reuse in care," in Australian Computer Science Communications, vol. 24, no. 1. Australian Computer Society, Inc., 2002, pp. 95–104.

[8] J.-J. Jeng and B. H. Cheng, "Specification matching for software reuse: a foundation," in ACM SIGSOFT Software Engineering Notes, vol. 20, no. SI. ACM, 1995, pp. 97–105.

[9] C. Runciman and I. Toyn, "Retrieving re-usable software components by polymorphic type," in Proceedings of the fourth international conference on Functional programming languages and computer architecture. ACM, 1989, pp. 166–173.

[10] S.-C. Chou, J.-Y. Chen, and C.-G. Chung, "A behavior-based classification and retrieval technique for object-oriented specification reuse," Software: Practice and Experience, vol. 26, no. 7, 1996, pp. 815–832.

[11] R. J. Hall, "Generalized behavior-based retrieval," in Proceedings of the 15th international conference on Software Engineering. IEEE Computer Society Press, 1993, pp. 371–380.

[12] S. Thummalapenta and T. Xie, "PARSEWeb: a programmer assistant for reusing open source code on the web," in Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. ACM, 2007, pp. 204–213.

[13] S. Paul and A. Prakash, "A framework for source code search using program patterns," Software Engineering, IEEE Transactions on, vol. 20, no. 6, 1994, pp. 463–475.

[14] G. Little and R. C. Miller, "Keyword programming in java," Automated Software Engineering, vol. 16, no. 1, 2009, pp. 37–71.

[15] D. Michie, "Memo functions and machine learning," Nature, vol. 218, no. 5136, 1968, pp. 19–22.

[16] The Eclipse Foundation, "Eclipse Java development tools (JDT)," http://eclipse.org/jdt/ [retrieved: July, 2014].

[17] ——, "AST View in eclipse Java development tools (JDT)," https://eclipse.org/jdt/ui/astview/index.php [retrieved: July, 2014].

[18] ——, "Eclipse integrated development environment," http://www.eclipse.org/ [retrieved: July, 2014].

[19] Machine Learning Group at the University of Waikato, "Weka 3: Data Mining Software in Java," http://www.cs.waikato.ac.nz/~ml/weka/ [retrieved: July, 2014].

Condition Monitoring of Casting Process using Multivariate Statistical Method

Hocine Bendjama, Kaddour Gherfi, Daoud Idiou
Welding and NDT Research Centre
(CSC)
Algiers, Algeria
e-mails: hocine_bendjama@daad-alumni.de,
gkaddour2@yahoo.fr, ddidiou@yahoo.com

Jürgen Bast
Institut für Maschinenbau
Technische Universität Bergakademie
Freiberg, Germany
e-mail: bast@imb.tu-freiberg.de

***Abstract*—Growing demand for higher performance, safety and reliability of industrial systems has increased the need for condition monitoring and fault diagnosis. A wide variety of techniques were used for process monitoring. This study will mainly investigate a technique based on principal component analysis in order to improve the accuracy for fault diagnosis of casting process. The process faults are identified using the following statistical parameters: Q-statistic, also called squared prediction error, and Q-residual contribution. The proposed method is evaluated using real sensor measurements from a pilot scale. The monitoring results indicate that the principal component analysis method can diagnose the abnormal change in the measured data.**

Keywords-fault diagnosis*; *process monitoring; principal component analysis*; *Q-statistic*; *Q-residual contribution

I. Introduction

The fault detection and diagnosis is an extremely important task in process monitoring. It provides operators with the process operating information, which helps monitor the process and quickly detect and diagnose the fault.

Investment casting process has known a great development due to its wide use in automotive industry. It can be used to create complex castings at a high production rate and low cost. Even in a controlled process, defects in the output can occur.

One critical process step is filling the mold with molten metal. Significant research has been performed to link factors like pouring temperature, metal velocity, sand and refractory coating, to the filling process and defect formation [1][2][3].

Casting defects are often very difficult to characterize. They will fall into one or more of the established seven categories of casting defects: metallic projections, cavities, discontinuities, defective surface, incomplete casting and incorrect dimensions or shape [4][5].

In a controlled process, defects do not just happen, they are caused. If a defect occurs, measures must be adopted to eliminate its cause and prevent its repetition. It is the purpose of this paper to diagnose process faults that can cause casting defects. Casting process fault diagnosis is an important research domain, and gotten large attention by a number of researchers. Several methods have been proposed to identify possible causes for reducing or eliminating casting defects, e.g., Abdelrahman et al. [6] presented a methodology for monitoring the metal filling process. In order to achieve this, a data collector and sensors were designed. An electrostatic simulation package was used to interpret signature obtained from the sensors during the metal filling. An artificial neural network was trained to indicate the metal filling profile based on the results of the electrostatic simulations. The results were verified by comparing the metal filling profile inferred from the neural network to the actual metal filling profile captured by an infrared camera. Similarly, a novel approach based on a fuzzy inference system was applied by Deabes et al. [7] for obtaining the profile of the liquid metal to monitor the filling process. Dobrzański et al. [8] developed a computer code based on the X-ray imaging and the artificial intelligence tools. The proposed method was used to ensure the automatic identification and classification of possible defects in cast aluminum alloys in order to reduce and even eliminate them. To quickly detect process faults, a monitoring method of the metal filling profile was proposed by Okaro et al. [9]. This method makes use of an array of capacitive sensors to detect the position and amount of the molten metal as it displaces into the mold. An iterative algorithm for the estimation of metal filling time was also used to provide a good prediction of the filling time. A recent study [10] has been carried out by Jafari et al. on the effects of some important casting process parameters on the quality and the properties of castings using full-factorial design of experiment. These methods usually adopt measurements as the essential basis and provide aid in early detection and diagnosis of process faults by extracting useful information from measured data.

The accuracy of diagnosing process faults from measured data can be improved using Principal Component Analysis (PCA) method [11]. The PCA is a data compression method; it produces a lower dimensional representation in a way that preserves the correlation structure among the original data. The PCA method has received a great deal of attention in recent years for their ability to successfully determine when a fault has occurred, a large number of applications have been reviewed [12][13][14][15].

This paper presents the PCA method for fault detection and diagnosis with application to low pressure lost foam casting process. The PCA is used to establish the statistical correlation among the measured data to detect and diagnose the abnormal situations and to provide information about the process state by using the statistical parameters; Q-statistic, also called SPE (Squared Prediction Error), and Q-residual contribution. The main goal of this method is to obtain more detailed information contained in the measured data.

The paper is organized as follows. Section II presents a brief overview of the low pressure lost foam casting process and the proposed method for its monitoring. The PCA method and process fault diagnosis using PCA, along with its formulations, are described in Section III. The monitoring results are discussed in Section IV. Finally, Section V concludes our contributions.

II. Materials and Method

In this section, the experimental setup and the proposed method used to monitor the casting process are presented.

A. *Casting Process*

The low pressure lost foam casting process was developed by Lang [16]; it is used to create complex castings. The casting process uses air pressure to push liquid metal up into a flask containing the foam pattern and unbounded sand. Fig. 1 illustrates the schematic of the low pressure lost foam casting process.

The casting machine employs a resistance furnace capable of melting standard aluminium base alloys. The components contacting the liquid metal, like the tube and the adapter, are made of cast iron with a refractory coating. A thin sheet of aluminium foil is used to protect the foam from thermal radiation of the liquid metal before the beginning of the mold filling process. Air pressure is applied to the chamber containing the crucible to raise the liquid metal into the mold.

B. *Data Acquisition*

The foam pattern was supported in the flask. The thermocouples were wired to the data acquisition unit. When the vessel is pressurized, the liquid metal rises through a steel pipe into the flask. All test parts were cast using AlSi12 alloy at temperatures between 730°C and 750°C, as presented in Table I.

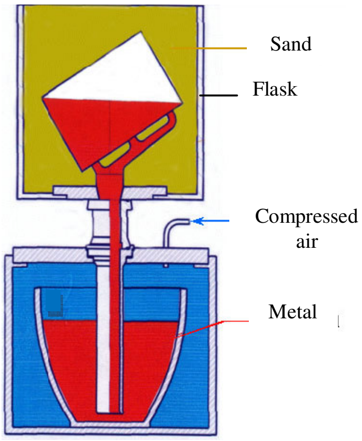


Figure 1. Schematic of low pressure lost foam casting process.

TABLE I. Measurement Conditions

Test n°	Pouring Temperat-ure (°C)	measured Temperatu-re (°C)	Holding Pressure (bar)	Mold Filling time (s)	Holding time (s)
1	735	711	0.24	6	90
2	750	705	0.24	6	90
3	750	700	0.24	6	90
4	730	711	0.24	6	90

Five temperature transducers were used to acquire data by five thermocouples for temperature input. These sensors were implemented in the process. Both the pressure and temperature inputs were wired to a National Instruments data acquisition board. Other signals are also included as well as the ability to drive outputs as needed. National Instruments DASY Lab software was used to collect and analyze the signals from the temperature sensors. The measured variables are listed in Table II and presented in Fig. 2.

TABLE II. Process Variables

Variables	Description	unit
T	Temperature	°C
P	Pressure	Bar
S	Rise (height of filling)	M
T1	Temperature 1	°C
T2	Temperature 2	°C
T3	Temperature 3	°C
T4	Temperature 4	°C
T5	Temperature 5	°C

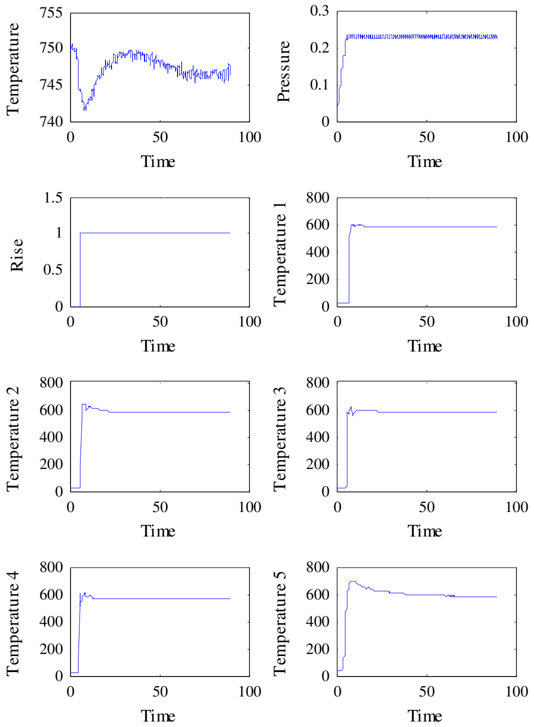


Figure 2. Measures of the process variables.

ISBN: 978-1-61208-354-4

C. Method

Casting defects will generally fall into one or more of the established seven categories of defects. Generally, a casting defect is defined as all observable and unplanned variation. When defects exist, the possible causes can be examined and the corrective action can be taken.

In the casting industry, there is little and inconsistent data about the conditions that cause casting defects. There is a temptation to attempt to diagnose a process fault by the possible causes. The proper identification of a fault is required to correct and control the quality of castings.

The requirements of productivity and quality impose the application of advanced monitoring methods. In this work, the PCA method is used for casting process monitoring with surface defect. The PCA model is trained with input matrix *X* that contains the eight variables presented in Table II; the matrix $X \in R^{m \times 8}$ represents *m* observations or samples of these variables.

In the next section, the PCA algorithm that is in charge of the identification of abnormal situations in the behavior of the process is presented.

III. Principal Component Analysis

The PCA [11][17] is a multivariate analysis technique and also a dimension reduction technique. It reduces the dimensionality of the original data by projecting the data set onto a subspace of lower dimensionality including a series of new variables to protect the main original data information.

For a given data matrix $X \in \Re^{m \times n}$, which contains *m* observations and *n* variables, the PCA actually relies on eigenvalue/eigenvector decomposition of the covariance or correlation matrix *C* given by:

$$C = \frac{1}{n-1} X^T X = V D V^T \tag{1}$$

where $D = diag(\lambda_1 \ldots . \lambda_n)$ is a diagonal matrix with diagonal elements in decreasing magnitude order and *V* contains the eigenvectors.

The PCA determines an optimal linear transformation of the data matrix *X* in terms of capturing the variation in the data as follows:

$$T = XP \tag{2}$$

$$\hat{X} = TP^T \tag{3}$$

where *T* is the principal component matrix and the matrix *P* contains the principal vectors which are the eigenvectors associated to the eigenvalues λ_i of the covariance matrix.

The difference between *X* and $\hat{X}$ is the residual matrix *E* (4). This residual captures the variations in the observation space, and it is the basis for fault detection and diagnosis.

$$E = X - \hat{X} = X(I - PP^T) \tag{4}$$

where *I* is the unit matrix.

The identification of the PCA model thus consists in estimating its parameters by an eigenvalue/eigenvector decomposition of the matrix *C*, and determining the number of Principal Components (PCs) *k* to retain. A key issue to develop a PCA model is to choose the adequate number of PCs. Many procedures have been proposed for selecting the number of the PCs to be retained [18]. In this paper, the experiential method [19] is used, which judges that the cumulative sum contribution of the anterior *k* PCs is higher than 0.85, as follows:

$$100 \times \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} > 85\% \tag{5}$$

where *k* is the index of the PCs, *n* is the number of process variables and λ_i is the eigenvalue.

A. Fault Detection and Diagnosis

The PCA is used to establish the normal statistical correlation among the coefficients of the multivariate process data. To perform process fault detection, a PCA model of the normal operating conditions must be built. When a new observation data is subject to faults, these new data can be compared to the PCA model. The correlation of the new data is detected by Q-statistic:

$$Q - statistic = SPE = e^T e = (x - \hat{x})^T (x - \hat{x}) \tag{6}$$

The process is considered normal if

$$Q - statistic \le \delta_Q^2 \tag{7}$$

where δ_Q^2 denote the confidence limit or threshold. It can be calculated from its approximate distribution [20]:

$$\delta_Q^2 = \theta_1 \left[\frac{C_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{8}$$

where $\theta_i = \sum_{j=k+1}^{n} \lambda_j^i \quad i = 1,2,3$ and $h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}$

where C_α is the critical value of the normal distribution.

The δ_Q^2 is used to determine whether the data is within range of the model. To compare the test set to the model using the Q-statistic, a plot of test data must be created with a confidence limit. A confidence limit of α=95% is used throughout this work. Any point below the confidence line is considered normal variance from the selected number of PCs, and any point above this line is considered to have an abnormally high level of variance.

In order to diagnose the process fault, a contribution plot is necessary. The contribution plots are bar graphs of the Q-residual contribution of each variable calculated as in (9) [21]. Variables having the largest residuals produce the worst compliance to the PCA model, and indicate the source of the fault.

$$Q - contribution = cont_i = \frac{\|e_i\|^2}{Q - statistic} \qquad (9)$$

where e_i presents the i^{th} element of the residual vector e and $cont_i$ is the contribution of the i^{th} variable to the total sum of variations in the residual space.

IV. Monitoring Results

The PCA algorithm implies two parts: the first one is the development and training of the PCA model, the second is the test of the process fault based on the trained model. The process data used in training represent the measurements in normal operation conditions.

The sets of representative normal and fault process data are gained through the experimental measurements, including two normal data sets: test1 and test2, and two fault data sets: test3 and test4. Each data set includes eight measurement variables. The sampling interval is 0.1s and data length is 900 observations or samples for each data set.

The eight process variables are used as input to the PCA algorithm. In total, 900 data points at different times were collected for training the PCA model. The variables are of different units, so the data are scaled to zero mean and unit variance.

The eigenvalues of the covariance matrix, which are the variances of PCs, are listed in Table III. Through the PCA, the anterior 2 principal components' accumulation sum contribution rate is 92.96%. As shown in Table III, the best monitoring performance is achieved when two PCs are used. The PCA model is established by using them, and then the fault detection with the process is progressed.

TABLE III. PCs Values

Variables	Eigenvalues	Variance (%)
1	**6.4323**	**80,40**
2	**1.0049**	**12,56**
3	0.2905	03,63
4	0.1222	01,53
5	0.0729	0,91
6	0.0415	0,52
7	0.0262	0,33
8	0.0095	0,12

The results in the training phase are shown in Fig. 3. The detection threshold is calculated according to (8), which is 1.5192; this is also shown in this figure with a dashed red line. To evaluate fault detection method, the detection ratio is used. It is defined as the number of samples whose Q-statistic values go beyond the threshold to the total number of samples. When the detection ratio is less than 20%, the faults are not detected successfully [21]. As shown in Fig. 3, only 13.66% of the total samples were above the threshold value. It implies that the model has captured the major correlation and variance among the process variables.

During the testing phase, the new data sets can be compared to the PCA model and its threshold. These new data has been scaled to zero mean and unit variance of the model. The fault detection results of the test data sets including test3 and test4 are presented in Figs. 4 and 5. As illustrated in these figures, all samples of Q-statistic violated the threshold. The model has not captured the majority of the variance; therefore, the PCA model does not describe the data adequately, the data are considered faulty.

After the fault is detected, the diagnosis is determined by the contribution plot. The bar graph of each variable is presented in Figs. 6 and 7. The process fault is produced through the 8th variable (temperature 5) for the test3 and 4th variable (temperature 1) for the test4.

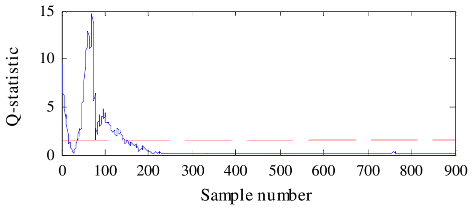


Figure 3. Q-statistic of training data (test2).

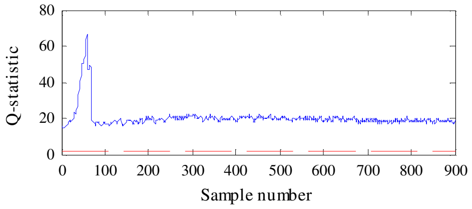


Figure 4. Q-statistic of test3.

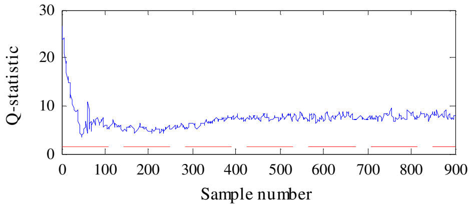


Figure 5. Q-statistic of test4.

ISBN: 978-1-61208-354-4

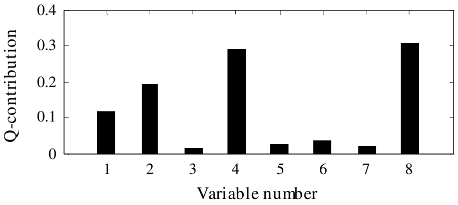


Figure 6. Q-contribution of test3.

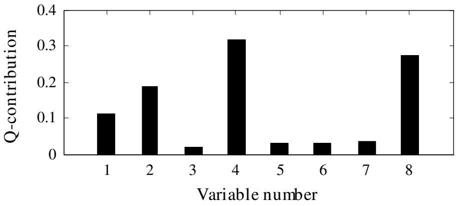


Figure 7. Q-contribution of test4.

The obtained results allow us to identify the pouring temperature as the main cause for the occurrence of the surface defect. Another possibility is that a surface defect may be formed because the thermocouple has to be embedded into the foam.

V. Conclusion and Future Work

In this paper, the PCA method is applied to improve the performance of casting process monitoring by using the statistical parameters; Q-statistic and Q-residual contribution. The aim of this application is to detect casting defects occurring at different stages of the process, and also to identify their causes. The obtained results demonstrate that the normal running state and the state with fault of the process can be clearly identified; the fault can be given by using the proposed method.

The PCA method used in this work is accurate in fault detection and diagnosis of low pressure lost foam casting process. The operator can combine the results obtained by the multivariate statistical analysis with the process knowledge, and easily find out the reasons that arouse the faults. The future work will be focused on the application of condition monitoring on other types of casting defects.

References

[1] C. E. Bates et al. “Advanced lost foam casting technology,” Summary report DOE, AFS, no. UAB-MTG-EPC95SUM, 1995.

[2] D. R. Hess, “Comparison of Aluminum alloys and EPS foams for use in the lost foam casting process,” Transactions of the American Foundry Society, vol. 112, 2004, pp. 1161-1174.

[3] T. Pacyniak, “The effect of refractory coating permeability on the lost foam process,” Archives of Foundry Engineering, vol. 8, no. 3, 2008, pp. 199-204.

[4] G. Hénon, C. Mascré, and G. Blanc, Search for quality castings, Technical Editions of Foundry Industries, Paris, 1986.

[5] P. Beeley, Foundry technology, Oxford: Butterworth Heinemann, 2001.

[6] M. Abdelrahman, J. P. Arulanantham, R. Dinwiddie, G. Walford, and F. Vondra, “Monitoring metal-fill in a lost foam casting process,” ISA Transactions, vol. 45, no. 4, 2006, pp. 459-475.

[7] W. A. Deabes, M. A. Abdelrahman, and P. K. Rajan, “A fuzzy-based reconstruction algorithm for estimating metal fill profile in lost foam casting,” Proc. American Control Conference, Seattle, USA, 2008, pp. 4868-4874, doi:10.1109/ACC.2008.4587265.

[8] L. A. Dobrzański, M. Krupiński, J. H. Sokolowski, P. Zarychta, and A. Włodarczyk-Fligier, “Methodology of analysis of casting defects,” Journal of Achievements in Materials and Manufacturing Engineering, vol. 18, no. 1-2, 2006, pp. 267-270.

[9] M. Okaro, M. Abdelrahman, and J. Graves, “Monitoring Metal Fill Profile in Lost Foam Casting Process Using Capacitive Sensors and Metal Fill Time Estimation,” Proc. IEEE Sensors Applications Symp, 2011, pp. 76-81, doi: 10.1109/SAS.2011.5739811.

[10] H. Jafari, M. H. Idris, and A. Shayganpour, “Evaluation of significant manufacturing parameters in lost foam casting of thin-wall Al–Si–Cu alloy using full factorial design of experiment,” Transaction of Nonferrous Metals Society of China, vol. 23, 2013, pp. 2843-2851.

[11] L. H. Chiang, E.L. Russell, and R.D. Braatz, Fault Detection and Diagnosis in Industrial Systems, London: Springer-Verlag, 2001.

[12] M. J. Zuo, J. Lin, and X. Fan, “Feature separation using ICA for a one-dimensional times series and its application in fault detection,” Journal of sound and vibration, vol. 287, 2005, pp. 614-624.

[13] X. Sun, H. J. Marquez, and T. Chen, “An improved PCA method with application to boiler leak detection,” ISA Transactions, vol. 44, 2005, pp. 379-397.

[14] M. Tamura and S. Tsujita, “A study on the number of principal components and sensitivity of fault detection using PCA,” Computers and Chemical Engineering, vol. 31, 2007, pp. 1035-1046.

[15] C. Y. Tsai and C. C. Chiu, “An efficient conserved region detection method for multiple protein sequences using principal component analysis and wavelet transform,” Pattern Recognition Letters, vol. 29, 2008, pp. 616-628.

[16] L. Lang, Development and Testing of a Low Pressure Lost Foam Casting Machine and an Examination of the Process, PhD thesis, Freiberg University, Germany, 1999.

[17] I. T. Jolliffe, Principal Component Analysis, 2nd ed., New York: Springer-Verlag. 2002.

[18] M. Kano and S. Hasebe, “A new multivariate statistical process monitoring method using principal component analysis,” Computers & Chemical Engineering, vol. 25, 2001, pp. 1103-1113.

[19] P. Nomikos and J. F. MacGregor, “Multivariate SPC charts for monitoring batch processes,” Technometrics, vol. 37, no. 1, 1995, pp. 41-59.

[20] J. E Jackson and G. S Mudholkar, “Control Procedures for residuals associated with principal components analysis,” Technometrics, vol. 21, 1979, pp. 341-349.

[21] X. Xu, F. Xiao, and S. Wang, “Enhanced chiller sensor fault detection, diagnosis and estimation using wavelet analysis and principal component analysis methods,” Applied Thermal Engineering, vol. 28, 2008, pp. 226-237.

Fragment-Based Computational Protein Structure Prediction

Nashat Mansour, Meghrig Terzian
Department of Computer Science and Mathematics
Lebanese American University, Lebanon
e-mail: nmansour@lau.edu.lb, meghrig.terzian@lau.edu

***Abstract*—Proteins consist of sequences of amino acids that fold into 3-dimensional structures. The 3-dimensional configuration determines a protein's function. Hence, it is very important to determine the correct structure in order to identify the wrong folding that indicates a disease situation. Computational protein structure prediction methods have been proposed in order to alleviate the enormous time taken by wet-lab methods. This paper presents a fragment-based protein tertiary structure prediction method which employs the CHARMM36 energy model. The method is based on a two-phase Scatter Search algorithm that minimizes the energy function. Backbone fragments are extracted from the Robetta server and side chains are, extracted from the Dunbrack Library. The results show that the algorithm produces tertiary structures with promising root mean square deviations.**

Keywords-protein structure prediction; scatter search; CHARMM36; protein fragments.

I. INTRODUCTION

Proteins are macromolecules found in all biological organisms. They are composed of a sequence of amino acids and are involved in a wide variety of functions within cells including cell structure, cell motility, cell signaling, enzyme catalysis, and substance transport. For example, enzymatic proteins, such as pepsin, are fundamental for the metabolism and accelerate the rates of biochemical reactions. The various functions are determined by the 3-dimensional folding that is based on the unique sequence of amino acids.

Predicting protein tertiary structure provides information about the functionality, localization and interactions between proteins and consequently contributes in drug design and disease prevention associated with protein misfold. The laboratory experimental methods for protein structure prediction, mainly X-ray crystallography and nuclear magnetic resonance, consume a lot time and are error-prone. Hence, computational methods may offer an alternative. Computational approaches for protein structure prediction lie in two groups. The first group, comparative modeling, predicts structures using proteins of known structures as templates [1]-[4]. The second, *ab initio*, predicts structures using the amino acid sequence of the structure to be predicted [5]-[7].

Ab initio approaches are based on Anfinsen's theory stating that the lowest energy value protein conformation is the most stable one [8]. *Ab initio* methods are divided into two classes. The first is fragment-based and the second biophysics-based. Fragment-based methods employ database information, whereas biophysics-based methods do not [5]. A typical *ab initio* method starts with random conformations, generates substitute conformations using heuristics, calculates their energies, and keeps on generating substitute conformations until the ending criterion is reached, where the solution is the conformation with the lowest energy. The efficiency of *ab initio* methods depends on the utilized energy function accuracy and the search algorithm efficiency.

The protein structure prediction problem is NP-Complete. Hence, there is a need for heuristic methods. The main challenge of structure prediction methods is the search space vastness. To limit the search space, a number of models, such as the Hydrophobic-Polar model [9], UNRES model [10], and dihedral angles model [6] have been developed. But, limiting the search space by simplifying the structure model may limit the quality of the predicted protein structure.

Atom-based *ab initio* methods either use fragment databases or are pure. Pure ab initio methods do not employ any prior information. Examples of such published pure *ab initio* work are based on scatter search algorithm [11] and a genetic algorithm [12].

Fragment-based protein structure prediction methods employ peptide fragments, secondary structures and statistical information from the Protein Data Bank (PDB) structures to predict protein tertiary structures. The basic principle behind this method is the presence of a strong relationship between an amino acid sequence and structure [4]. A typical *ab initio* fragment-based method starts with generating fragments from the PDB. Then, heuristics are used to optimize conformations and generate native like structures by using energy functions and evaluation methods.

Iterative Threading Assembly Refinement (I-TASSER) is a unified meta server for protein structure and function prediction [13]. Fragments are utilized to assemble well-aligned structural regions of the segments with unaligned regions. Starting from the query sequence, I-TASSER uses Basic Local Alignment Search Tool (BLAST) to identify sequence homologs. Then, the homologs are aligned using multiple sequence alignment to form a sequence profile and are utilized for predicting secondary structures which are threaded by gathering top template hits from ten threading programs.

The University College London (UCL) bioinformatics group developed several algorithms to tackle protein structure prediction and function annotation including

Fragment-based protein folding (FRAGFOLD) for prediction of tertiary structure [14]. FRAGFOLD starts the folding simulation with supersecondary fragment selection for each position in the query sequence. The energy function utilized includes terms for short-range, long-range, solvation, steric clashes, and hydrogen bonds with their corresponding weights. The energy minimization phase is conducted using a Simulated Annealing approach [15].

ROSETTA [16], an integrated package for protein structure prediction and functional design, is one of the leading *ab initio* performers in Critical Assessment of Protein Structure Prediction (CASP). ROSETTA uses fragments to model the protein backbone. Then, the model is refined and rotamers from the Dunbrack library are assembled to model the side chains. The fragment assembly phase is guided by Monte Carlo Simulated Annealing search [17]. Two energy functions are used in ROSETTA; the probability values used in the energy function are collected using Bayesian statistics from the PDB [17].

This work presents a fragment-based protein tertiary structure prediction method that yields good suboptimal structures. The method employs the CHARMM36 energy model [18] and is based on designing a two-phase scatter search metaheuristic that minimizes the energy function. Backbone fragments are extracted from the Robetta server and, later, side chains are extracted from the Dunbrack Library. The results of applying our method to three proteins are assessed by calculating their energy and root mean square deviation (RMSD) values and by visualizing them. The best structures generated are compared with structures generated by ROSETTA, I-TASSER, and previous work performed by Mansour et al. [19]. The adapted scatter search algorithm yields promising results.

The paper is organized as follows. Section 2 presents a protein structure model, the energy function used, and the assumptions made. Section 3 explains the design of the proposed scatter search algorithm. Section 4 discusses the experiments performed and the results obtained. Section 5 concludes the paper.

II. Protein Model and Energy Function

In the dihedral angles model of protein presentation, the backbone conformation is determined by three torsion angles, Phi φ, Psi ψ and Omega ω, and the conformation of side chains is determined by the Chi χ angles. Phi is formed by the C-N-Cα and N-Cα-C planes and rotating around the N-Cα bond. Psi is formed by the N-Cα-C and Cα-C-N planes and rotating around the Cα-C bond. Omega is formed by the Cα-C-N and C-N-Cα planes and rotating around the C-N bond.

The All-atom Chemistry at HARvard Macromolecular Mechanics (CHARMM36) protein force field function computes the potential energy of a protein structure. The potential energy is the sum of individual terms representing the internal and non-bonded contributions. Internal terms include bond, angle, Urey-Bradley, improper torsion, torsion, and backbone torsional correction energy values. The non-bonded terms include electrostatic, Van der Waals, and solvation values. The following equation represents the nine terms of the CHARMM36 energy function E as a function of the conformation c [18][20].

$$E(c) = \sum_{bonds} K_b (b - b_o)^2 + \sum_{angles} K_\theta (\theta - \theta_o)^2 + \sum_{impropers} K_{imp} (\varphi - \varphi_o)^2 +$$

$$\sum_{torsions} K_x (1 + \cos(n\chi - \delta)) + \sum_{solvation} \sigma_i A_i + \sum_{electrostatic} \frac{q_i q_j}{r_{ij}} +$$

$$\sum_{vanderWaals} \varepsilon_{ij} \left(\left(\frac{R\min_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R\min_{ij}}{r_{ij}} \right)^{6} \right) + \sum_{Urey-Bradley} Ku(u - u_o)^2 + \sum_{\alpha carbons} CMAP(\phi, \psi)$$

Assumptions have been made in this work to simplify the representation of a solution including: constant bond lengths and bond angles; and insignificant improper torsion, Urey-Bradley, and CMAP components. Also, Hydrogen atoms are combined with neighboring heavy atoms referred to as the "extended-atom representation", which reduces the size of the problem.

III. Scatter Search Basic Algorithm

Scatter Search (SS) is a population based, evolutionary and stochastic meta-heuristic that generates and maintains high quality solutions by controlling the search space through randomization, recombination and diversification [21]. Scatter Search generates a random set of candidate solutions, improves them and selects 20% of these solutions and places them in the reference set. Half of the selected solutions are high quality and the other half diverse. Then, it iterates through a subset generation, solution combination, improvement and reference set update methods, where new subsets are generated, combined, improved and included in the reference set according to a certain criteria. In the following sections, we describe the design of the various methods that adapt scatter search to provide good suboptimal solutions for the protein structure prediction problem.

A. *Solution Encoding*

A PROTEIN candidate solution is represented as a list of consecutive objects, AMINO ACIDs. The position of an Amino Acid in a PROTEIN object list is consistent with its position in the protein chain. Consequently, the size of the PROTEIN object is equal to the number of amino acids of the protein. Each AMINO ACID consists of a name, Phi, Psi, omega, Chi1 to Chi4 angle values (if present), van der Waals, electrostatic, torsion, and ASA energy values, and a list of ATOM objects representing the atoms of that particular AMINO ACID. An ATOM has a name and a POSITION object, representing the Cartesian coordinates of that atom.

B. *Diversification Generation Method*

The Diversification Generation Method (DGM) generates random, diverse and valid initial solutions. These solutions are formed by randomly selecting a nine width window of consecutive amino acids from the protein chain, then randomly selecting a 9 width fragment (containing phi, psi and omega values) for this particular position and placing the torsion angles in their corresponding spots. Next, the Cartesian coordinates of the atoms of each amino acid are calculated and the energy of the solution is computed. These steps are repeated until the chain is full and the generated structure is valid, that is, each amino acid in the chain has phi, psi and omega values and there are no collisions between the atoms.

C. Improvement Method

The Improvement Method (IM) enhances the solutions generated by the DGM. After saving the existing values of the torsion angles, for every amino acid position in the chain a fragment is randomly selected for that position and torsion angle values are inserted into the solution. If the newly generated solution is feasible and its potential energy value is lower than the old solution, the move is accepted. The improvement method is run 25 times the protein size. Then, the same procedure is repeated with length 3 fragments, with number of moves being 50 * protein size.

D. Reference Set Update Method

The Reference Set Update Method (RSUM) constructs two reference sets (RefSet), high-quality and diverse solutions. The RefSet b contains b1 high-quality solutions, and b2 diverse solutions. Since b=20% of the population's size and PopSize=100, RefSet has 20 solutions. The b1 solutions are the top 10 minimum energy valued solutions generated from IM. The b2 solutions are the solutions having diverse energy values from the b1 high-quality solutions. After selecting the top 10 solutions of minimum energy and placing them in the RefSet (HQRefSet), for every solution not in the HQRefSet, the minimum distance between this solution and all solutions in the HQRefSet is computed and sorted in decreasing order of minimum distances. The first b2 (most diverse) solutions having the highest energy values are inserted into the RefSet (DivRefSet). The algorithm terminates when no new solutions are found to be inserted into the RefSet or when the number of added solutions reaches a limit.

E. Subset Generation Method and Solution Combination Methods

In the Subset Generation Method (SGM), subsets of the reference set are generated by using a method that groups every pair of elements in a subset. (b!/2!(b – 2)!) subsets are generated, where b is the size of the RefSet.

Then, the pairs generated by the SGM are combined to generate one candidate solution for each pair. For every amino acid, the dihedral angles from either candidate solution are used and the partial energy function, up to this amino acid, is calculated. The angle values that yield a lower energy value of the structure are chosen to be included in the combined candidate solution.

F. Side chain Assembly

After the termination of phase one, the solution with the lowest Cα-RMSD value in the final Reference Set is chosen to go through the side chain assembly phase. In this phase, fragments from the Dunbrack library are chosen and inserted into the solution. The method utilized is the same method utilized in the Improvement method of the Scatter Search algorithm in phase one, with 100 * protein size attempted moves. In this phase, the energy function includes the energy values produced by the side chain atoms and all-atom RMSD value is calculated.

IV. EXPERIMENTAL RESULTS

A. Fragment-based SS and Mansour et al. Results

In this section, we compare our results to the results generated by the pure ab initio results of Mansour et al. [19]. The generated structures are evaluated by computing the root mean square deviation expressed in Å.

Table 1 tabulates the minimum RMSD values generated by both algorithms for 1CRN, 1ROP and 1UTG proteins. Figures 1-3 display the tertiary structures of the three proteins in their native state (PDB) and generated by the two algorithms. Table I shows that the RMSD for 1CRN dropped from 9.01 Å to 8.05 Å, for 1ROP from 12.14 Å to 5.43 Å, and for 1UTG from 14.78 Å 12.34 Å. This shows that the approach utilized in this study significantly improves the three protein RMSD values. Furthermore, unlike [19], the structures generated, have no discontinuities in them.

TABLE1. FRAGMENT-BASED SS RMSD VALUES

Methodology Proteins	1CRN	1ROP	1UTG
Mansour et al. [19]	9.01 Å	12.14 Å	14.78 Å
Fragment based SS	8.05 Å	5.43 Å	12.34 Å

B. Fragment-based SS, ROSETTA, and I-TASSER Results

In these experiments, we compare the generated structures from our algorithm with those generated by I-TASSER and ROSETTA. Since I-TASSER and ROSETTA do not set the first amino acid coordinates of the structures to the coordinates of the corresponding PDB protein, after generating the structures from their servers we translated the coordinates to calculate the RMSDs and to visualize them.

As shown in Table II, the RMSD results generated by our code are the lowest for the three proteins. However, it seems that since RMSD is a global measure, a small disorientation in one part of a protein results in a large root mean square deviation increase. For all the three tested proteins, the visualized generated structures by I-TASEER and ROSETTA looked reasonable. Hence, their RMSDs should have been less. Figure 4 is a case in point. Consequently, the results of our fragment-based SS algorithm can be interpreted as comparable to those of I-TASSER and ROSETTA for these three proteins.

TABLE II. RMSD VALUES GENERATED BY FRAGMENT-BASED SS, I-TASSER AND ROSETTA

Method Proteins	1CRN	1ROP	1UTG
I-TASSER	12.14 Å	26.14 Å	19.94 Å
ROSETTA	11.35 Å	23.28 Å	18.20 Å
Fragment-based SS	8.05 Å	5.43 Å	12.34 Å

V. CONCLUSIONS

In this paper, an *ab initio* fragment-based protein structure prediction method is presented. This method is based on a scatter search metaheuristic. Given a protein sequence and its corresponding fragments, the algorithm first assembles the backbone of the candidate solutions then the side chains of the best generated solution. The RMSD values of the generated structures of three proteins show promising results that are comparable to those of well-recognized algorithms.

Major limitations of this work are presented by the inaccuracy of the energy function and the lack of accuracy of is the dihedral to Cartesian transformation method utilized that is not 100% accurate. Further future work can focus on including more terms in the energy functions. The CMAP term ignored for simplicity, should be added to the energy function. In addition, hydrogen atoms, can be added to the solution representation, thus adding the hydrogen bonding term to the energy function. This would require parallelizing the algorithm in order to speed up the processing and to explore areas of the search space.

ACKNOWLEDGMENT

This work was partially supported by the Lebanese American University and the National Council for Scientific Research.

REFERENCES

[1] M. S. Abual-Rub and R. Abdullah, “A survey of protein fold recognition algorithms,” Journal of Computer Science, vol. 4, no. 9, pp. 768-776, 2008.

[2] L. Chen, G. Liu, Q. Wang, and W. Hou, “Homology modeling of the three-dimensional structure of bovine serum albumin” Proc. 3rd International Conference on Biomedical Engineering and Informatics, Yantai, Oct. 2009, pp. 2377-2381, 2009.

[3] L. Jaroszewski, Z. Li, X. H. Cai, C. Weber, and A. Godzik, “FFAS server: Novel features and applications,” Nucleic Acids Research, vol. 39, pp. 38-44, 2011.

[4] J. Kopp and T. Schwede, “Automated protein structure homology modeling: a progress report,” Pharmacogenomics, vol. 5, no. 4, pp. 405-416, 2004.

[5] C. A. Floudas, “Computational methods in protein structure prediction,” Biotechnology and Bioengineering, vol. 97, no. 2, pp. 207-213, 2007.

[6] P. Bradley, K. M. S. Misura, and D. Baker, “Toward high-resolution de novo structure prediction for small proteins,” Science, vol. 309, pp. 1868-1871, 2005.

[7] F. Liang and W. H. Wong, “Evolutionary Monte Carlo for protein folding simulations,” Journal of Chemical Physics, vol. 115, no. 7, pp. 3374-3381, 2001.

[8] C. B. Anfinsen, “Principles that govern the folding of protein chains,” Science, vol. 181, pp. 223-230, 1973.

[9] N. Mansour, F. Kanj, and H. Khachfe, “Particle swarm optimization approach for protein structure prediction in the 3D HP model,” Interdisciplinary Science: Computational Life Science, vol. 4, pp. 190–200, 2012.

[10] A. Liwo, J. Pillardy, C. Czaplewski, J. Lee, D. R. Ripoll, et al. “UNRES: A united-residue force field for energy-based prediction of protein structure-origin and significance of multibody terms,” Proc. 4th Int. Conf. on Computational Molecular Biology, Tokyo, Japan, April 2000, pp. 193-200.

[11] N. Mansour, C. Kehyayan, and H. Khachfe, “Scatter search algorithm for protein structure prediction,” Int. Journal of Bioinformatics Research and Applications, vol. 5, pp. 501–515, 2009.

[12] S. Schulze-Kremer, “Genetic algorithms and protein folding,” Methods in Molecular Biology, vol. 143, pp.175–222, 2000.

[13] A. Roy, J. Yang, and Y. Zhang, “COFACTOR: An accurate comparative algorithm for structure-based protein function annotation,” Nucleic Acids Research, vol. 40, pp. 471- 477, 2012. Doi:10.1093/nar/gks372.

[14] D. W. Buchan, S. M. Ward, A. E. Lobley, T. C. Nugent, K. Bryson, and D. T. Jones, “Protein annotation and modelling servers at University College London,” Nucleic Acids Research, vol. 38, pp. 563-568, 2010.

[15] D. T. Jones and L. TJ. McGuffin, “Assembling novel protein folds from super-secondary structural fragments,” Proteins, vol. 53(S6), pp. 480-485, 2003.

[16] K. W. Kaufmann, G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, and J. Meiler, “ Practically useful: What the Rosetta protein modeling suite can do for you,” Biochemistry, vol. 49, no. 1, pp. 2987-2998, 2009.

[17] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, “Protein structure prediction using Rosetta,” Methods in Enzymology, vol. 383, pp. 66-93, 2004.

[18] J. Huang and A. D. MacKerell, “CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data,” J. Comput. Chem., vol. 34, pp. 2135–2145, 2013. DOI: 10.1002/jcc.23354.

[19] N. Mansour, I. Ghalayini, M. El-Sibai, and S. Rizk, “Evolutionary algorithm for predicting all-atom protein structure,” Proc. ISCA Third International Conference on Bioinformatics and Computational Biology, New Orleans, Louisiana, March 2011, pp. 7-12.

[20] B. R. Brooks, B.E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, “CHARMM: A program for macromolecular energy, minimization, and dynamics calculations,” Journal of Computational Chemistry, vol. 4, no. 2, pp. 187-217, 1983.

[21] F. Glover, “A template for scatter search and path relinking,” In J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (Eds.), pp. 13-54, 1997, Springer-Verlag.

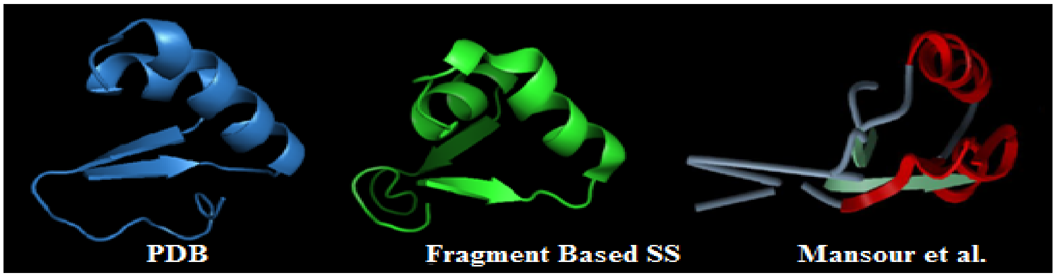


Figure 1. Structures generated by the two methods and the PDB structure for 1CRN.

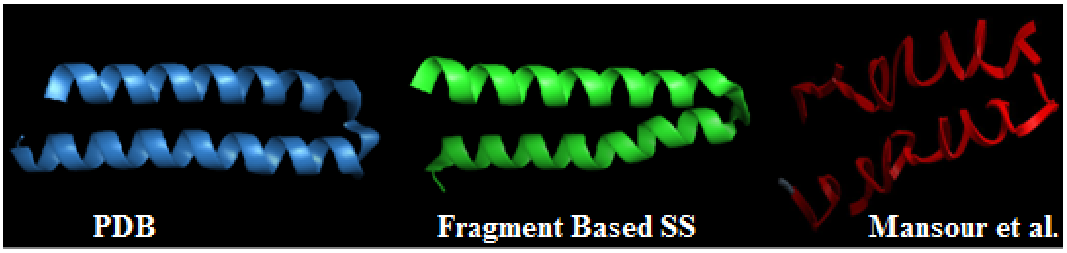


Figure 2. Structures generated by the two methods and the PDB structure for 1ROP.

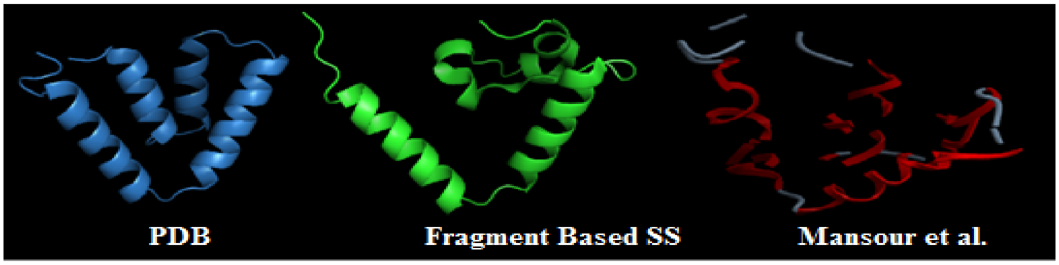


Figure 3. Structures generated by the two methods and the PDB structure for 1UTG.

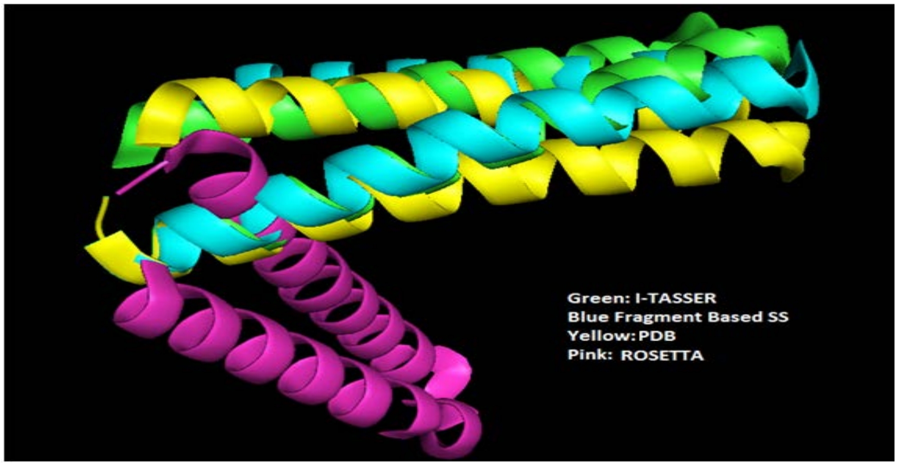


Figure 4. 1ROP Structures Generated.

Visualization of Numerical Information for Construction Project Management using BIM Objects

Leen-Seok Kang
Dept of Civil Engineering
Gyeongsang National University
Jinju, Korea
lskang@gnu.ac.kr

Hyeon-Seong Kim
Dept of Civil Engineering
Gyeongsang National University
Jinju, Korea
wjdchs2003@gmail.com

Chang-Hak Kim
Dept of Civil Engineering
Gyeongsang National University of Science and Technology
Jinju, Korea
Ch-kim@gntech.ac.kr

Young-Hwan Kim
Dept of Civil Engineering
Gyeongsang National University
Jinju, Korea
over2391@naver.com

Soo-Young Yoon
Dept of Civil Engineering
Gyeongsang National University
Jinju, Korea
sionesoju7@naver.com

Bit-Na Cho
Dept of Civil Engineering
Gyeongsang National University
Jinju, Korea
Bey2020@naver.com

***Abstract*— Construction projects are recently growing in size and a wide array of new construction techniques is being introduced, gradually increasing relevant project information. In most construction fields, however, project management operations are undertaken in a way that is heavily dependent upon the experience-based intuition of managers. For those reasons, 4D, 5D, and nD CAD systems are being applied to visualize construction progress information. Augmented Reality (AR) technique is also being an important factor for using Building Information Modeling (BIM) system for construction project. In addition, because the risk analysis information is provided as a mathematical analysis basis, the visualization system of construction risk information is being developed using the BIM tool. This study presents a various application cases of nD CAD objects for the construction project.**

Keywords-nD CAD object; simulation; building information modeling; project management.

I. INTRODUCTION

BIM (Building Information Modeling) based on 3-dimensional information models contribute most noticeably in the visualization of construction information throughout a lifecycle. The chief effects would be interference management at the design phase using 3D blueprint information and schedule management at the construction phase using 4D CAD. The visualization range has included cost and resource information to 4D objects to expand into 5D and 6D CAD systems. Fig. 1 represents a conference method transitioning from numerical information reliance to BIM-based visual information reliance.

The areas of improvement within construction using BIM can be divided into the technical (BIM function improvement) and policy-based aspects (BIM applicability improvement). This paper suggests a direction for function improvement, which requires a distinction between passive and active BIM systems.

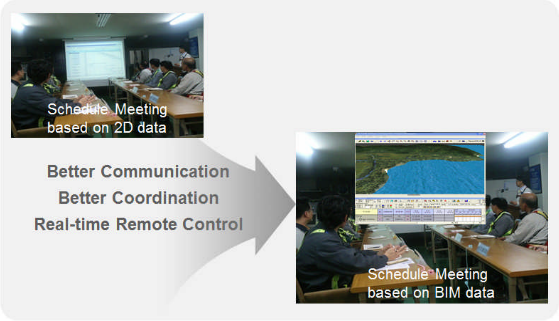

Schedule Meeting based on 2D data
Better Communication
Better Coordination
Real-time Remote Control
Schedule Meeting based on BIM data

Figure 1. Using BIM to visually represent numerical construction information

Passive BIM that is the current BIM system is a tool that expresses the task situation visually. Active BIM goes one step further and contributes analysis that can help offer an optimized solution given task constraints. This paper uses a 4D CAD system, which is used mostly at the construction phase, to present an example of active BIM functions.

ISBN: 978-1-61208-354-4

Recently, there is much research for the visualization of construction information. Most of them are focused on building structures and plant project. Succar [4] suggests that BIM is an expansive knowledge domain within the Architecture, Engineering, Construction and Operations (AECO) industry. Zhang [5] developed an automated safety checking platform using BIM that informs construction engineers what safety measures are needed for preventing fall-related accidents before construction starts. For the risk information analysis, Carr and Tah [1] presented a hierarchical risk classification system as an official model for accurate risk evaluation and a prototype model for risk management system. Zeng et al. [2] proposed new risk analysis methodology based on the fuzzy theory to overcome the failure of conventional risk analysis methods to deal immediately with complex and diverse construction environments. These studies suggest mathematical interpretation-based analysis methods with limited practical applicability. Moon et al. [3] suggested a method to visualize construction risk information using BIM functions.

II. Construction Progress Management with Visual Information

For the 4D system to replace the existing construction management tools, a methodology to create and visualize 3D objects-based on plan versus actual progress is required. Such a progress management function is an essential part of schedule management visualization, and should be applied to 4D system. The system developed by this research team has both a general 4D implementation function and a 4D progress management function that allows analysis of progress state at each point. Fig. 2 expresses visual progress management distinguished by colors, shown in the 4D system.

The progress information visualized this way distinguishes delayed activities as well as activities ahead of schedule by different colors, making it possible for users to establish more advanced schedule management plans when compared to a schedule management tool based on numerical progress information. In Fig. 2, the actual progress versus plan is shown as ahead of schedule, delayed and normal operation, which are marked in colors blue, red and green, respectively. In other words, a methodology is introduced to distinguish ahead-of-schedule areas or delayed areas versus initial plan based on the progress rate calculated by schedule management module with each area shown in different colors, enabling visualization of the current progress state. This visualized progress areas can express detailed information such as ahead of schedule or delay in schedule with numerical information.

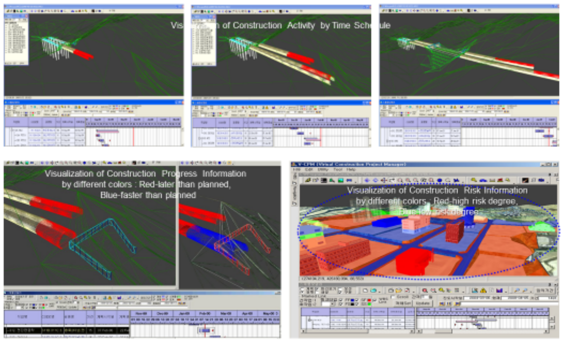

Vis
n of Construction Activity by Time Sch
ule
Visualization of Construction Progress Information
by different colors : Red-later than planned,
Blue-faster than planned
Visualization of Construction Risk Information
by different colors , Red-high risk degree,
Blue-low risk degree

Figure 2. Application of BIM object for construction project

The progress status compared to planned progress is marked with green (ahead of plan), blue (normal progress compared to plan), and red (delayed compared to plan), to allow visual understanding of the overall progress status.

In particular, the progress information visualized this way lets users understand not only progress versus plan but also such specific items as scheduling information, location, and preceding and succeeding activities. Therefore it can be used as effective information when establishing revised plans for delayed activities.

III. 4D Application to Tunnel Construction Project

Of tunnel construction works, long tunnels often exceed 10km in length, and the case to which the system is applied to is a road tunnel construction site whose length is 10.965km long. Therefore, it is bound to limit intuitive management, unlike in the case of buildings, plants and bridge constructions, where users can see relevant structures on a screen. A function would allow management of tunnels by “section” in the process of actual construction, as in Fig. 3.

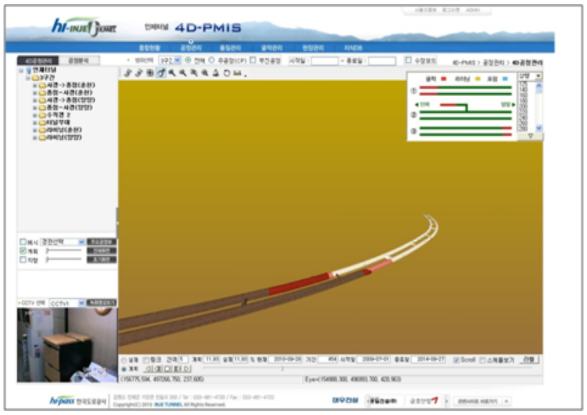


Figure 3. 4D process management function through “section” segmentation at a super long tunnel construction site

While it is possible to see a 3D model of the entire structure in one screen, it will be less efficient when compared to structures and bridge works which allow viewing of entire construction progress at one glance. For 4D process management of civil engineering structures involving a large or long site, effective division of functions into one that manages sections (manageable size) and another that manages the entire work as a whole is a key to enhancing the system's efficiency.

For this particular case, a function utilizing both measurement data and quality test data at a 3D model of the 4D system was developed and applied, in addition to the 4D process management function. Also, video image of CCTV was used in conjunction with 3D to implement the tele-presence functionality.

IV. 4D Application to river facility construction

River facility construction is extensive in nature, so application of a system that combines Google Earth's functionality with 4D for management may be of great use. In this case of application, a system that combines Google Earth with 4D system for management was developed and applied to an actual site (see Fig. 4).

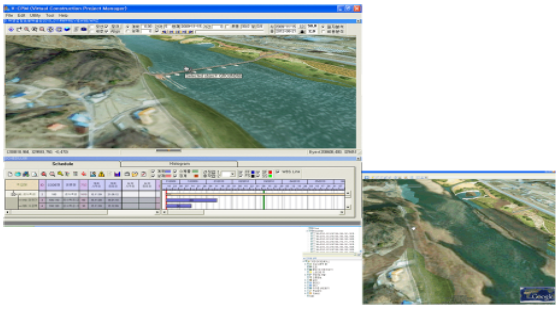


Figure 4. 4D application for river facility project

Fig. 5 shows the process of 4D simulation that includes temporary works that are assembled and dismantled afterwards (coffering works, etc. in this case). In other words, it describes the entire construction process that includes even temporary facilities that are built temporarily to help build permanent structures and then dismantled afterwards, rather than permanent works that are built consecutively and exist continuously.

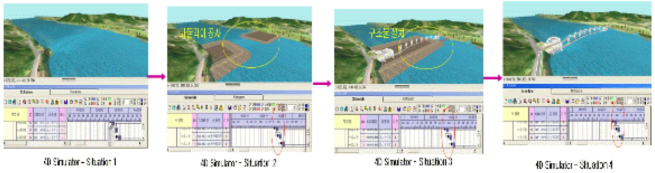

4D Simulator - Situation 1
4D Simulator - Situation 2
4D Simulator - Situation 3
4D Simulator - Situation 4

Figure 5. Simulation of girder structure installation process during river facility construction

In addition, in this case of river facility construction, 4D system was utilized as a way to visualize flood management. A system to visually predict and confirm flood levels based on anticipated precipitation was developed within a 4D system for application; such a technology can be utilized as a useful tool for disaster management during river facility construction projects (see Fig. 6).

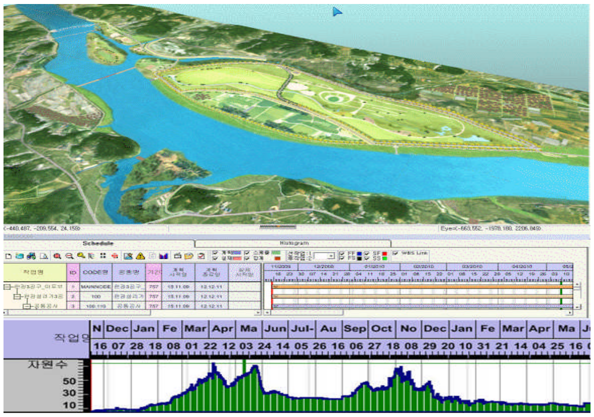


Figure 6. Prediction of flood levels using 4D CAD system

V. Visualization of construction risk information

The analytic hierarchy process (AHP) and fuzzy analysis can be used for the risk analysis of construction information, as described in Fig. 7. First, construction environment factors affecting risk criteria are analyzed and are selected as risk evaluation factors. For example, three risk factors - risk analysis on delay in the construction period (time), risk analysis on greater-than-planned construction budget (cost), and risk analysis on accidents during construction work (work condition)—are calculated in line with AHP analysis procedures. The results are provided for fuzzy analysis procedures to quantify risk criteria in accordance with evaluation factors. The quantitative analysis values of each risk factor are calculated on the basis of fuzzy procedures. For this purpose, linguistic variables are defined as very low (VL), low (L), moderate (M), high (H) and very high (VH). The analysis results of each risk criteria are thus suggested as risk level and comprehensive priorities.

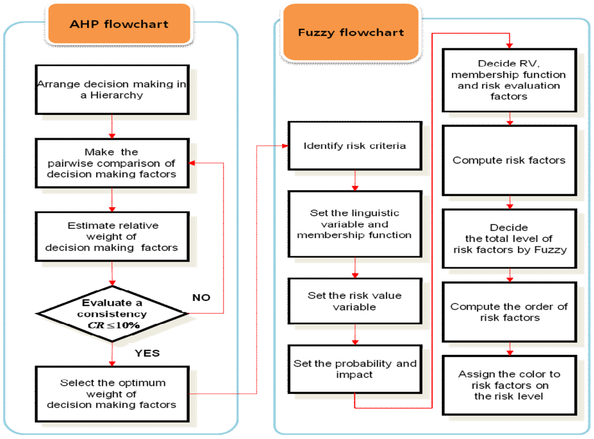

AHP flowchart
Arrange decision making in a Hierarchy
Make the pairwise comparison of decision making factors
Estimate relative weight of decision making factors
Evaluate a consistency $CR \leq 10\%$
NO
YES
Select the optimum weight of decision making factors
Fuzzy flowchart
Identify risk criteria
Set the linguistic variable and membership function
Set the risk value variable
Set the probability and impact
Decide RV, membership function and risk evaluation factors
Compute risk factors
Decide the total level of risk factors by Fuzzy
Compute the order of risk factors
Assign the color to risk factors on the risk level

Figure 7. AHP and Fuzzy analysis Procedures

ISBN: 978-1-61208-354-4

To visualize the process risk, a fuzzy method is used to link to the 4D object. The colors of each process are then changed depending on the level of risk, and the simulation is run for the 4D object. Project managers should be able to recognize which processes will have an especially high amount of risk, increasing their effectiveness. Fig. 8 shows an example of process risk simulation.

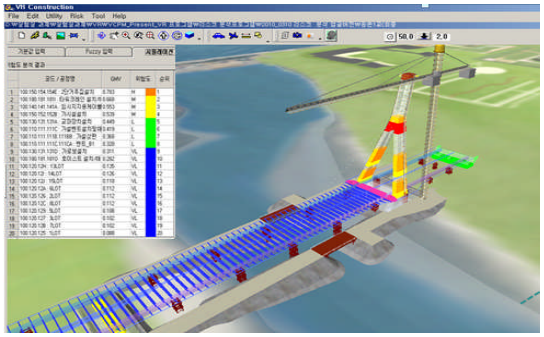


Figure 8. Active 4D CAD system(e.g., representing process risk levels)

The example used in Fig. 8 is a bridge construction project currently underway with 250 separate activities. For risk analysis, each task's possible risk of a safety accident was used as the basis, and further initial information was found through expert interviews and input into the system. Each process was given a probability P and an intensity I by the on-site experts. To objectify the subjective risk levels, a fuzzy analysis was used to set a priority order depending on risk, 5 levels were designated, and the 4D object was simulated. Representing tasks and processes with higher risks visually will allow high risk processes to be managed at a higher level of care by project managers.

VI. Environment Improvement for BIM Application

Fig. 9 outlines the functions required to improve generation convenience in 3D and 4D objects. The first item required is a method to easily transform 2D objects into 3D objects. A possible method may be to create a library for important structure schedules to make schedule information composition of 4D objects a simpler process. The convertibility of 3D objects by parametric variables is also necessary.

In addition, the 3D objects that compose the design phase of construction are mostly single layer objects that are final products of separate complex processes. As a 4D object must be comprised of multiple objects with layers specified to the process level, a simple object disassembly function must be developed. Aside from these former improvements, creating a widely diverse library of 3D objects for vital structural parts is necessary.

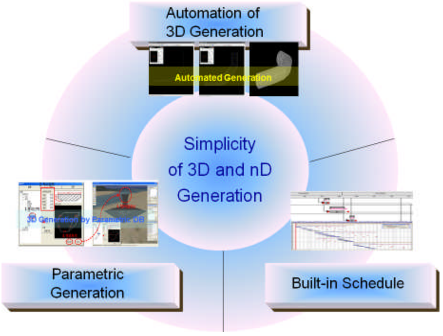


Figure 9. Establishing enhanced convenience in creating 3D and 4D objects

Fig. 10 presents the necessary functions for establishing compatibility with BIM objects including 3D objects. First, the comprehensive completion of IFC codes must advance for open BIM to be created, and developing IFC codes for civil engineering facilities must progress at a faster rate.

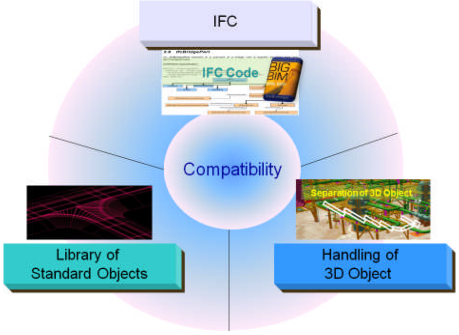


Figure 10. Establishing BIM object compatibility

Fig. 11 details the necessary diversification of specification of BIM tools and evaluation methods. For civil engineering work to move away from the established 2D-based management system to a BIM based system, application guidelines and regulations must be introduced. There is also a need for tools that can evaluate a specific BIM management system's applicability and select specific BIM systems based on the construction project's characteristic properties.

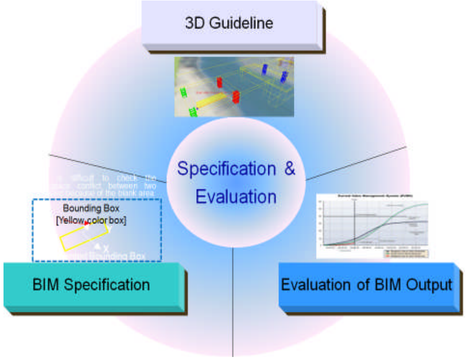


Figure 11. BIM operation system guidelines and product evaluation

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-354-4

VII. CONCLUSION

This research presented a BIM operating process for the construction phase that can raise the usefulness of BIM on civil construction projects. The proposed BIM operating process enables integrated management of construction data through analysis of progress, sequencing errors and risks. Also as the system uses visual representation of complex and numerical information, it can be expected that the system will be actively used as an effective decision making tool.

To validate the system's applicability in real situations, a case study was carried out using bridge, tunnel and railway projects. Because the BIM functions can be used for visualizing the most information through the construction life-cycle, the application of information technology and computer science tools will be gradually increased for the construction project management.

ACKNOWLEDGMENT

This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2011-0016064).

REFERENCES

[1] V. Carr and J. H. Tah, "A fuzzy approach to construction project risk assessment and analysis : construction project risk management system," Advances in Engineering Software, vol. 32, pp. 847-857, October-November 2001.

[2] J. Zeng, M. An, and N. J. Smith, "Application of a fuzzy based decision making methodology to construction project risk assessment," International Journal of Project Management, vol. 25, pp. 589-600, August 2007.

[3] H. S. Moon, H. S. Kim, L. S. Kang and C. H. Kim, "Development of Active BIM Functions for Supporting and Optimized Construction Management in Civil Engineering Projects," The proceedings of ISARC conference (ISARC 2012), June. 2012.

[4] B. Succar, "Building information modelling framework: A research and delivery foundation for industry stakeholders," Automation in Construction, vol. 18, pp. 357-375, May 2009.

[5] S. Zhang, J. Teizer, J. K. Lee, C. M. Eastman and M. Venugopal, "Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules," Automation in Construction, vol. 29, pp. 183-195, January 2013.

Medical Image Retrieval Using Visual and Semantic Features

Supreethi K P
Department of Computer Science and Engineering.
College of Engineering, JNTU
Hyderabad, India
e-mail:supreethi.pujari@gmail.com

Kavitha Pammi,M.Tech(CS)-VI Semester
Department of Computer Science and Engineering.
College of Engineering, JNTU
Hyderabad, India
e-mail:pammikavitha@gmail.com

***Abstract*— Medical images are being digitized and the medical databases are rapidly growing. These images are used in academics, diagnoses, and hospitals for planning treatment. Data mining techniques are applied to medical images, for a quick diagnosis. Thus, the technique of Content-based Medical Image Retrieval (CBMIR) emerges as the times require. For medical image retrieval, current CBMIR is not sufficient to capture the semantic content of an image and difficult to provide good results according to the predefined categories in the medical domain by using less medical knowledge. In this paper, the retrieval system is a combination of low-level image feature and high-level semantics and it includes three main parts: In the first part, the low-level fusion visual features are extracted based on intensity, texture, and their extended versions. Secondly, a set of disjoint semantic tokens with appearance in lung CT images is selected to define a vocabulary based on medical knowledge representation. Finally, a mapping is investigated to associate low-level visual image features with their high-level semantics. In this paper a mapping modelling of visual feature and knowledge representation is presented to approach for medical image retrieval. One important contribution of this paper is the use of physicians defined linguistic variables closely related to known pathologies. This framework could be the foundation of building a novel and flexible model for diagnostic medical image retrieval that uses physician-defined semantics.**

Keywords- Medical image retrieval; low-level features; knowledge representation; semantic Features.

I. INTRODUCTION

With the increasing influence of computer techniques on the medical industry, the production of digitized medical data is also increasing heavily. In recent years, rapid advances in software and hardware in the field of information technology along with a digital imaging revolution in the medical domain facilitate the generation and storage of large collections of images by hospitals and clinics. Though the size of the medical data repository is increasing heavily, it is not being utilized efficiently, apart from just being used once for the specific medical case diagnosis. In such cases, the time spent on the process of analyzing the data is also being utilized for that one case only. But, if the time and data were to be utilized in solving multiple medical cases then, the medical industry can benefit intensively from the medical experts' time in providing new and more effective ways of handling and inventing medical solutions for the future. This can be made possible by combining two most prominent fields in the field of computer science – *data mining techniques* and *image processing techniques.*

Medical imaging is the technique used to create images of the human body for medical procedures (i.e., to reveal, diagnose or examine disease) or for medical science. Medical imaging is often perceived to designate the set of techniques that noninvasively produce images of the internal aspect of the body. Due to increase in efficient medical imaging techniques, there is an incredible increase in the number of medical images. These images, if archived and maintained, would aid the medical industry (doctors and radiologists) in ensuring efficient diagnosis.

The core of the medical data are the digital images, obtained after processing the X-ray medical images; these should be processed in-order to improve their texture and quality using image processing techniques and the data mining techniques may be applied in-order to retrieve the relevant and significant data from the existing million of tons of medical data with the entire manual way to maintain the image data, which is inefficient in meeting the needs of searching with the huge medical image database and is affecting the function of the image used in the diagnoses adversely? Thus, the technique of Content-based Medical Image Retrieval (CBMIR) is considered to be an effect way to tackle the problem. In specialized fields, namely in the medical domain, absolute color or grey level features are often of very limited expressive power unless exact reference points exist as it is the case for computed tomography images [13]. In the medical image system, low-level visual features (e.g., color, texture, shape, edge, etc.) are generated in a vector form and stored to represent the query and target images in the database. Queries by image content require that, prior to storage, images are processed, and appropriate descriptions of their content are extracted and stored in the database [28]. When a user makes a query, medical image retrievals are performed based on computing similarity in the feature space and most similar to the query image are returned to the user based on similarity values computed.

A diagnosis by a specialist often requires a visit to a radiology department to obtain various images that highlight the suspected pathology. Despite the high resolution of the acquired images, image-based diagnosis often utilizes a considerable amount of qualitative measures. To improve the diagnosis and efficiency, the research in medical image analysis has focused on the computation of quantitative measures by automating some of the error-prone and time-consuming tasks, such as segmentation of a structure.

The Bag Of visual Words (BoW) model is commonly used in natural language processing and information retrieval for text documents [1]. In this model, a document is modeled as an instance of a multinomial word distribution and it is represented as a frequency of occurrence word histogram. The representation as a frequency vector of word occurrences does not take grammar rules or word order into account. It preserves key information about the content of the document. This representation can be used to compare documents, and to identify document topics. The BoW representation is successfully used in document classification, clustering, and retrieval tasks and is the cornerstone of all Internet search engines [1].

To represent an image using the BoW model, the image must be treated as a document. Unlike the text world, there is no natural concept for a word or a dictionary. Thus there is a need to find a way to break down the image into a list of visual elements (patches), and a way to differentiate the visual element space, since the number of possible visual elements in an image is very huge. In the visual BoW model, the image feature extraction step takes place in a procedure involving detection of points-of-interest, feature description, and codebook generation. The visual word Model can thus take the form of a histogram representation of the image, based on a collection of its local features. Each bin in the histogram is a codeword index out of a finite vocabulary of visual code words, generated in an unsupervised way from the data. Images are compared and classified based on this discrete and compact histogram representation.

In recent years, the Bow approach has successfully been applied to general scene and object recognition tasks [9] [11] [19]. Varma et al.[19], introduced the idea of using the joint distribution of intensity values over compact neighborhoods for the task of texture classification was introduced. In vector quantization of invariant local image, descriptors were used to form clusters, referred to as visual "words." They then searched for objects throughout a movie sequence by analogy to text retrieval. Natural scene categories were learned using visual words in [11]. Local words were either grayscale patches or scale-invariant feature transform (SIFT) descriptors [26], sampled on a grid, randomly, or at interest points. Then, they then learned a generative hierarchical model to describe the resulting visual word distribution. Spatial pyramids [34] were introduced as a technique of partitioning the image into increasingly fine sub regions, and computing histograms of local features within each sub region.

In this paper , Section II describes the existing systems for medical image retrieval, limitations of the existing system thus motivation and advantages of the proposed system. Section III presents the Architecture of the system and the methods followed to retrieve the result. Section IV depicts the module that is developed to retrieve the result. Section V illustrates the overall work done for implementation and Future work.

II.STATE OF THE ART

In picture archiving and communication system (PACS), image information is retrieved by using limited text keyword in special fields in the image header (e.g., patient identifier). Content-based image retrieval (CBIR) has received significant attention in the literature as a promising technique to facilitate improved image management in PACS system [13][16]. The Image Retrieval for Medical Applications (IRMA) project [16][38] aims to provide visually rich image management through CBIR techniques applied to medical images using intensity distribution and texture measures taken globally over the entire image. This approach permits queries on a heterogeneous image collection and helps in identifying images that are similar with respect to global features e.g., all chest x-rays in the AP (Anterior-Posterior) view. The IRMA system lacks the ability for finding particular pathology that may be localized in particular regions within the image. In contrast, the Spine Pathology and Image Retrieval System (SPIRS) [39][40][41] provides localized vertebral shape-based CBIR methods for pathologically sensitive retrieval of digitized spine x-rays and associated person metadata. Image Map [42] is so far, the only existing medical image retrieval that considers how to handle multiple organs of interest and it is based on spatial similarity. Consequently, a problem caused by user subjectivity is likely to occur, and therefore, the retrieved image will represent an unexpected organ. ASSERT [43] (Automatic Search and Selection Engine with Retrieval Tools) is a content–based retrieval system focusing on the analysis of textures in high resolution Computed Tomography (CT) scan of the lung. In WebMIRS [44] system, the user manipulates GUI tools to create a query such as, "Search for all records for people over the age of 65 who reported chronic back pain. Return the age, race, sex and age at pain onset for these people." In response, the system return values for these four fields of all matching records along with a display of the associated x-ray images. So there is a need of absolute error free, efficient and automatic CBMIR system which can really helpful in medical stream.

Medical images are being digitized and the medical databases are rapidly growing. These images are used in academics, diagnoses, and hospitals for planning treatment. The existing CBMIR [3] systems are capable of retrieving medical images. They take input as an image and produce results that match the low level features of the image. Visual features such as color, shape and texture are implemented for retrieval of images in CBMIR [3].

Limitations / Disadvantages of the existing System

  - The current CBMIR is not sufficient to capture the semantic content of an image [9] because CBMIR is a technique for retrieving image on the basis of automatically derived features such as color, texture and shape to index images with minimal human intervention.
  - Therefore it is difficult to provide good results according to the predefined categories in the medical domain for less using the medical knowledge.
- Accordingly, in this paper, a mapping model of visual feature and knowledge representation is proposed.

- The proposed approach is described in the following section which takes the advantage of semantic feature retrieval along with the visual features of the medical images.

III. PROPOSED SYSTEM

A. System Overview

In this paper, we propose to use medical concepts based on medical knowledge to represent lung CT image. It allows our system to work at a higher semantic level and to standardize the semantic index of medical data, facilitating the communication between visual and textual indexing and Retrieval. Here, a concise presentation of the main theme of this paper is given.

As depicted in Fig. 1, the main components in Essence are: 1) *Semantic domain*; 2) *Images space*; 3) *Feature extraction algorithms*; 4) *Feature domain*; 5) *Query system*;. Knowledge components are represented in rectangles, and knowledge-driven actions, such as search and discovery, are represented in oval shapes.

The *Semantic domain* is organized as a local-as-view data integration subsystem [35]. This system let users build, refine, and further decompose their semantics independently, with minimum effort. The *Semantic domain* represents the expert's knowledge in an XML format. Using a similar format, the framework represents the knowledge of a specific case, a medical image, in *Feature domain.*

Each element in the *Feature domain* is a signature of a medical image in the *Image space*. The signature is computed by executing the *Feature extraction algorithms.*

The *Query system* searches the knowledge base, selects relevant images, and translates the result into a human-readable format. It provides two mechanisms to access the knowledge: 1) query by semantics and 2) mapping low level features with semantic terms.

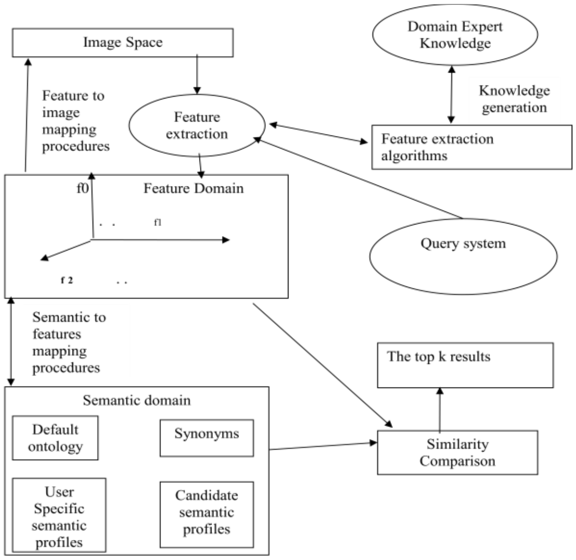


Figure 1. Proposed System Architecture

In the first part, the low-level fusion visual features are extracted based on intensity, texture, and their extended versions. Secondly, a set of disjoint semantic tokens with appearance in lung CT images is selected to define a vocabulary based on medical knowledge representation. Finally, a mapping is investigated to associate low-level visual image features with their high-level semantics.

IV. IMPLEMENTATION

This section describes about the implementation of each module like Pre-Processing, Feature Extraction, mapping algorithm, in which detailed description of each module is give below.

A. Pre-Processing

Pre-processing includes the process of removing the unwanted data from the image and improves the quality of the images. This process of removing unwanted data (like stop-words in the data mining process) can be achieved by the techniques such as *cropping, image enhancement, etc.* In this section, a series of effective pre-processing methods [31] are adopted to extract the pulmonary parenchyma which will improve the quality of feature extraction and then increase the retrieval performance in accuracy and speed. The process of extraction of pulmonary parenchyma is as follow.

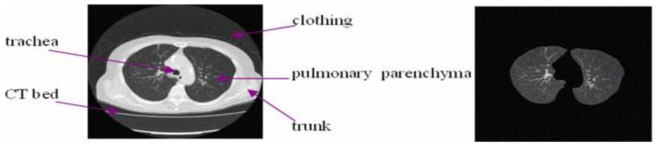


Figure 2. Pre-processing Result

Step 1: Cutting out the background region

- First, pre-processing is applied to the original CT scan image. Both lungs and their nearby portions are areas of interest and pixel values external to this area being insignificant are removed.

Step 2: Segmentation with optimal threshold and noise Cancellation

- The next step is applying threshold to the image to achieve two categories of pixels in the image or a binary image. Then tested for separation of left and right lungs if no threshold value is adjusted.

Step 3: Elimination of trachea and main bronchus.

- During acquisition or digitization process of CT scan images a noise could be introduced that needs to be reduced. An appropriate filter need to be chosen which can enhance the image quality even for non uniform noise, like salt and pepper noise, and also preserve the important edges.
- There may be the presence of noise and other components, i.e., airways and bronchi in the image. These components are to be eradicated.
- It is evident that two major objects in the threshold image are both lungs. Connected component analysis is applied here.
- The connected component labelling algorithm assigns distinct labels to all the regions in the image so as to manipulate the regions fulfilling the specific criteria set for regions. Keeping this in view, extract the two largest components.

Step 4: Adaptive segmentation of left and right lung

- A fully automatic method based on adaptive thresholding for segmenting the lungs in three-dimensional (3-D) pulmonary X ray CT images consists of eight steps.
- In the first step, a threshold is selected to convert a CT image into a binary image.
- In the second step, the lung objects are removed from the ribcage to obtain the external mask.
- In the third step, the right and left lung area is extracted by applying the external mask.
- In the fourth step, the large airways are removed utilizing the mean and the deviation of pixel intensities.
- In the fifth step, a test is made to see if the selected threshold is good. If the selected threshold is not good, the
- Threshold will be adjusted and the algorithm goes back to step 1.
- In the sixth step, a morphology operation is applied to smooth the mediastinum.
- In the seventh step, a split curve is derived from the gap of the separated left and right lungs.
- In the last step, the left and right lungs are segmented.

Step 5: Refinement processing and mask generation

- The external mask is adopted to eliminate unwanted objects surrounding the lungs, so the whole lung region can be extracted precisely.
- To obtain the external mask, remove the two lungs to form the non-lung mask. And then the non-lung mask will be inverted to build the external mask.

Thus, pulmonary parenchyma is useful for the extraction of image feature. One of segmentation result is shown in Fig .2, various features such as trachea, CTbed will be extracted.

B. Low-Level Feature Extraction

Low-level image feature extraction is the basis of CBIR systems. To performance CBIR, image features can be extracted from the entire image.

- *Gray Level Co-Occurrence Matrix (GLCM) Statistical Feature Vector:*

With the created GLCM [36], various features can be computed out. Fourteen parameters were summarized before, but with the special characteristics of lung CT image, four parameters are chosen to descript the texture. These feature description groups along with the images are in a database for the retrieval purpose.

• Energy

Measure the number of repeated pairs. The energy is expected to be high if the occurrence of repeated pixel pairs is high.

$$F_{ENE} = \sum_{j=0}^{N-1} . \sum_{j=0}^{N-1} [p(i.j)|d,\theta)]^2 |(\mathrm{d},\theta)]^2 \qquad (1)$$

• Entropy

Measure the randomness of a gray-level distribution. The entropy is expected to be high if the gray-levels are distributed randomly throughout the image.

$$F_{ENT} = \sum_{i=0}^{N-1} . \sum_{j=0}^{N-1} [p(i.j)|d,\theta)] \log[p(i,j|d,\ \theta)] \quad (2)$$

• Contrast

Measure the local contrast of an image. The contrast is expected to be low if the gray levels of each pixel pair are similar.

$$F_{CON} = \sum_{i=0}^{N-1} . \sum_{j=0}^{N-1} (i-j)^2\, p(i,j|d,\theta)|\mathrm{d},\theta) \quad (3)$$

• Correlation

Provide a correlation between the two pixels in the pixel pair. The correlation is expected to be high if the gray levels of the pixel pairs

are highly correlated.

$$F_{COR} = \frac{\sum_{i=0}^{N-1} \cdot \sum_{j=0}^{N-1} ijp(i,j|d,\theta) - \mu_x \mu_y}{\sigma_x \sigma_y} \qquad (4)$$

Where μ_x , μ_y, σ_x ,σ_y are mean value and standard variance of

$$p_x , p_y , p_x = \sum_{j=0}^{N-1} p(i,j \mid d,\theta) , p_y = \sum_{i=0}^{N-1} p(i,j \mid d,\theta) .$$

So, the above GLCM parameters are used as retrieval feature vector: $F_{GI.CM}$ = {F_{ENG}, F_{ENT}, F_{COR}, F_{LOC}}.And d= 1, $\theta = 0°, 45°, 90°, 135°$.

- *Wavelet Statistical Feature Vector:*

Wavelet transform [37] has been successfully used in image compression, enhancement, analysis, and classification. An image is a 2-D signal, so the 2-D discrete wavelet transform (DWT) can be implemented to approach to texture analysis.

In this paper, a multi-resolution representation is gotten by using 2-D wavelet transform with three-level decomposition. The statistical information, which is from the texture feature with different multi-resolution, will constitute the retrieval feature vector. When the distinct texture characteristics appear in certain frequency and direction the output of wavelet channel has more energy. So the mean and variance of energy distribution in every decomposition level can represent the texture feature. And we use the mean and variance of this energy as the retrieval feature vector F_{WAVI}. .

$$F_{WAVL} = \{E_{10}, E_{11}, E_{12}, E_{20}, E_{21}, E_{22}, E_{30}, E_{31}, E_{32}\} \qquad (5)$$

C. Knowledge Representation and Semantic Features Identification Phase

Most of the decisions in the medical domain are made by comparing the data in hand against existing domain knowledge. During the decision-making process, physicians base their diagnoses on a set of heuristics developed from different areas as a "multi-dimensional intuition" in which tacit knowledge plays a very important role. Several perceptual categories are usually used for recognizing pathologies in lung CT images by physicians.

In this phase, a set of disjoint semantic tokens with appearance in medical images is selected to define a vocabulary based on medical knowledge representation. Here we use the keywords of diagnosis report from the doctor to represent each token in the medical domain.

Semantic vocabulary used

1. Reticular Opacities
2. Nodular Opacities
3. High Density Areas
4. Low Density Areas
5. Cavitary
6. Cystic structure
7. Emphysema
8. Calcification
9. Honeycombing
10. hydrothorax

- *Mapping Algorithm*

The main difficulty in image retrieval based on semantics is to use image's low-level features to replace "word" (semantic) in the text retrieval.

1. A set of disjoint semantic concepts with visual appearance in medical images is first selected to define a vocabulary based on medical knowledge representation.
2. Low-level features are extracted from medical image *z* to represent each vocabulary term.
3. These low-level features are used as training examples to build hierarchical semantic classifiers according to the semantic vocabulary. The classifier for the medical semantic vocabulary is designed using a hierarchical classification scheme based on Support Vector Machine (SVM) classifiers.
4. Hierarchical classification scheme is based on Support Vector Machine (SVM) classifiers. A tree, whose leaves are the medical semantic vocabulary terms is designed and constructed in a top-down manner, guided by the possible hierarchy of the associated terms in semantic vocabulary. Fig. 3 depicts the tree. The upper levels of the tree consist of auxiliary classes that group similar terms with respect to their visual appearances.

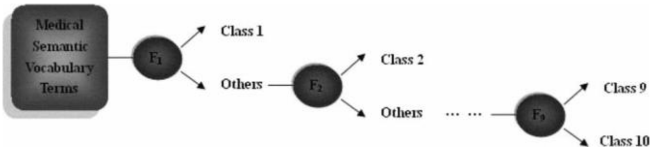


Figure 3. Tree Structure for Medical Semantic Vocabulary Classifier

D. Retrieval Phase

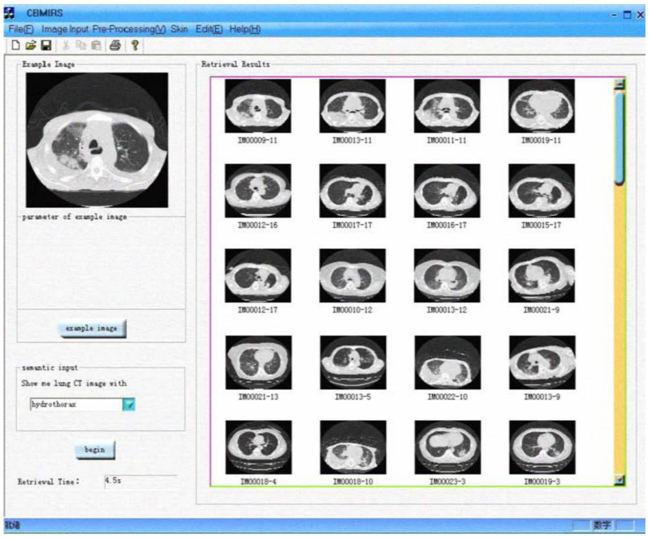
For training of SVM, 100 images with physicians labelling are selected as training set and rest of the images are utilized to test the retrieval approaches. For SVM based image classification, recent work shows that the Radial Basis kernel Function (RBF) works well when the relation between class labels and attributes is nonlinear [33].

Therefore, we use RBF kernel as a reasonable first choice. The RBF kernel function is

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0 \qquad (6)$$

There are two tuneable parameters, while using RBF kernels: c and *r*. We define C =200 and *r* =0.0002 for example.

In this phase, the low level features are used as training examples to build a semantic classifier according to the above vocabulary. The visual feature and semantic feature are mixed as Indexing.



Show me lung CT image with hydrothorax

Retrieval Time: 4.5s

Figure 4. A Retrieval Result

In the experiments, every image from the test DB is served as a query image. We design two types of experiments to evaluate the retrieval system.

I) Retrieval based on low level features only. For every medical semantic category, we chose relevant image as example to conduct the retrieval.

2) Retrieval based on semantic description. For every medical semantic category 10 tests are conducted and 100 tests are conducted totally. One of semantic concept with hydrothorax retrieval results is shown in Fig. 4. Based on the input image as shown in the left top corner in Fig. 4, all the similar images from the database are retrieved.

Our system has a small knowledge base, which can be further enhanced. Ensemble classifiers can be used in the future work. Sophisticated knowledge representation algorithms may be considered.

E. *Evaluation of the Results*

Retrieval precision and ranking measures *(Average-r)* are used as parameters in system evaluation.

(I) The retrieval precision is defined as follows :

Precision = a / a+b (7)

Where *a* is the number of similar images and *b* is the number of dissimilar images in the results .

(2) Suppose the query is *q* , $r_1, r_2, \ldots r_m$ are the correct results retrieved by the system, *rank(r_j)* is the No. *j* correct result's ranking position, so the average ranking value is calculated as follows:

Average-r = 1/m $\sum^{m}$ rank(r_j) (8)

j=1

This value reflects the average ranking of query in the retrieval results. So, the smaller it is, the better. In the experiment, we set m=10. The statistic results are shown in Table 1.

TABLE 1. The Experiment Static Result

	Low-level feature retrieval (%)		Semantic Vocabulary retrieval (%)	
semantic terms	*Precision*	*Average – r*	*Precision*	*Average – r*
Reticular Opacities	55.20	7.55	71.25	5.90
Nodular Opacities	66.50	6.80	66.35	6.55
High Density Areas	52.80	7.85	51.26	8.00
Low Density Areas	46.90	8.40	52.18	7.98
Cavitary	45.36	8.55	40.75	8.45
Cystic structure	49.58	7.95	40.65	8.64
Emphysema	40.20	8.75	45.78	8.30
Calcification	50.13	7.90	60.64	7.04
Honeycombing	60.15	7.00	70.56	6.00
hydrothorax	47.25	8.20	62.12	6.95

From the above statistic results , we can see that the image with outstanding texture information always get better precision and smaller average ranking value. For example the images which possess the pathological characteristics of nodular opacities and honeycombing get the best precision. This is Because the low-level feature extraction procedure is mainly used of texture analysis algorithm.

In addition, when the visual features are difficult to present the method use semantic correlation can put up a satisfied result. For above ten kinds of queries, the semantic correlation gets an average precision. So we can see the method we proposed has a good robustness.

V CONCLUSION AND FUTURE WORK

In this paper, visual, semantic features and knowledge representation are used for medical image retrieval. This framework could be the foundation for building flexible model for diagnosis of medical images. This framework can use physician-needed semantics. The expressions of diseases in medical image are complex and various. From the statistical results, the image with outstanding texture information always gets better precision and smaller average ranking value. The images which possess the pathological characteristics of nodular opacities and honeycombing get the best precision. In addition, when the visual features are difficult to present the method uses semantic correlation, which can put up a satisfied result .Our results prove that our proposed system has good robustness.

REFERENCES

[1] U.Avni, H. Greenspan, Eli Konen, M.Sharon, and J. Goldberger, "X-ray Categorization and Retrieval on the Organ and Pathology Level, Using Patch-Based Visual Words", IEEE Transactions on Medical Imaging , vol 30, no.3, March 2011.

[2] U.Avni, J. Goldberger,M. Sharon, E.Konen, and H. Greenspan, “Chest X-ray characterization: From the organ identification to the pathology categorization,” presented at the 11thACMSIGMM International Conference on Multimedia Information Retrieval (MIR-2010), Philadelphia, PA, Mar.2010, pp. 29–31.

[3] C. B. Akgul, D. L. Rubin, S. Napel, C. F. Beaulieu, H. Greenspan, and B. Acar, “Content based image retrieval in radiology: Current status and future directions,” J. Digital Imag., Jan. 2010.

[4] S.Bhadoria and Dr. C. G. Dether, “Study of Medical Image Retrieval System”, International Conference on Data Storage and Data Engineering., 2010.

[5] U. Avni, J. Goldberger, and H. Greenspan, “Dense simple features for fast and accurate medical X-ray annotation,” in 10th Workshop of the Cross-Language Evaluation Forum (CLEF 2009), LNCS. New York: Springer,Lecture Notes in Computer Science, 2010.

[6] U.Avni, J. Goldberger,M. Sharon, E.Konen, and H. Greenspan, “Chest X-ray characterization: From the organ identification to the pathology categorization,” presented at the 11thACMSIGMM International Conference on Multimedia Information Retrieval (MIR-2010), Philadelphia, PA, Mar.2010, pp. 29–31.

[7] Li Jin, L.Hong, and T.Lianzhi, "A Mapping Modelling of Visual Feature and Knowledge Representation Approach for Medical Image Retrieval". Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation, Changchun, China,August 2009.

[8] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” Int. J. Comput. Vis., vol. 73, no. 2, 2007,pp. 213–238.

[9] E. Nowak, F. Jurie, and B. Triggs, “Sampling strategies for bag-of-features image classification,” in Proc. ECCV,2006, pp. 490–503.

[10] Liu, J., Hu, Y., Li, M., Ma, and W.-Y., 2006. Medical image annotation and retrieval using visual features. In: Working Notes of the 2006 CLEF Workshop.

[11] L. Fei-Fei and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” in Proc. CVPR, 2005, vol. 2, pp. 524–531.

[12] H. Alto, R. M. Rangayyan, and J. E. L. Desautels, “Content-based retrieval and analysis of mammographic masses,”J. Electron. Imag., vol. 14, no. 2, 2005.

[13] H. Müller, N. Michoux, D. Bandon, and A. Geissbühler, “A review of content-based image retrieval systems in medical applications—Clinical benefits and future directions,” I. J. Med. Informat., vol. 73, no. 1, 2004, pp. 1–23.

[14] S. Hong, C.Wencheng, Z.Jiwu and Z.Hong, "Medical image retrieval based on low level features and semantic features," Journal of Image and Graphic, vol. 9, no. 2, 2004, pp.220-224.

[15] T. F.Wu, C. J. Lin, R. C. Weng, “Probability Estimates for Multi-class Classification by Pairwise Coupling.”,Journal ofMachine Learning Research, vol. 5,2004, pp. 975–1005.

[16] T. M. Lehmann et al., “Content-based image retrieval in medical applications,” Methods Inf. Medicine, vol. 43, no. 4, Oct. 2004,pp.354–361,.

[17] Khanh Vu, Hua K.A and Tavanapong W., “Image Retrieval Based on Regions of Interest”, Knowledge and Data Engineering, IEEE Transactions on, vol. 15, April 2003,pp.1045-1049.

[18] E. Chang, G. Kingshy, G. Sychay, and G. Wu. “CBSA:content-based soft annotation for multimodal image retrieval using Bayes point machines”, IEEE Trans. on CSVT, 13(1), 2003,pp. 26–38.

[19] M. Varma and A. Zisserman, “Texture classification: Are filter banks necessary?,” in Proc. CVPR, vol. 2,2003, pp. 691–698.

[20] P. J. Eakins, “Towards Intelligent image retrieval.”, Pattern Recognition, vol. 35,2002, pp. 3–14.

[21] M. R. Naphade, C. Lin, J. R. Smith, B. Tseng, and S.Basu, “Learning to Annotate Video Databases,” Proceedings of SPIE, vol. 4676,2002, pp. 264–275.

[22] A. Smeulder, M.Worring, S. Santini, A. Gupta, R. Jain, “Content-Based Image Retrieval at the End of the Early Years.”, IEEE Trans. on Pattern Anal. and Machine Intell., vol.22, 2002, pp. 1349–1380.

[23] J. C. Bezdek, et al., "Fuzzy Models and Algorithms for Pattern Recognition and Image Processing"., Kluwer Academic Publishers, Boston, 1999.

[24] O. Chapelle, P. Haffner, V. Vapnik, “SVMs for histogram-based image classification.”, IEEE Trans. on Neural Networks, vol.10(5), 1999,pp. 1055–1064.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering:a review.”, ACM Computing Surveys, vol. 31(3),1999, pp.264–323.

[26] D. G. Lowe, “Object recognition from local scale-invariant features,” Proc. ICCV, vol. 2, 1999,pp. 1150–1157.

[27] P. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, “Fast and effective retrieval of medical tumor shapes,”IEEE Trans. Knowl. Data Eng., vol. 10, no. 6, Nov./Dec. 1998, pp. 889–904.

[28] E.G.M. Petrakis and C. Faloutsos, “Similarity searching in medical image databases,” IEEE Trans. Knowl. Data Eng., vol. 9, no. 3, 1997,pp. 435–447.

[29] C. Cortes and V. Vapnik, “Support-vector network”,Machine Learning, vol. 20, 1995,pp. 273–297.

[30] K. Fukunaga, "Introduction to Statistical Pattern Recognition".,second ed., Academic Press, 1990.

[31] C. Lei, Z. Jie, Y. Xiao-e, et al, "Fast lung segmentation algorithm for thoracic CT based on automated thresholding," Computer Engineering and Applications, vol. 44, no.12, 2008, pp.178-181.

[32] Qing-zhu Wang , Ke Wang "Medical Image Retrieval Based on Low Level Feature and High Level Semantic Feature," 2nd International

Conference on Computer Engineering and Technology, vol. 7, no.12, 2010,pp.430-432.

[33] O. Chapelle, P. Haffner and V. Vapnik, "SVMs for histogram based Image classification". IEEE Trans. On Neural Networks, vol 10,1999, pp.1055-1064.

[34] A.Abdullah, Remco C. Veltkamp and Marco A. Wiering, " Spatial Pyramids and Two-layer Stacking SVM Classifiers for Image categorization: A Comparative Study".

[35] I. F. Cruz and A. Rajendran, “Semantic data integration in hierarchical domains,” *IEEE Intell. Syst.*, vol. 18, no. 2, Mar.– Apr. 2003, pp-63-73.s

[36] P.Maheshwary, N.Sricastava, "Prototype System for Retrieval of Remote Sensing Images on Color Moment and Gray Level Co-occurrence Matrix," *IJCSI International Journal of Computer Science Issues*, Vol 3, 2009.

[37] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux. “Wavelet optimization for content-based image retrieval in medical databases.”, Medical Image Analysis, 14(2), 2010, pp.227 - 241.

[38] C. Thies, M.O. Guld, B Fischer, and T.M. Lehmann, "Content-based queries on the CasImage database within the IRMA framework”, Lecture Notes in Computer Science, Springer 3491, 2005, pp. 781–792.

[39] S. Antani, L.R. Long, and G.R. Thoma, “Content-based image retrieval for large biomedical image Archives”, Proceedings of 11th World Congress Medical Informatics,2004,pp.829–833.

[40] L.R. Long, S.K. Antani, and G.R. Thoma, “Image informatics at national research center”, Computer Medical Imaging and graphics (ELSEVIER), Vol. 29, 2005, pp.171–193.

[41] G.R. Thoma, L.R. Long, and S.K. Antani, “Biomedical imaging research and development: knowledge from images in the medical enterprise”, Technical Report Lister Hill National Centre for Biomedical Communications, 2006.

[42] E.G.M. Petrakis, and C. Faloutsos, “ImageMap: An Image Indexing Method Based on Spatial Similarity”, IEEE Transaction on Knowledge and Data Engineering, 2002, pp. 979–987.

[43] Chi-Ren Shyu, Carla E. Brodley, Avinash C. Kak, and Akio Kosaka,“ASSERT:A Physician-in-the-Loop Content-Based Retrieval System for HRCT Image Databases”, Computer Vision and Image Understanding, Vol. 75, No. 1, 1999, pp. 111–132.

[44] L.R. Long, S.R. Pillemer, R.C. Lawrence, GH Goh, L. Neve, and G.R. Thoma, “WebMIRS:Web-based Medical Information Retrieval System” , Proceedings of SPIE Storage and Retrieval for Image and Video Databases VI,SPIE ,Vol. 3312, 1998, pp. 392-403.

Cursor Control by Point-of-Regard Estimation for a Computer With Integrated Webcam

Stefania Cristina, Kenneth P. Camilleri

Department of Systems and Control Engineering
University of Malta, Malta
Email: stefania.cristina@um.edu.mt
kenneth.camilleri@um.edu.mt

***Abstract*—The problem of eye-gaze tracking by videooculography has been receiving extensive interest throughout the years owing to the wide range of applications associated with this technology. Nonetheless, the emergence of a new paradigm referred to as pervasive eye-gaze tracking, introduces new challenges that go beyond the typical conditions for which classical video-based eye-gaze tracking methods have been developed. In this paper, we propose to deal with the problem of point-of-regard estimation from low-quality images acquired by an integrated camera inside a notebook computer. The proposed method detects the iris region from low-resolution eye region images by its intensity values rather than the shape, ensuring that this region can also be detected at different angles of rotation and under partial occlusion by the eyelids. Following the calculation of the point-of-regard from the estimated iris center coordinates, a number of Kalman filters improve upon the noisy point-of-regard estimates to smoothen the trajectory of the mouse cursor on the monitor screen. Quantitative results obtained from a validation procedure reveal a low mean error that is within the footprint of the average on-screen icon.**

Keywords–Point-of-regard estimation; Eye-gaze tracking; Iris center localization.

I. Introduction

The idea of estimating the human eye-gaze has been receiving increasing interest since at least the 1870s [1], following the realization that the eye movements hold important information that relates to visual attention. Throughout the years, efforts in improving eye-gaze tracking devices to minimize discomfort and direct contact with the user led to the conception of videooculography (VOG), whereby the eye movements are tracked remotely from a stream of images that is captured by digital cameras. Eye-gaze tracking by VOG quickly found its way into a host of applications, ranging from human-computer interaction (HCI) [2], to automotive engineering [3][4]. Indeed, with the advent of the personal computer, eye-gaze tracking technology was identified as an alternative controlling medium enabling the user to operate the mouse cursor using the eye movements alone [2].

Following the emergence and widespread use of highly mobile devices with integrated imaging hardware, there has been an increasing interest in mobile eye-gaze tracking that blends well into the daily life setting of the user [5]. This emerging interest led to the conception of a new paradigm that is referred to as *pervasive tracking*, which term refers to the endeavor for tracking the eye movements continuously in different real-life scenarios [5]. This notion of pervasive eye-gaze tracking is multi-faceted, typically characterized by different aspects such as the capability of a tracking platform to permit tracking inside less constrained conditions, to track the user remotely and unobtrusively and to integrate well into devices that already comprise imaging hardware without necessitating hardware modification.

Nevertheless, this new paradigm brings challenges that go beyond the typical conditions for which classical video-based eye-gaze tracking methods have been developed. Despite considerable advances in the field of eye-gaze tracking as evidenced by an abundance of methods proposed over the years [6], video-based eye-gaze tracking has been mainly considered a desktop technology, often requiring specific conditions to operate. Commercially available eye-gaze tracking systems, for instance, are usually equipped with high-grade cameras and actively project infra-red illumination over the face and the eyes to obtain accurate eye movement measurements. In utilizing specialized hardware to operate, active eye-gaze tracking fails to integrate well into devices that already comprise imaging hardware, while its usability is constrained to controlled environments away from interfering infra-red sources. On the other hand, passive eye-gaze tracking which operates via standard imaging hardware and exploits the appearance of the eye without relying on specialized illumination sources for localization and tracking, provides a solution that promises to integrate better into pervasive scenarios.

Nonetheless, utilizing existing passive eye-gaze tracking methods to address the challenges associated with pervasive tracking, such as the measurement of eye movement from low-quality images captured by lower-grade hardware, may not necessarily be a suitable solution. For instance, existing shape-based methods that localize the eye region inside an image frame by fitting curves to its contours, often require images of suitable quality and good contrast in which the boundaries between different components such as the eyelids, the sclera and the iris are clearly distinguishable [7]–[10]. Similarly, feature-based methods that search for distinctive features such as the limbus boundary [11][12], necessitate these features to be clearly identifiable. It has been reported in [13], that appearance-based methods relying on a trained classifier, such as a Support Vector Machine (SVM), to estimate the 2D point-of-regard directly from an eye region image without identifying its separate components, perform relatively well on lower-quality images as long as the training data includes images of similar quality as well. However, this performance usually comes at the cost of lengthy calibration sessions that serve to gather the user-dependent data that is required for training [14][15]. Moreover, recent attempts to track the eye-gaze on mobile platforms by existing eye-gaze tracking methods [16][17], have reported undesirable constraints such as a requirement for close-up eye region images [18] and lengthy calibration sessions [16].

In light of the challenges associated with pervasive tracking, we propose a passive eye-gaze tracking method to estimate the point-of-regard (POR) on a monitor screen from lower-quality images acquired by an integrated camera inside a notebook computer. To localize the iris center coordinates from low-resolution eye region images while the user sits at a distance from the monitor screen, we propose an appearance-based method that localizes the iris region by its intensity values rather than the shape. In addition, our method ensures that the iris region can be located at different angles of rotation and under partial occlusion by the eyelids, and can be automatically relocated after this has been entirely occluded during blinking. Following iris localization, the iris center coordinates extracted earlier are mapped to a POR on the monitor screen via linear mapping functions that are estimated through a brief calibration procedure. A number of Kalman filters finally improve upon the noisy POR estimates to smoothen the trajectory of the mouse cursor on the monitor screen.

This paper is organized as follows. Section II describes the details of the proposed passive eye-gaze tracking method. Section III presents and discusses the experimental results, while Section IV draws the final remarks which conclude the paper.

II. Method

The following sections describe the stages of the proposed method, starting off with eye region detection and tracking up to the estimation of the POR onto the monitor screen.

A. Eye Region Detection

The estimation of the POR on the monitor screen requires that the eye region is initially detected inside the first few image frames. Searching for the eye region over an entire image frame can be computationally expensive for a real-time application and can lead to the occurrence of several false positive detections. Therefore, prior to detecting the eye region, the bounding box that encloses the face region is detected first such that this constrains the search range for the eye region, reducing the searching time as well as the possibility of false positives. The eye region is subsequently detected within the area delimited by the boundaries of the face region.

Given the real-time nature of our application, we chose the Viola-Jones algorithm for rapid detection of the face and eye region [19]. Within the Viola-Jones framework, features of interest are detected by sliding rectangular windows of Haar-like operators over an image frame, subtracting the underlying image pixels that fall within the shaded regions of the Haar-like operators from the image pixels that fall within the clear regions. Candidate image patches are classified between positive and negative samples by a cascade of weak classifiers arranged in order of increasing complexity. Every weak classifier is trained to search for a specific set of Haar features by a technique called boosting, such that each stage processes the samples that pass through the preceding classifier and rejects the negative samples as early into the cascade as possible to ensure computational efficiency.

The face and eye region detection stages in our work utilize freely available cascades of classifiers that come with the OpenCV library [20], which had been previously trained on a wide variety of training images such that detection generalizes well across different users. Since the training data for these classifiers was mainly composed of frontal face and eye region samples, the user is required to hold a frontal head pose for a brief period of time until the face and eye regions have been successfully detected. In case multiple candidates are detected by the face region classifier, the proposed method chooses the candidate that is closest to the monitor screen characterized by the largest bounding box, and discards the others.

B. Eye Region Tracking

To allow for small and natural head movement during tracking without requiring the uncomfortable use of a chin-rest, the initial position of the eye region detected earlier needs to be updated at every image frame to account for its displacement in the x- and y-directions. While performing eye region detection on a frame-by-frame basis would be a possible solution to estimate the eye region displacement through an image sequence, such an approach would be sub-optimal in terms of computational efficiency for a real-time application. Therefore, assuming gradual and small head displacement, the eye region is tracked between successive image frames by template matching, using the last known position of the eye region inside the previous image frame to constrain the search area inside the next frame.

A template image of the eye region is captured and stored following earlier detection of this region by the Viola-Jones algorithm. The template image is then matched to the search image inside a window of fixed size, centered around the last known position of the feature of interest. Template matching utilizes the normalized sum of squared differences (NSSD) as a measure of similarity, denoted as follows,

$$NSSD(x,y) = \frac{\sum_{x',y'}[T(x',x') - I(x+x',y+y')]^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \sum_{x',y'} I(x+x',y+y')}} \tag{1}$$

where T denotes the template image and I denotes the search image. A NSSD value of zero represents a perfect match between the template and search image, whereas a higher value denotes increasing mismatch between the two images. This permits the identification of the new position of the feature of interest, which is specified by the location inside the search image that gives the minimum NSSD value after template matching.

C. Iris Center Localization

The movement of the eyes is commonly represented by the trajectory of the iris or pupil center in a stream of image frames [6], and hence the significance of localizing the iris or pupil center coordinates after the eye region has been detected. Given the small footprint of the eye region inside the image space, we opt to localize the iris center coordinates rather than the pupil, since the iris occupies a larger area inside the eye region and can be detected more reliably.

While there exist different methods that permit localization of the iris region inside an image frame, not all of these methods are suitable for localizing the iris region from low-resolution images, especially if fine details such as the contours of different components of the eye [7]–[10] need to be clearly distinguishable. We propose an appearance-based method that segments the iris region via a Bayes' classifier to localize it. The Bayes' classifier is trained during an offline training stage to classify between iris and non-iris pixels based on their red channel value in the RGB color space. During tracking, intensity values of pixels residing within the eye region are classified as belonging to the iris region if their likelihood exceeds a pre-defined threshold value, θ:

$$\frac{p(x_r(i,j) \mid \varpi_{iris})}{p(x_r(i,j) \mid \varpi_{non-iris})} \geq \theta \qquad (2)$$

where $p(x_r(i,j) \mid \varpi_{iris})$ denotes the class-conditional probability of observing a red-band measurement at pixel *(i, j)* knowing it belongs to the iris class, while $p(x_r(i,j) \mid \varpi_{non-iris})$ denotes the class-conditional probability of observing the same red-band measurement at pixel *(i, j)* knowing it belongs to the non-iris class. The resulting binary image contains a blob of pixels that belongs to the iris region, whose center of mass is taken to represent the iris center coordinates. In case the eyebrow is also mistakenly classified as belonging to the iris region due to the resemblance in color with dark irises, the blob of pixels that is closer to the center of the eye region is considered to represent the iris.

The Bayes' classifier had been previously used for skin region segmentation in images [21], but to our knowledge it has never been adopted to the problem of iris region localization for eye-gaze tracking until our work. Preliminary results have shown this method to be suitable in localizing the iris region from low-quality images, owing especially to the fact that the proposed localization method depends upon statistical color modeling rather than geometrical information. Another advantage that is also related to its independency from geometrical information is the ability to locate the iris region at different angles of rotation and under partial occlusion by the eyelids. The main downside of this method is its susceptibility to illumination variations, which problem is however alleviated by training the Bayes' classifier on iris and non-iris pixels acquired under different illumination conditions.

D. POR Estimation

Having determined the iris center coordinates, the final stage seeks to map these coordinates to screen coordinates in order to estimate the user's POR on the monitor screen.

For simplicity, we assume the iris center in the image space to displace along a flat plane, such that we can define a linear mapping relationship between the image and screen coordinates as follows,

$$(\mathbf{x}_s^{(3)} - \mathbf{x}_s^{(1)}) = \frac{(\mathbf{x}_s^{(2)} - \mathbf{x}_s^{(1)})}{(\mathbf{x}_i^{(2)} - \mathbf{x}_i^{(1)})}(\mathbf{x}_i^{(3)} - \mathbf{x}_i^{(1)}) \qquad (3)$$

where $\mathbf{x}_s^{(1)}$ and $\mathbf{x}_s^{(2)}$ denote the screen coordinates of two calibration points respectively, whereas $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ denote the corresponding iris center coordinates inside the eye region which are estimated while the user fixates at the two calibration points. During tracking, the mapping function in (3) computes the displacement in screen coordinates between the new POR $\mathbf{x}_s^{(3)}$ and the calibration point $\mathbf{x}_s^{(1)}$, following the estimation of the displacement in image coordinates between the new iris center location $\mathbf{x}_i^{(3)}$ and the previously estimated $\mathbf{x}_i^{(1)}$. In order to compensate for the assumption of planar iris movement, the monitor screen is divided into four separate quadrants by strategically placed calibration points as illustrated in Figure 1, such that each quadrant is assigned different parameter values that best describe the linear mapping between the image-to-screen coordinates.

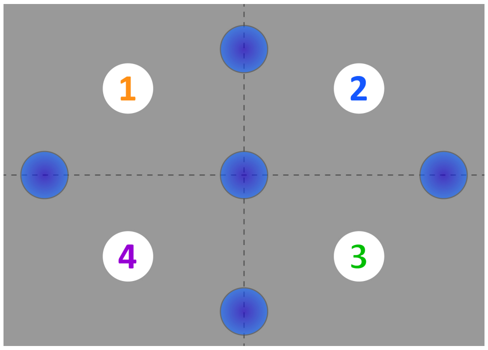


Figure 1. Strategically placed calibration points divide the screen display into four separate quadrants.

To alleviate the issue of noisy iris center estimations from low-quality images, and hence smoothen the trajectory of the mouse cursor on the monitor screen after mapping the iris center coordinates to a POR, we propose to use a Kalman filter to improve upon these noisy measurements. Indeed, the Kalman filter is an algorithm that recursively utilizes noisy measurements observed over time to produce estimates of desired variables that tend to be more accurate than the single measurements alone [22]. We define the Kalman filter parameters for our specific application of smoothing the mouse cursor trajectory as follows:

State Vector: We define the state vector $\mathbf{x}_{k+1}$ as,

$$\mathbf{x}_{k+1} = [\Delta x_s \quad \Delta y_s]^T \qquad (4)$$

where Δx_s denotes the horizontal on-screen displacement, $(x_s^{(3)} - x_s^{(1)})$, and similarly for the vertical on-screen displacement, Δy_s.

Transition Matrix: Assuming the eye movement during tracking to consist of fixation periods and smooth movement between one visual stimulus and another, we represent the transition matrix A_{k+1} by a simple linear model of the ideal mouse cursor trajectory during fixations and shifts between visual stimuli as follows,

$$A_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (5)$$

Measurement Vector: In our work, the measurement vector $\mathbf{z}_{k+1}$ holds the estimated displacement of the iris center in image coordinates and is defined as,

$$\mathbf{z}_{k+1} = [\Delta x_i \quad \Delta y_i]^T \qquad (6)$$

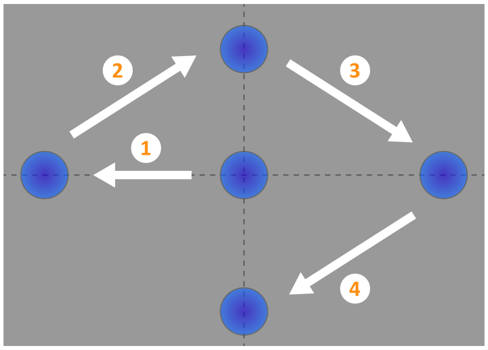


Figure 2. Five visual stimuli were displayed in succession on the monitor screen during a brief calibration procedure in order to collect image-screen coordinate pairs.

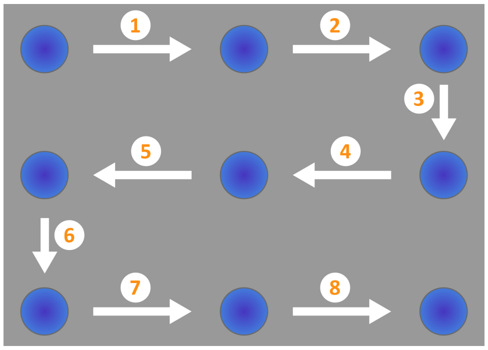


Figure 3. A validation session consisting of nine visual stimuli displayed in succession served to calculate the error of the estimated PORs.

where Δx_i represents the horizontal image displacement, $(x_i^{(3)} - x_i^{(1)})$, and similarly for the vertical image displacement, Δy_i.

Measurement Matrix: The measurement matrix defines the relationship that maps the true state space onto the measurement. In our work, the values that populate the measurement matrix can be derived from (3), such that this matrix maps the screen coordinates onto the image coordinates,

$$H_{k+1} = \begin{bmatrix} \frac{(x_i^{(2)} - x_i^{(1)})}{(x_s^{(2)} - x_s^{(1)})} & 0 \\ 0 & \frac{(y_i^{(2)} - y_i^{(1)})}{(y_s^{(2)} - y_s^{(1)})} \end{bmatrix} \qquad (7)$$

Measurement Noise and Process Noise: The measurement noise is represented by vector, $\mathbf{v}_{k+1} = [v_{k+1}^x v_{k+1}^y]$, characterized by standard deviations δ_{v_x} and δ_{v_y} in the x- and y-directions respectively, and similarly the process noise is represented by vector, $\mathbf{w}_{k+1} = [w_{k+1}^x w_{k+1}^y]$, characterized by standard deviations δ_{w_x} and δ_{w_y} in the respective x- and y-directions. The process noise is taken to represent the characteristics inherent to the visual system itself, such that the standard deviations δ_{w_x} and δ_{w_y} are therefore set to a low value to model the small, microsaccadic movements performed by the eye during periods of fixation. Values for the standard deviations, δ_{v_x} and δ_{v_y}, that adequately smooth the mouse cursor trajectory after the estimation of noisy iris center measurements were found experimentally.

Separate Kalman filters are assigned to every screen quadrant, with each filter being characterized by a different measurement matrix corresponding to the screen quadrant for which it is responsible. During tracking, all Kalman filters are updated online to produce an estimate of the POR following the estimation of the iris center coordinates, such that the on-screen position of the mouse cursor can subsequently be updated according to the Kalman filter estimate that corresponds to the quadrant of interest. In updating the Kalman filters at every time step, we ensure a smooth hand over between one filter and another as the mouse cursor trajectory crosses over adjacent screen quadrants.

III. Experimental Results and Discussion

To evaluate the proposed eye-gaze tracking method, a group of five participants consisting of two females and three males with a mean age of 38.2 and standard deviation of 15.9, were recruited for an experimental session. All participants were proficient computer users without any prior experience in the field of eye-gaze tracking, except one who was already accustomed to the technology. The experimental procedure was carried out on a 15.6” notebook display while each participant was seated inside a well-lit indoor environment at an approximate distance of 60 cm from the monitor screen and the camera. Image data was acquired by the webcam that was readily available on-board the notebook computer.

Following detection and tracking of the eye region, and iris center localization, each participant was requested to sit through a brief calibration procedure that served to estimate the mapping functions required to transform the iris center coordinates into a POR on the monitor screen. During the calibration procedure, the participants were instructed to fixate at five visual stimuli appearing in succession on the monitor screen as shown in Figure 2, and requested to minimize their head movement such that pairs of image-screen coordinates were collected. The five visual stimuli were positioned strategically in order to divide the screen into four separate quadrants, as illustrated in Figure 1. A different mapping function was estimated for each quadrant according to the relationship between the image and screen coordinates collected earlier.

Every participant was then requested to sit through a validation procedure that served to estimate the error between the estimated POR and ground truth data. The validation procedure consisted of nine visual stimuli which were evenly spread throughout the monitor screen and displayed in succession as shown in Figure 3. The participants were instructed to move the mouse cursor with their eyes as close to each visual stimulus as possible and hold its position for a brief period of time such that the on-screen coordinates of the mouse cursor were recorded, as shown in Figure 4 for one of the participants. During the validation procedure, the participants were allowed a small degree of head movement and were requested to displace their head in a natural way. Table I displays the mean and standard deviation of the error in pixels for each participant in the x- and y-directions.

By analyzing the results in Table I, it can be observed that in all cases the mean and standard deviation of the error in the x-direction exceeds the error in the y-direction. The main source for this discrepancy in error relates to inaccuracies in estimating the iris center coordinates. For instance, it was noted that despite retaining similar surrounding conditions between different participants to reduce any bias in the results, the

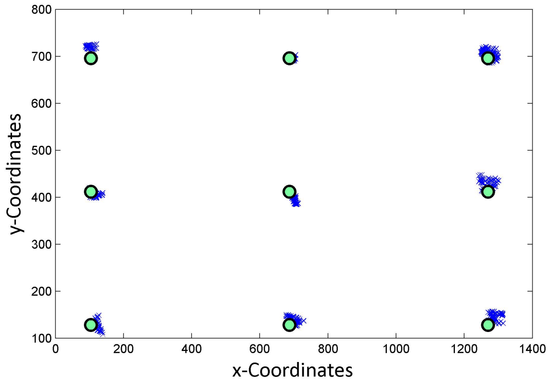


Figure 4. Validation result for one of the participants showing the displayed visual stimuli (green) and the estimated on-screen PORs (blue).

TABLE I. MEAN AND STANDARD DEVIATION OF THE ERROR IN PIXELS IN THE X- AND Y-DIRECTIONS, OF THE ESTIMATED ON-SCREEN POR COORDINATES.

Participant Number	Mean (x, y)	Standard Deviation (x, y)
1	(62.47, 19.63)	(130.89, 14.14)
2	(41.88, 13.63)	(132.76, 10.82)
3	(88.37, 45.62)	(177.96, 79.82)
4	(47.21, 16.89)	(107.75, 24.10)
5	(36.57, 17.10)	(109.19, 34.81)

accuracy of the collected data was subject to the anatomical characteristics of the participants. The protruding ridge of the brow above the eye was more accentuated for some participants rather than others, and this tended to create a dark shadow around the inner corner of the eye which was at times incorrectly segmented along with the iris region. Dark colored pixels belonging to the eyelashes were also often segmented with the iris region as shown in Figure 5, due to their close resemblance in color to dark brown irises. This erroneous inclusion of pixels which do not belong to the iris region served to shift the center of mass of the segmented blob of pixels horizontally towards the inner or outer eye corners, away from the true iris center. It was found that even a seemingly trivial error of a few pixels in the estimation of the iris center coordinates inside the image frame, could result in a significant error in the estimation of the POR at an approximate distance of 60 cm from the monitor screen.

The main source for the error in the y-direction could be the less than ideal positioning of the webcam at the top of the monitor screen, in relation to the positioning of the eyes as the user sits in front of the display. Indeed, commercial systems usually place the tracking device below the monitor screen in order to capture a better view of the visible portion of the eyeball that is not concealed below the eyelid. Being situated at the top of the screen, the webcam that is utilized in our work captures a smaller portion of the iris especially when the user gazes downwards, partially occluding the iris region below the eyelid and potentially introducing an error in the estimation of the iris center coordinates. It is, however, worth noting that the proposed method for iris region segmentation was equally capable of detecting the iris region under partial occlusion by the eyelids as shown in Figure 6, and therefore suitably alleviated the issue of the less than ideal positioning of the webcam.

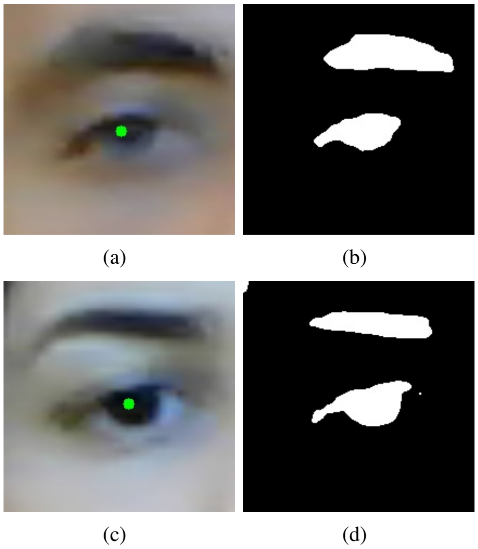

(a) (b)
(c) (d)

Figure 5. Inclusion of eyelashes in the segmented iris regions in Figures (b) and (d) corresponding to the eye region images in Figures (a) and (c).

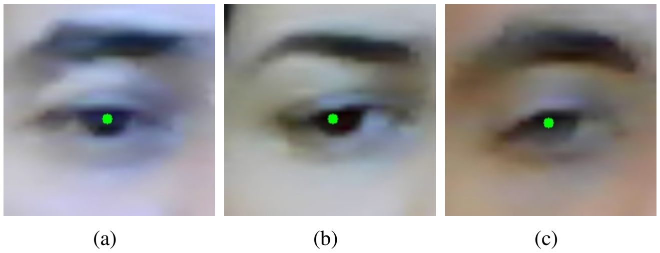

(a) (b) (c)

Figure 6. The proposed method for iris region segmentation was capable of localizing the iris center coordinates marked in green, under partial occlusion of the iris region by the eyelids.

In order to put the error values tabulated in Table I into context, the mean error in pixels across all participants was calculated and compared to the resolution of the computer screen. Given that the resolution of the monitor screen is equal to 1366 × 768 pixels, a mean error value of (55.30, 22.57) constitutes around 4% and 3% of the screen resolution in the horizontal and vertical directions respectively. Also, at a distance of 60 cm away from the monitor screen, a mean error of (55.30, 22.57) pixels corresponds to (1.46°, 0.71°) in visual angle. If the average on-screen icon is taken to have an average size of 45 × 45 pixels, the mean error that is achieved through the proposed method can be considered to be within the footprint of the average on-screen icon and therefore applicable to an HCI scenario.

IV. CONCLUSION

In this paper, we have proposed a passive eye-gaze tracking method to estimate the POR on a monitor screen from low-quality image data acquired by an integrated camera inside a notebook computer. Following eye region detection and tracking, we proposed an appearance-based method which allows the localization of the iris center coordinates from low-resolution eye region images. The iris center coordinates were subsequently mapped to a POR on the monitor screen by linear mapping functions, following which each POR estimate was improved by Kalman filters to smoothen the mouse cursor trajectory.

The experimental results obtained following a validation procedure revealed a noticeable discrepancy between the error in the x- and y-directions, with the error in the x-direction being the dominant between the two. The source for this error was observed to be the incorrect segmentation of pixels belonging to shadow or artifacts, such as the eyelashes, along with the iris region, producing a horizontal shift away from the true iris center. Nonetheless, the proposed method for iris region segmentation was capable of detecting the iris region under partial occlusion by the eyelids, especially when the user gazes downwards, permitting the estimation of the POR in less than ideal conditions. It is noteworthy to mention that despite the availability of low-resolution eye region images, the proposed method achieved a relatively low mean error of (1.46°, 0.71°) in visual angle. Future work aims to compensate for head movement inside the mapping functions, which map the iris center coordinates to a POR on the monitor screen, in order to permit larger head movement during tracking.

ACKNOWLEDGMENT

This work forms part of the project *Eye-Communicate* funded by the Malta Council for Science and Technology through the National Research & Innovation Programme (2012) under Research Grant No. R&I-2012-057.

REFERENCES

[1] R. J. K. Jacob, "What you look at is what you get: eye movement-based interaction techniques," in Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, 1990, pp. 11–18.

[2] J. L. Levine, "An eye-controlled computer," in IBM Thomas J. Watson Research Center Res. Rep. RC-8857, Yorktown Heights, N.Y., 1981.

[3] S. J. Lee, J. Jo, H. G. June, K. R. Park, and J. Kim, "Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System," IEEE Transactions on Intelligent Transportation Systems, vol. 12, 2011.

[4] M. Shahid, T. Nawaz, and H. A. Habib, "Eye-Gaze and Augmented Reality Framework for Driver Assistance," Life Science Journal, vol. 10, 2013, pp. 1571–1578.

[5] A. Bulling, A. T. Duchowski, and P. Majaranta, "PETMEI 2011: The 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction," in Proceedings of the 13th International Conference on Ubiquitous Computing: UbiComp 2011, 2011.

[6] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, 2010, pp. 478–500.

[7] R. Valenti, A. Lablack, N. Sebe, C. Djeraba, and T. Gevers, "Visual Gaze Estimation by Joint Head and Eye Information," in 20th International Conference on Pattern Recognition, Aug. 2010, pp. 3870–3873.

[8] R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation," IEEE Transactions on Image Processing, vol. 21, 2012, pp. 802–815.

[9] W. Haiyuan, Y. Kitagawaa, and T. Wada, "Tracking Iris Contour with a 3D Eye-Model for Gaze Estimation," in Proceedings of the 8th Asian Conference on Computer Vision, Nov. 2007, pp. 688–697.

[10] T. Moriyama, T. Kanade, J. Xiao, and J. F. Cohn, "Meticulously Detailed Eye Region Model and Its Application to Analysis of Facial Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, 2006.

[11] F. Timm and E. Barth, "Accurate Eye Centre Localisation by Means of Gradients," in Proceedings of the Sixth International Conference on Computer Vision Theory and Applications, Mar. 2011, pp. 125–130.

[12] E. G. Dehkordi, M. Mahlouji, and H. E. Komleh, "Human Eye Tracking Using Particle Filters," International Journal of Computer Science Issues, vol. 10, 2013, pp. 107–115.

[13] J. Mansanet, A. Albiol, R. Paredes, J. M. Mossi, and A. Albiol, "Estimating Point of Regard with a Consumer Camera at a Distance," Pattern Recognition and Image Analysis, 2013, pp. 881–888.

[14] W. Sewell and O. Komogortsev, "Real-Time Eye Gaze Tracking With an Unmodified Commodity Webcam Employing a Neural Network," in Proceedings of the 28th International Conference Extended Abstracts on Human Factors in Computing Systems, Apr. 2010, pp. 3739–3744.

[15] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking Eyes and Monitoring Eye Gaze," in Proceedings of the Workshop on Perceptual User Interfaces, Oct. 1997, pp. 98–100.

[16] C. Holland and O. Komogortsev, "Eye Tracking on Unmodified Common Tablets: Challenges and Solutions," in Proceedings of the Symposium on Eye Tracking Research and Applications, Mar. 2012, pp. 277–280.

[17] K. Kunze, S. Ishimaru, Y. Utsumi, and K. Kise, "My Reading Life - Towards Utilizing Eyetracking on Unmodified Tablets and Phones," in Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing, 2013, pp. 283–286.

[18] Y. Zhang, A. Bulling, and H. Gellersen, "Towards pervasive eye tracking using low-level image features," in Proceedings of the Symposium on Eye Tracking Research and Applications, 2012, pp. 511–518.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jul. 2001, pp. 1231–1238.

[20] "OpenCV," 2014, URL: http://http://opencv.org/ [accessed: 2014-04-21].

[21] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," in Proceedings of the GraphiCon, 2003, pp. 85–92.

[22] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, 1960, pp. 35–45.

Automating Green Patterns to Compensate CO_2 Emissions of Cloud-based Business Processes

Alexander Nowak, Uwe Breitenbücher, Frank Leymann
Institute of Architecture of Application Systems (IAAS)
University of Stuttgart
Stuttgart, Germany
firstname.lastname@iaas.uni-stuttgart.de

***Abstract*—The usefulness of patterns to optimize the environmental impact of business processes and their infrastructure has already been described in literature. However, due to the abstract description of pattern solutions, the individual application of patterns has to be done manually which is time consuming, complex, and error-prone. In this work, we show how the Green Compensation pattern can be applied automatically to different individual Cloud-based business processes in order to lower the negative environmental impact of the employed Virtual Machines without any manual effort. We show how our Management Planlet Framework can be used to implement this concrete refined pattern solution in a reusable way.**

Keywords-Green Business Process Patterns; Management Automation; Cloud Computing; Infrastructure as a Service.

I. INTRODUCTION

Today, organizations widely use business processes to describe the way they are doing business. Business processes compose and orchestrate various business services that are used to reach certain business objectives. The management and optimization of these business processes and their supporting infrastructure, therefore, has become an inherent part of organizational development. So far, typical optimization dimensions have been limited to cost, time, quality, and flexibility. This scope has been extended in recent years by the consideration of ecological indicators and measurements [1][2]. Based on legal requirements and increased customer awareness, more and more organizations keep track of and improve their environmental impact to save or extend their market shares [3][4].

There have been developed first approaches to guide and support organizations with tracking their environmental impact, like the ISO Standard 14001 [5]. Beyond that, a fundamental concept to improve the environmental impact of business processes is based on patterns that capture abstract knowledge about solutions for frequently recurring ecological issues, forces, and challenges. In Nowak et al. [6][7], we presented a set of *Green Business Process Patterns* that capture expertise about how to optimize business processes in terms of their environmental impact. However, these pattern-based approaches typically need to be refined manually for individual use cases due to the generic nature of patterns. Thus, the abstract solution described by patterns needs to be refined towards the concrete individual application to which it is applied to, in this context, the actual business processes, services, and underlying infrastructures. The problem of this manual refinement step is that (i) a deep technical knowledge as well as profound application management expertise is required to map the described abstract solution concepts to concrete circumstances [12], (ii) manual transformation steps are typically error-prone and time consuming [8], and (iii) the integration of heterogeneous process and application infrastructures needs to be handled. Especially for complex business processes that are commonly employed in enterprises today, a manual application of patterns does also come with additional risks, namely (i) unintended changes of the processes' semantics, (ii) unpredictable side effects to other processes, and (iii) technical errors. Hence, the risk is much too high to apply such changes manually.

As a result, there is a gap between the abstract nature of patterns and their actual application to real use cases that prevents applying the concept of patterns efficiently to the domain of Green Business Process Management. In addition, when applying changes to Cloud-based applications invoked from business process, the changes must be automated to ensure Cloud properties such as self-service, on-demand computing, and elasticity [29]. Therefore, we need a means that allows applying the generic knowledge captured in Green Business Process Patterns automatically to individual use cases. In this paper, we tackle these issues and present a conceptual approach to apply a pattern called *Green Compensation* [6] generically to various kinds of IaaS-based business processes that employ Virtual Machines (VMs) as underlying infrastructure service. The contributions of this paper are threefold: we first (i) describe the *concrete* refinement of the *abstract* Green Compensation pattern towards this VM-based use case, (ii) we present how the refined pattern can be automated independently from individual applications using the Management Planlet Framework [9][10][11][12], and subsequently we describe(iii) how our realization enables Cloud providers as well as customers to decrease the ecological impact of their applications significantly without any manual effort.

The remainder is structured as follows: Section II describes background information about patterns and their application in general as well as the Green Compensation pattern and its use in a motivating scenario. Section III describes our approach by refining the Green Compensation pattern solution and applying the methods from the employed Management Planlet Framework. Section IV provides related work and Section V concludes the paper.

II. Background, Requirements, and Motivation Scenario

In this section, we provide background information about patterns in general and the difficulties of automating their application. We introduce a Green Business Process Pattern named *Green Compensation* and a motivating scenario that is used throughout the paper explain our approach.

A. Patterns and their Application

Patterns describe problems that we can find over and over again in our environment or knowledge domain and the corresponding core of a solution to that problem [13]. Such an abstract solution, therefore, does not describe a single solution for a specific use case but describes the solution in a way such that it can be used again and again in individual scenarios [13]. Patterns usually have relations to other patterns indicating whether the fit together or not, or can be used as an alternative. Such a set of patterns for a particular domain is typically described as a *pattern language*.

To address ecological aspects of enterprises, we introduced *Green Business Process Patterns* [6][7] that help stakeholders to reduce the negative environmental impact of their business processes. Here, the term “green” is used as a synonym for all aspects related to the environmental impact such as CO_2 emissions, pollution, waste, etc. Considering Cloud-based business processes, the environmental impact of a process not only depends on the structure of that process but also on the services that are invoked. Therefore, Green Business Process Patterns do not focus on the control and data flow of processes only, but also on the resources that are used as well as the communication among them.

Due to the abstract description of pattern solutions, their application typically requires a *manual refinement* of the patterns towards the actual use case in which they shall be applied [12][14]. Thus, the abstract solution described by a pattern must be transformed into a description of concrete management tasks that have to be executed to apply the pattern. In the domain of Cloud Computing, this means dealing with low level technical issues such as orchestrating management APIs or executing configuration scripts [10]. Unfortunately, this is a technically complex, time consuming, and error prone challenge [8][12]. To tackle these issues, Cloud application management and, therefore, applying patterns in this domain must be automated. As a result, the two main issues that have to be tackled when applying the concept of patterns efficiently to the domain of Cloud Computing are (i) handling the technical complexity of pattern refinement for individual use cases and (ii) automating this refinement step and its actual execution to avoid manual errors [12]. The approach presented in this paper considers these issues and provides an automated means to apply patterns to individual use cases based on a formerly developed Application Management Framework.

B. Green Compensation Pattern

In this paper, we use a motivating scenario to better describe our approach. As part of this motivating scenario, we show how the pattern *Green Compensation* can be refined and applied in an automated manner in order to cover the complex management tasks necessary to realize this pattern. Therefore, we will first introduce the Green Compensation pattern by providing a shortened description of the pattern. The complete description of the Green Compensation pattern can be found in Nowak et al. [6].

To ease the use, handling, and identification of patterns, they are typically described in a common format. We use a shortened format originating from Nowak et al. [6] that consists of an *icon* and *intend* that allow a quick identification and classification of the pattern. Subsequently, the *context* describes the scenario in which the pattern might be used, the *problem* describes the forces the pattern is confronted with, the *solution* provides an abstract and implementation independent solution of the problem, the *result* describes the context after applying the pattern, and the *example* provides different scenarios where the pattern has been successfully applied to. Given that format, the Green Compensation pattern is described as follows:

Pattern: Green Compensation

Improve the negative environmental impact without changing the process model or the resources of a business process.

Context: An Enterprise uses a multi-services value stream and wants to contribute to the harmony and health of the environment without changing processes and resources.

Problem: Enterprises typically describe their business processes based on various functional requirements. This line of action ensures that business processes are able to reach defined business objectives. However, when considering environmental objectives, the design of a process may differ from the functional driven design. Due to internal or legislative guidelines, these changes in the process model may not be allowed. Thus, the challenge is to improve the negative environmental impact of a business process without changing its structure or employed resources.

Solution: Each time a business process or service that can’t be altered is instantiated, a corresponding compensation process or activity will be instantiated as well. A compensation process or activity is used to compensate at least parts of the negative environmental impact by using services that invest in climate projects, for example. Therefore, the environmental impact must be quantified accordingly. Figure 1 describes the different steps of the abstract solution as a process modeled in BPMN [15].

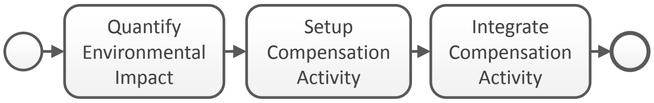

Quantify Environmental Impact
Setup Compensation Activity
Integrate Compensation Activity

Figure 1. Abstract Solution of the Green Compensation Pattern.

Result: Although the original processes or services will not be altered, their negative environmental impact can be reduced from a global point of view. The negative impact will not be eliminated, however, it will be compensated with a positive impact from other projects.

Example: The oil company JET offers customers in Germany a sustainability program that compensates the CO_2 emission caused by the amount of gas they refuel [16]. The railroad company Deutsche Bahn also offers a sustainability program where the CO_2 emissions of customers may be compensated [17]. Companies that do not run own programs can use the service of MyClimate [18], for example.

C. Motivating Scenario

In this section, we introduce a case study that is used as motivating scenario throughout the paper to illustrate the difficulties of refining abstract solutions from patterns and applying them in an automated fashion to individual use cases. Let us consider a company called *WiFo* that provides a service for electricity producers that forecasts the wind situation for the next day. For flexibility reasons, WiFo runs various service-based applications in a Cloud that are orchestrated by business processes. One of their main processes is depicted in Figure 2 and consists of three steps: (1) the simulation data needs to be set up, (2) the forecast calculation is performed, and (3) the results are analyzed.

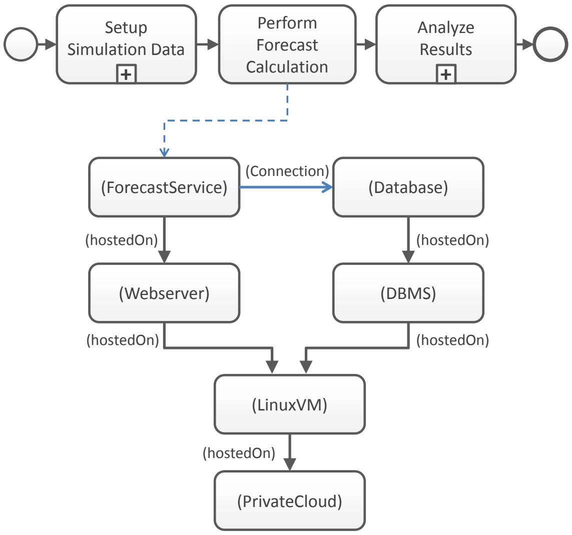


Figure 2. Motivating Scenario.

As we can see in Figure 2, step (1) and step (3) are modeled as sub-processes, i.e., these tasks include several other subtasks that are omitted here. Step (2) is modeled as single task that invokes a forecast service that is hosted on an Apache Tomcat running on an Ubuntu Linux operating system. The complete application stack is hosted on their private Cloud environment. To analyze their business processes, WiFo has made use of existing approaches in the domain of Green Business Process Management [19]. As a result of the analysis, WiFo decided to apply the Green Compensation pattern [6] as they do not want to change their process from a functional perspective, i.e., they do not want to change the control flow or employed resources.

For the application of the selected Green Compensation pattern, WiFo needs to take care of the following tasks. First, the negative environmental impact (often described as CO_2 equivalents) that is induced by the process needs to be quantified. The impact of using IT services depends mainly on the electricity consumption of the services and their corresponding infrastructure. In Nowak et al. [2], we investigated the main drivers for electricity consumption of a computer system and found that CPU utilization is the most dominant one. The challenge here is to (i) identify the infrastructure and all components that are employed a business process and (ii) to analyze which components are responsible for a negative environmental impact. In addition, (iii) these steps must be automated as argued in Section II.A.

The second challenging task is improving the environmental impact of the infrastructure and the employed components. Naively, WiFo could simply change the resources they are using. However, as WiFo does not want to change the employed resources and their configurations, we need some external means to improve the environmental impact without changing the environment. As described in the Green Compensation pattern, it might be suitable to use external CO_2 compensation services that offset the CO_2 emissions by supporting various climate projects. Thus, a suitable service for CO_2 compensation has to be found and an appropriate invocation of this service in the existing infrastructure has to be integrated based on CPU utilization

However, the integration of the CO_2 compensation service is not trivial as the infrastructure must be accessed to set up the service invocation. This typically requires a lot of technical expertise, for example, the login to a VM via SSH and the installation of the corresponding components. In addition, a monitoring agent that keeps track of the CPU utilization needs to be installed, configured, and connected to determine the CO_2 compensation of a service based on its utilization [2]. As an additionally aspect, all these steps must be performed rapidly to avoid system downtime. In summary, without having detailed knowledge about these steps, applying the abstract Green Compensation pattern results in a major management issue that must be executed carefully to (i) avoid system downtime caused by technical errors or slow execution and (ii) donating too much money caused by a wrong configuration that leads to false (e.g., too high) estimated CO_2 emissions. These results comply with the two issues described in Section II.A. Applying patterns in the domain of Cloud application management must, therefore, tackle (i) the technical complexity of solution refinement and (ii) the automation of refinement and solution execution. In this paper, we show how this can be realized using our approach for the Green Compensation pattern.

III. AUTOMATION OF THE GREEN COMPENSATION PATTERN FOR VM-BASED BUSINESS PROCESSES

In this section, we present our approach to apply the Green Compensation pattern automatically to various kinds of VM-based applications that are hosted in the Cloud. We additionally show how the two requirements for applying patterns in the domain of Cloud Computing, as introduced in Section II.A, are addressed and how the approach can be used to automate the application of the Green Compensation pattern to our motivating scenario described in Section II.C. This section is structured as follows: in the next subsection III.A, we first describe how CO_2 emissions of Virtual Machines can be determined and compensated. In section III.B, we refine the abstract solution of the Green Compensation pattern based on these compensation mechanisms towards an application for Virtual Machine-based business processes hosted in a Cloud. In section III.C, we show how the refined solution can be automated using our Pattern-based Management Planlet Framework we developed in former work. The result of this approach is a refined and fully-automated Green Compensation pattern that can be applied to various Virtual Machine-based applications for compensating their CO_2 emission.

A. Determination and Compensation of CO_2 Emissions caused by Virtual Machines

To apply the Green Compensation pattern to Virtual Machine-based service of a business processes, it is vital to identify their environmental impact. In Nowak et al. [2], we presented an approach that allows determining the energy consumption of a Web Service. In a nutshell, we distribute the total energy consumption of a physical system between the different services running on that system. As a means for distributing the total energy consumption, we are using the CPU utilization that is allocatable to each service. In case of having multiple VMs running on a physical system, we can extend that approach transitively and use the CPU utilization of a VM to distribute the total energy consumption of the hypervisor between the VMs running on that hypervisor.

To ease the use of external compensation services, we translate the energy consumption of a service and its underlying VM to CO_2 equivalents. To calculate the corresponding CO_2 equivalent from the energy consumption, the individual energy mix of a server location needs to be considered. In Germany, the Federal Ministry for Environment [21] provided an average of 0,55kg CO_2 per KWh in 2010. In the US, the average was about 0,59kg CO_2 per KWh [22] in 2009. Based on that conversion, various compensation services can be used such as PrimaKlima [21] or CarbonFund [25]. The main objective of those services is to offset CO_2-emissions through, for example, reforestation, investments in renewable energy, or the purchase of carbon credits that prevents businesses from pollution. We employ such compensation services to compensate the determined CO_2 emission of the Virtual Machines for which the measured negative environmental impact shall be improved.

B. A Method to refine the Green Compensation Pattern

All Green Business Process Patterns documented in Nowak et al. [6], including the Green Compensation Pattern shown in Section II.C, describe solutions in a very abstracted fashion. In order to apply them to automated business processes and their underlying IT-infrastructure, the given solution must be refined, i.e., the solution must be described as management steps that have to be performed in order to apply the pattern in the context of and focus on Virtual Machines. If we recall the motivating scenario from Section II.C, WiFo wants to compensate the “Perform Forecast Calculation” activity of their business process. This activity invokes a Cloud service that is performing the actual calculation. To compensate the negative environmental impact of the underlying Virtual Machine that is hosting that service, we need to describe a refined solution towards this VM-based context that covers all steps to be performed.

The Green Compensation pattern ensures that the changes to be applied do not change the actual processes and employed resources (cf. Section II.B). Therefore, to instantiate the compensation activity that improves the negative environmental impact by compensating CO_2 emission, this must be considered and the process as well as the resources must not be changed. As a result, we need a means to integrate this compensation activity *transparently* to the process. Therefore, we only install a new monitoring component on the Virtual Machine and do not change the other components or configurations of the process.

Figure 3 shows an overview of the method we applied to refine the pattern as BPMN diagram [15]. In the first step, the Virtual Machines that are responsible for the negative environmental impact of an activity need to be identified. Using that information, the next step includes the installation of a monitoring agent. In the BMWi funded project “Migrate!” [24], we have developed such an agent that is based on *sigar* [23]. That agent is able to gather the relevant CPU utilization of a VM. Details on which other performance counters should be tracked and on what time interval can be set in the following “Configure Monitoring Agent task”. After choosing a Compensation Service, the Monitoring Agent needs to be connected to that service in order to initiate the compensation based on the actual environmental impact of the service. Here, different rules may be specified and configured depending on the business objectives. If WiFo wants to compensate, for example, 100 percent of their forecast service, the compensation service initiates a compensation for 100 percent of the total energy consumption of the service. In case the compensation should additionally be based on other aspects, like the number of invokes or the total environmental impact of WiFo, additional components need to be provided and integrated.

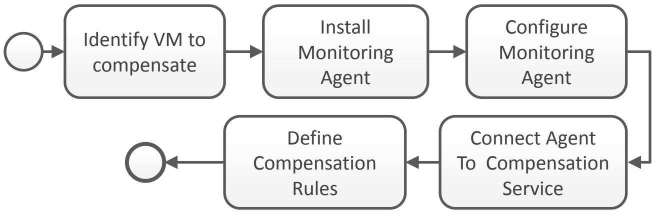

Identify VM to compensate
Install Monitoring Agent
Configure Monitoring Agent
Connect Agent To Compensation Service
Define Compensation Rules

Figure 3. Refined Green Compensation pattern solution for Virtual Machine-based business processes.

C. Employed Management Framework

The employed Management Planlet Framework [9][10][11][12] enables the automation of applying management patterns by generating declarative models that can be briefly

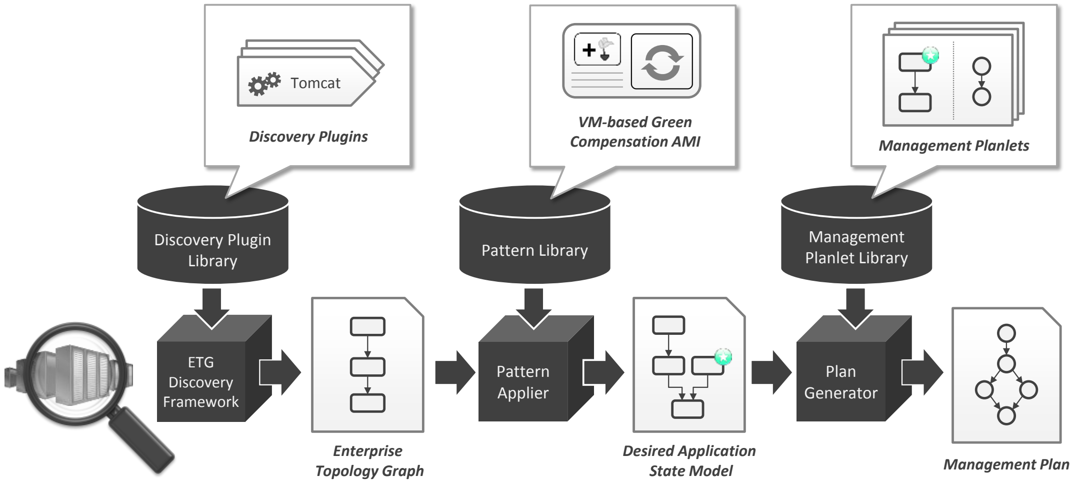


Figure 4. Overview of the Management Planlet Framework's pattern refinement and automation method (adapted from [9][12])

transformed fully automatically into executable workflows. In this section, we describe the management framework to provide required information. The framework's architecture and employed artifacts are shown in Figure 4.

We now describe how this framework can be used to apply a management pattern fully automatically to a running application. In the first step, the application's structure and all runtime information are captured as formal ETG. This model contains all nodes and relations of the process including runtime information in the form of properties, e.g., VM nodes provide IP-Address and SSH credentials. ETGs can be discovered fully automatically using the plugin-based ETG Discovery Framework [26]. Thus, this framework enables to automatically identify all VMs of a business process that are related to its environmental impact.

ETGs can then be transformed into *Desired Application State Models (DASMs)* through automatically applying management patterns. A DASM is a declarative description of management tasks to be performed on the application. It consists of (i) the application's ETG and (ii) one or more *Management Annotations*, which are declared on nodes and relations to specify management tasks to be executed on the associated element, e. g., that a given node shall be created, configured, or destroyed. Management Annotations define management tasks only declaratively, i.e., only *what* has to be done, but not *how*. Therefore, the framework employs a *Plan Generator* that consumes DASMs and generates executable workflows by orchestrating so called *Management Planlets*, which are small workflows that execute the Management Annotations declared on nodes and relations. Thus, they provide the *imperative* logic required to execute *declarative* Management Annotations in DASMs. Because the actual workflow generation is not important to understand the presented approach, we refer to Breitenbücher et al. [9][10][11][12] for more details and explanations.

To automatically transform ETGs into DASMs, the framework employs (i) *Semi-Automated Management Patterns (SAMPs)* and (ii) *Automated Management Idioms (AMIs)*, which both consume ETGs and output DASMs that describe the tasks to be performed. SAMPs implement the generic solution of high-level patterns and can be used to *semi-automate* the application of patterns. For example, the Green Compensation pattern is such a high-level pattern because it provides only generic information and requires additional manual refinement for its application. In contrast to this, an AMI provides a detailed refinement of a certain SAMP for a concrete use case. For example, the refinement of the Green Compensation pattern for Virtual Machines as shown in Figure 3 can be implemented as AMI that contains all required information to *fully automate* its application. Consequently, the output DASMs of SAMPs typically require additional manual refinement whereas the output DASMs of AMIs can be used directly to generate the workflows. Therefore, we realize the VM-based refinement of the Green Compensation pattern as fully automated AMI.

An AMI consists of two parts: (i) a *Topology Fragment* and (ii) a *Topology Transformation*. The fragment is a small segment of a topology used to check if the AMI is applicable to a certain ETG. Therefore, the fragment defines nodes and relations that must match elements in the ETG of the application in order to apply the pattern to the matching elements. The second part is a Topology Transformation that defines how to transform the input ETG to the DASM. Thus, it implements the AMI's refined solution by attaching Management Annotations to the affected nodes and relations. The resulting DASM then describes the management tasks to be performed on the application to apply the refined pattern.

D. VM-based Green Compensation AMI

In this section, we automate the refined Green Compensation pattern solution presented in Section X and show how the concept of AMIs can be used to implement a *VM-based Green Compensation AMI* that can be applied automatically to various VM-based business processes for compensating their CO_2 emissions. To create an AMI, the Topology Fragment must be defined first. In our scenario, this means that only a node of type *LinuxVM* must be defined as it (i) does not matter which underlying infrastructure is employed to host the VM, (ii) all Linux Virtual Machines enable remote access via SSH, and (iii) it does not matter which components are hosted on the Virtual Machine. Thus, the AMI's Topology Fragment is as simple as shown in Figure 5 on the left and defines that the AMI can be applied to all nodes in ETGs that are of type *LinuxVM*.

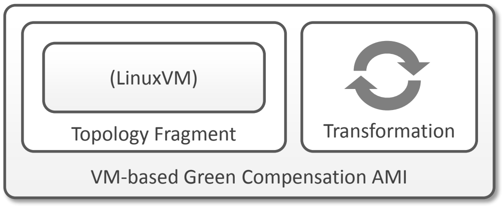


Figure 5. Automated Management Idiom (AMI) that refines the Green Compensation pattern for VM-based business processes

Secondly, the refined solution of the Green Compensation pattern shown in Figure 5 needs to be implemented as Topology Transformation. Therefore, the transformation has to modify the ETG for creating the DASM as follows: it has to (i) add a new node of type *Monitoring Agent* with attached *Create-Management Annotation* that defines that this node must be created, (ii) add a *hostedOn* relation from the Monitoring Agent node to the VM node including a Create-Management Annotation, (iii) add a CO_2 *Compensation Service* node without Management Annotation as this node already exists from third parties, and finally (iv) connect the Monitoring Agent to the Compensation Service node by a relation of type *uses* that must be created, too. Therefore, this relation has an attached Create-Annotation as well. To configure the Monitoring Agent node, an additional *Configure-Management Annotation* is attached to the Monitoring Agent node that defines that a configuration (also described in the Annotation) has to be set. This configuration can be implemented directly in the AMI.

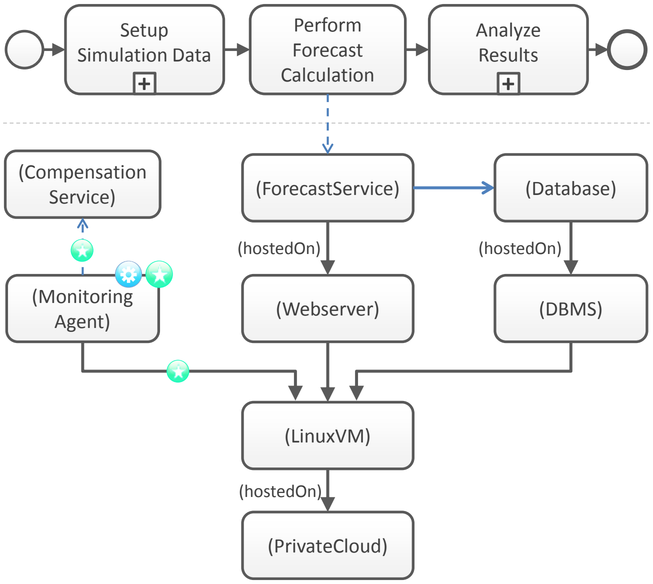


Figure 6. DASM resulting from applying the VM-based Green Compensation AMI to the motivating scenario

When this AMI is applied to a VM node contained in an ETG, the transformation will be applied and the corresponding DASM that can be used to generate the corresponding workflow is returned. Thus, the AMI is not bound to individual applications and can be applied generically to all business processes that contain Virtual Machine nodes in their ETG. Figure 6 shows the DASM that results from applying the described AMI to the motivating scenario. The green circles represent Create-Management Annotations while the blue circle represents the Configure-Management Annotation. The shown DASM defines exactly the management tasks to be executed for applying the pattern as described in the refined process shown in Figure 3. This DASM is then consumed by the framework's Plan Generator that searches and orchestrates appropriate Management Planlets implementing the required management logic to execute the management tasks described by the Management Annotations declared on nodes and relations in the DASM.

Consequently, this section showed how the requirements analyzed in Section II.A can be tackled for the automation of the Green Compensation pattern towards VM-based applications: first, the technical complexity is hidden completely by the AMI's transformation and the Management Planlets. Only the configuration of the monitoring agent may be up to the administrator, but this can be modeled also directly in the AMI's transformation if wanted. Secondly, the whole application of the pattern including its refinement for VM-based business processes is automated by the AMI. Thus, the result can be applied fully automatically to individual VMs to compensate their CO_2 emission. The implementation of our prototype proves the technical feasibility of the approach in general [9] [12].

However, the presented approach also requires significant effort that needs to be spent upfront. Automation is not free of charge but patterns need to be refined, AMIs need to be implemented as well as the corresponding Management Planlets. Whereas the definition of Topology Fragments is a rather trivial task the implementation of Topology Transformations may result in many unpredictable issues. The implementation of Management Planlets that capture these transformations require a lot of technical expertise regarding the employed technologies, e.g., connecting to a virtual machine using SSH and installing the required monitoring components is a non-trivial task due to firewalls and other security mechanisms. In addition, different structures of various kinds of VM-based applications must be considered in advance to provide a correct solution implementation. On the other hand the advantage is that these steps only have to be done once and the result can be used repeatedly. For example, Management Planlets can be used in various different automated management tasks and, therefore, the reusability aspect of this approach typically outvotes the effort that must be spent during the initial implementation. Moreover, the implementation of AMIs can be guided using a development method we presented in [27].

Besides the required effort and expertise the presented approach improves existing ones in the following aspects: (i) the generic Management Planlet Framework enables the integration of various kinds of proprietary and heterogeneous technologies seamlessly into the overall Management Plan generation [10]. The refined solution that is implemented in our AMI is independent of individual technologies. Thus, it does not matter if the virtual machine is hosted on a public Cloud, for example Amazon EC2, or if a physical server is employed in the local infrastructure of an enterprise. (ii) the refinement of the Green Compensation pattern for use in VM-based applications demonstrates how non-functional requirements of the optimization of applications can be automated based on patterns. This automation is vital in the domain of Cloud application management. Therefore, the approach shows how the concept of patterns can be implemented to consider ecological aspects in enterprises without requiring the deep-technical knowledge that is mandatory to refine patterns manually each time. (iii) the automated refinement of patterns supports *economics of management*. The implemented transformation routines, i.e., the AMIs and Planlets, are reusable for individual applications and, therefore, further effort for managing these applications is minimized to a simple AMI selection.

The actual CO_2 emission of an application that can be compensated mainly depends on the individual application. Therefore, we are not able to present some dedicated figures representing absolute savings or improvements. Moreover, the approach shows holistically how improvements may be realized when using Cloud-based applications.

IV. Related Work

The Green Compensation pattern we used in this work is only one possible pattern to improve the CO_2 emissions of enterprises. In Nowak et al. [6][7], we present a bunch of patterns that can help to improve the negative environmental impact of enterprises. Moreover, more and more companies have own strategies to improve CO_2 emissions. Amazon AWS, for example, has built datacenters in Oregon that only use green hydro-electric power [28].

Falkenthal et al. [14] present an approach that enables capturing directly reusable implementations of patterns by linking them to the patterns they originate from. In addition, they describe how such *Solution Implementations* can be aggregated using explicit operators. Therefore, this approach could be used to link the workflow that implements the VM-based solution refinement directly with the Green Compensation pattern. However, this does not tackle the issue considering the technical complexity described in Section II.A: during the workflow's implementation, the technical complexity of accessing the VM, installing and configuring the agent, etc. must be implemented manually. This requires a lot of technical expertise that is shifted in our approach completely to the employed management framework: implementing the AMI's transformation requires no technical knowledge about the underlying technologies.

Fehling et al. [30] present an approach that enables annotating so called *Implementation Artifacts* to architectural patterns that can be used during their application to instantiate the pattern's solution. These artifacts may range from software artifacts and configuration files to executable processes that implement certain management functionality of concrete components that can be used to implement the pattern. The annotated artifacts may support and guide applying and refining patterns. However, the approach supports only manual pattern application. In our work, we focus on fully automated pattern application. In a further work, Fehling et al. [29] present how the application of *Cloud Application Management Patterns* can be supported by providing abstract processes that define the high-level steps to perform the solution. These abstract processes must be then refined *manually* for individual use cases. Despite these management patterns are interrelated with their architectural patterns that may provide reusable management flows in the form of Implementation Artifacts, the refinement approach remains mainly manual: the artifacts may be used to ease implementing parts of the solution process, but their orchestration must be done manually.

In Breitenbücher et al. [12], we showed how the *Stateless Component Swapping Pattern*, which is a pattern that enables migrating an application component without downtime, can be automated using the Management Planlet Framework similarly as presented in this paper. Thus, the suitability of using the framework for automating patterns is technically feasible in general. Therefore, the findings of Breitenbücher et al. [12] and this paper provide the basis for continuing the automation of patterns following this concept.

The original Management Planlet Framework [9] is extended in several publications. In Breitenbücher et al. [11], we showed how management tasks can be annotated with policies that define non-functional requirements on the execution of the task, e.g., security requirements such as the location of a Virtual Machine. In Breitenbücher et al. [10], we presented an approach to specify and integrate imperative logic into the declarative concept of DASMs. Thus, these extensions show that the Management Planlet Framework provides a flexible means for automating pattern.

V. Conclusion

In this paper, we addressed the gap between the abstract description of pattern solutions and their application to individual Cloud-based business processes. As the manual application of a pattern is typically very complex, time consuming, and error-prone we proposed an approach that is able to apply the Green Compensation pattern to individual use cases automatically over and over again. Therefore, we first described a method on how to refine the Green Compensation pattern to represent all necessary management tasks for its application. Next, we showed how the employed Management Planlet Framework is used to implement that concrete and detailed pattern solution in a reusable way.

In our future work, we (i) plan to extend that approach to support further Green Business Process Management patterns and (ii) evaluate the refinement of patterns from other domains to identify possible extensions.

Acknowlegments

This work was partially funded by the BMWi projects CloudCycle (01MD11023) and Migrate! (01ME11055).

REFERENCES

[1] A. Nowak, F. Leymann, D. Schumm, and B. Wetzstein, "An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering" ICT-GLOW 2011, Springer, 2011, pp. 150-164.

[2] A. Nowak, T. Binz, F. Leymann, and N. Urbach "Determining Power Consumption of Business Processes and their Activities to Enable Green Business Process Reengineering" EDOC 2013, IEEE, 2013, pp. 259-266.

[3] GreenBiz.com. How Will Sustainability Change at Your Company in 2011. http://www.greenbiz.com/blog/2010/12/29/how-will-sustainability-change-yourcompany-2011, retrieved 05.2014.

[4] Gartner Research. Gartner Identifies the Top 10 Strategic Technologies for 2010. http://www.gartner.com/it/page.jsp?id=1210613, retrieved 05.2014.

[5] Intl. Org. for Standardization. ISO14000. Environmental Management. http://www.iso.org/iso/iso14000, retrieved 05.2014.

[6] A. Nowak, F. Leymann, D. Schleicher, D. Schumm, and S. Wagner, "Green Business Process Patterns" PLoP 2011, ACM, 2011, in press.

[7] A. Nowak and F. Leymann, "Green Business Process Patterns - Part II" SOCA 2013, IEEE, 2013, pp. 168-173.

[8] D. Oppenheimer, A. Ganapathi, and D. Patterson, "Why do internet services fail, and what can be done about it?" USENIX 2003, ACM, 2003.

[9] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Pattern-based Runtime Management of Composite Cloud Applications" CLOSER 2013, SciTePress, 2013, pp. 475-482.

[10] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and J. Wettinger, "Integrated Cloud Application Provisioning: Interconnecting Service-Centric and Script-Centric Management Technologies" CoopIS 2013, Springer, 2013, pp. 130-148.

[11] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning of Cloud Applications" SECURWARE 2013, IARIA Xpert Publishing Services, 2013, pp. 86-95.

[12] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Automating Cloud Application Management Using Management Idioms" PATTERNS 2014, IARIA Xpert Publishing Services, 2014, pp. 60-69.

[13] C. Alexander, A Pattern Language. Towns, Buildings, Construction. Oxford University Press, New York, 1977.

[14] M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, and F. Leymann, "From Pattern Languages to Solution Implementations" PATTERNS 2014, Xpert, 2014, in press.

[15] OMG, Business Process Model and Notation (BPMN), Version 2.0.

[16] JET Tankstellen Deutschland GmbH. www.arktik.de, retrieved 05.2014.

[17] Deutsche Bahn AG. http://www.dbecoprogram.com /index.php?lang=en, retrieved 05.2014.

[18] MyClimate. http://de.myclimate.org, retrieved 05.2014.

[19] A. Nowak, F. Leymann, and D. Schumm, "The Differences and Commonalities between Green and Conventional Business Process Management" CGC 2011, IEEE, 2011, pp. 569 - 576.

[20] T. Binz, C. Fehling, F. Leymann, A. Nowak, and D. Schumm, "Formalizing the Cloud through Enterprise Topology Graphs" CLOUD 2012, IEEE, 2012, pp. 742 - 749.

[21] PrimaKlima. http://www.prima-klima-weltweit.de/co2/kompens -berechnen.php, retrieved 05.2014.

[22] EPA. http://www.epa.gov/cleanenergy/documents/egridzips/eGRID 2012V1_0_year09_SummaryTables.pdf, retrieved 05.2014.

[23] Sigar. http://www.hyperic.com/products/sigar, retrieved 05.2014.

[24] Migrate!. http://www.migrate-it2green.de/, retrieved 05.2014.

[25] CarbonFund. http://www.carbonfund.org/, retrieved 05.2014.

[26] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "Automated Discovery and Maintenance of Enterprise Topology Graphs" SOCA 2013, IEEE, 2013, pp. 126 - 134.

[27] U. Breitenbücher, T. Binz, and F. Leymann, "A Method to Automate Cloud Application Management Patterns", ADVCOMP 2014, IARIA Xpert Publishing Services, 2014, in press.

[28] R. Miller. Amazon's Cloud goes Modular in Oregon. http://www.datacenterknowledge.com/archives/2011/03/28/amazons-cloud-goes-modular-in-oregon/, retrieved 05.2014.

[29] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, "Pattern-Based Development and Management of Cloud Applications" Future Internet 2012, 4(1), pp. 110-141, 2012.

[30] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An Architectural Pattern Language of Cloud-based Applications" PLoP 2011, ACM, 2011, in press.

A Method to Automate Cloud Application Management Patterns

Uwe Breitenbücher, Tobias Binz, Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart, Stuttgart, Germany
{breitenbuecher, lastname}@iaas.uni-stuttgart.de

***Abstract*—Management patterns are a well-established concept to document reusable solutions for recurring application management issues in a certain context. Their generic nature provides a powerful means to describe application management knowledge in an abstract fashion that can be refined for individual use cases manually. However, manual refinement of abstract management patterns for concrete applications prevents applying the concept of patterns efficiently to the domain of Cloud Computing, which requires a fast and immediate execution of arising management tasks. Thus, the application of management patterns must be automated to fulfill these requirements. In this paper, we present a method that guides the automation of Cloud Application Management Patterns using the Management Planlet Framework, which enables applying them fully automatically to individual running applications. We explain how existing management patterns can be implemented as Automated Management Patterns and show how these implementations can be tested afterwards to ensure their correctness. To validate the approach, we conduct a detailed case study on a real migration scenario.**

Keywords*—*Application Management; Cloud Computing; Management Patterns; Management Automation.

I. Introduction

Management patterns are a well-established concept to document reusable solution expertise for frequently recurring application management problems in a certain context [1]. They provide the basis for the implementation of management processes and influence the architecture and design of applications. The generic nature of patterns enables management experts to document knowledge about proven solutions for challenging management issues in an abstract, structured, and reusable fashion. This supports application managers in solving concrete instances of the general problem. Applying management patterns, e. g., to scale or to migrate application components, to concrete real use cases in the form of running applications requires, therefore, typically a *manual refinement* of the pattern's abstract high-level solution towards the individual use case [2]. However, the manual refinement and application of management patterns is time-consuming and, therefore, not appropriate in the domain of Cloud Computing since the immediate and fast execution of arising management tasks is of vital importance to achieve Cloud properties such as pay-as-you-go pricing models and on-demand computing [1]. This is additionally underscored by the fact that human errors are the largest cause of failures of internet services and large systems [3][4]. Especially the rapid evolution of management technologies additionally strengthens this effect: complex management tasks can be executed much easier and quicker due to powerful management interfaces offered by Cloud providers that abstract from technical details. However, this increases the probability of human errors because there is hardly any notion of the underlying physical infrastructure and the actual impact the executed tasks may have [1]. As a consequence, to use the concept of patterns efficiently in the domain of *Cloud Application Management*, the (i) refinement of management patterns for individual use cases as well as (ii) the execution of the refined solution must be *automated* since manual realizations are too slow, costly, and error prone [2].

However, the difficulties of automating management patterns are manifold. Especially the immense technical expertise required to refine a pattern's abstract solution towards a concrete use case is one of the biggest challenges in terms of automation. To tackle these issues, we presented the pattern-based *Management Planlet Framework* in former works [2][5][6][7][8], which enables applying management patterns automatically to concrete running applications for executing typical management tasks such as migrating applications or updating components without downtime [2]. The framework employs so called *Automated Management Patterns*, which implement a certain management pattern in a way that enables its application to various individual use cases either semi-automatically or even fully-automatically. However, the implementation of these automated patterns is a non-trivial task that requires special attention to ensure a high quality and correctness of their automated executions on real applications. This issue is tackled in this paper. We present a method that enables automating *Cloud Application Management Patterns* using the Management Planlet Framework introduced above. We show how management patterns described in natural text can be analyzed and implemented in a generic way that enables applying the captured solution logic automatically to concrete use cases in the form of running applications—independently from individual manifestations. To guide this analysis, the method describes how the relevant information required to automate a management pattern can be extracted from its textual description. The presented method can be used to automate various kinds of management patterns, which enables applying this concept efficiently in the domain of Cloud Application Management. We prove the feasibility of our approach by a detailed case study that considers the automation of an existing migration pattern and various applications of the presented method to automate other management patterns.

The remainder of this paper is structured as follows: in Section II, we explain the employed Management Planlet Framework, which provides a generic means to automatically apply management patterns to individual applications. Section III presents the main contribution of this paper in the form of a method to automate existing Cloud Application Management Patterns using the employed Management Planlet Framework. We conduct a detailed case study in Section IV to illustrate how the method can be applied to automate an existing migration pattern. Section V discusses related work. Section VI concludes the paper and provides an outlook on planned future work.

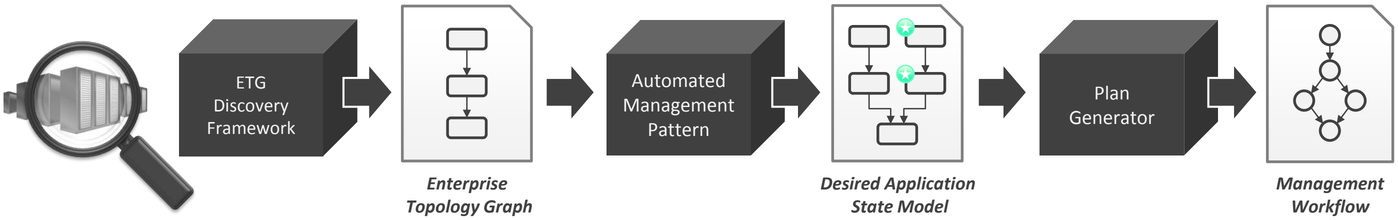


Figure 1. Architecture and concept of the Management Planlet Framework (adapted from [2][6][7][8]).

II. Employed Management Framework

In this section, we present the employed *Management Planlet Framework* [2][5][6][7][8] that provides the basis for automating management patterns. The framework can be used for the (i) initial provisioning of applications [7] as well as (ii) for runtime management of applications [6], which is the focus of this paper. The framework's management approach is shown in Figure 1 and can be summarized as follows: first, the application to be managed is captured by a formal model called *Enterprise Topology Graph*. In the second step, an *Automated Management Pattern* is applied that transforms this Enterprise Topology Graph into a *Desired Application State Model*, which declaratively specifies the management tasks to be performed. This DASM is then transformed into an executable *Management Workflow* by a *Plan Generator* in the last step. In the following, we explain these steps and the involved artifacts in detail.

A. Enterprise Topology Graph (ETG)

An Enterprise Topology Graph (ETG) [9] is a formal model that describes the current structure of a running application including its state. ETGs are modelled as directed graphs that consist of nodes and relations (*elements*) representing the application's components and dependencies. Each element has a certain type and provides properties that capture runtime information. For example, a node may be of type "VirtualMachine" and provides properties such as its "IP-Address". ETGs can be discovered fully automatically using the ETG *Discovery Framework* [10]. This framework only requires an entry point of the application, e. g., the URL of an application's Web frontend, to discover the whole ETG fully automatically including all software and infrastructure components of the application.

B. Automated Management Patterns (AMP)

An Automated Management Pattern (AMP) is a generic implementation of a management pattern that can be applied automatically to individual applications that match a predefined application structure, e. g., to migrate an application without downtime. An AMP consumes the ETG of the application to which the implemented pattern shall be applied and automatically specifies the management tasks that have to be performed on the application's nodes and relations. Therefore, AMPs consist of two parts: the (i) *Topology Fragment* describes the application structure to which an AMP can be applied, i. e., it models the nodes and relations that must match elements in an ETG to apply the pattern to these matching elements. The (ii) *Topology Transformation* consumes the application's ETG and automatically creates a *Desired Application State Model*, in which it specifies the management tasks to be performed in the form of abstract *Management Annotations* that are declared by the transformation on nodes and relations of the ETG.

C. Desired Application State Model (DASM)

A Desired Application State Model (DASM) describes management tasks to be performed on nodes and relations of a running application in a *declarative* manner. It consists of (i) the application's ETG and (ii) *Management Annotations*, which are declared on nodes or relations of the ETG. A Management Annotation (depicted as coloured circle) specifies a small management task to be executed on the associated element, but defines only the *abstract semantics* of the task, e. g., that a node shall be created, but not its technical realization. For example, a *Create-Annotation* attached to a "MySQLDatabase" that has a "hostedOn" relation to an "UbuntuVM" means that the database shall be installed on the VM. Similarly, there are annotations that specify specific management tasks, e. g., an *ImportData-Annotation* attached to a database defines that data has to be imported. Management Annotations may additionally define that they must be executed before, after, or concurrently with another annotation. Due to the declarative nature of DASMs, only the *what* is described, but not the *how*. Thus, in contrast to imperative descriptions such as executable workflows [11] that define all technical details, DASMs can not be executed directly and are transformed into workflows by the *Plan Generator*.

D. Management Planlets & Plan Generator

In the last step, the created DASM is automatically transformed into an executable Management Workflow. This is done by the framework's Plan Generator, which orchestrates so called *Management Planlets*. A Management Planlet is a small workflow that executes one or more Management Annotations on a certain combination of nodes and relations. For example, a Planlet may deploy a Java application on a Tomcat Webserver. The Plan Generator tries to find a suitable Management Planlet for each Management Annotation specified in the DASM that executes the corresponding management task. Thus, Management Planlets are reusable building blocks that provide the low-level imperative management logic to execute the declarative Management Annotations declared in DASMs.

E. Development of Automated Management Patterns

DASMs provide the basis to implement AMPs on a high-level of abstraction: management tasks need to be specified only abstractly in the form of declarative Management Annotations without the need to deal with the complex, low-level, and technical issues required for their execution. These technical details are considered only by the responsible Management Planlets. However, since management patterns typically capture multiple steps to be performed, the development of AMPs is a challenging task and needs careful consideration. Therefore, we present a method that guides the development of AMPs.

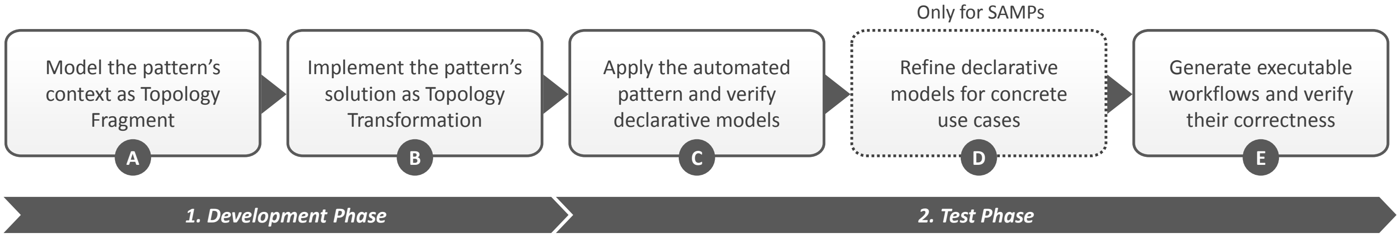

Only for SAMPs
Model the pattern's context as Topology Fragment
A
Implement the pattern's solution as Topology Transformation
B
Apply the automated pattern and verify declarative models
C
Refine declarative models for concrete use cases
D
Generate executable workflows and verify their correctness
E
1. Development Phase
2. Test Phase

Figure 2. Method to automate existing Cloud Application Management Patterns.

III. A Method to Automate Cloud Application Management Patterns

In this section, we present a method to automate existing management patterns through implementing them as AMPs. We distinguish here between two kinds of AMPs: *Semi-Automated Management Patterns* (SAMPs) [6] implement patterns on an abstract level, e. g., to migrate any kind of application to another location. Therefore, applying SAMPs typically requires a manual refinement of the resulting DASMs before they can be transformed into the corresponding workflows, e. g., abstract nodes types must be replaced, additional nodes or relations have to be inserted, or further Management Annotations have to be added. An *Automated Management Idiom* (AMI) [2] implements a refined version of a management pattern for a concrete use case, e. g., to migrate a Java application hosted on an Apache Tomcat Webserver to the Amazon Cloud. As a consequence, applying AMIs results in already refined DASMs that can be translated directly into workflows. The method is shown in Figure 2 and consists of two phases: in the (i) *Development Phase*, the management pattern to be automated is analyzed and implemented as SAMP or AMI. In the (ii) *Test Phase*, the implementation is verified for correctness. In the following subsections, we explain the five steps of the method in detail.

A. Model the pattern's context as Topology Fragment

Analyze the management pattern's context with focus on the application structures to which the pattern can be applied and model this information as Topology Fragment.

Cloud Application Management Patterns typically consist of several parts that describe the pattern in natural text [1]. In the method's first step, the textual description of the pattern to be automated is analyzed in terms of the application structures to which the AMP shall be applicable. This information is then modelled as Topology Fragment. If a concrete AMI shall be created, the fragment typically needs to be refined for the respective use case. For example, the aforementioned refinement of the pattern that migrates a Java application to the Amazon Cloud results in a Topology Fragment that models a node of type "JavaApplication" that has a relation of type "hostedOn" to a node of type "ApacheTomcat". This AMI is then applicable to all ETGs that contain this combination of elements. Thus, as the Topology Fragment provides the basis for matchmaking SAMPs and AMIs with ETGs, it must define exactly the nodes and relations to which the automated pattern is applicable. This kind of information is typically described in the *context* section of management patterns, but also other sections such as *problem* or even the *solution* can be used to extract this information.

B. Implement the pattern's solution as Topology Transformation

Analyze the management pattern's solution and implement the described management logic as Topology Transformation.

In the second step, the pattern's solution logic is captured in a way that enables its automated application to individual applications. Therefore, the textual description of the pattern's solution is analyzed in terms of the management tasks that have to be executed to apply the pattern. The analyzed procedure is then implemented as Topology Transformation that acts on the nodes and relations defined by the Topology Fragment: the Topology Transformation must declare exactly the Management Annotations on the ETG that declaratively specify the analyzed management tasks to be executed following the pattern's solution. In case of automating a management pattern as abstract SAMP, the Topology Transformation implements only the pattern's original abstract solution logic and remains rather vague: executing this transformation on an ETG typically results in a DASM that additionally needs to be refined manually afterwards. If the pattern is automated as detailed AMI for a concrete use case, the abstract solution logic must be first (i) refined towards this use case in order to provide detailed information about the management tasks to be executed. These are (ii) implemented afterwards in the Topology Transformation that specifies the corresponding Management Annotations. Executing such fine-grained AMI-transformations typically results in fully refined DASMs that can be transformed directly into executable workflows without further manual effort.

C. Apply the automated pattern and verify declarative models

Apply the created SAMP / AMI to concrete use cases and compare the resulting DASMs with the solution described by the original pattern or refinement to verify their correctness.

After the textual description of the pattern is translated into its corresponding SAMP / AMI, the correctness of the realization needs to be verified. Therefore, in the first part of the *Test Phase*, the implemented patterns are tested against various concrete use cases, i. e., ETGs of different running applications. First, a set of appropriate use cases must be identified that captures all possible application structures to which the pattern can be applied and that are affected by the pattern's transformation. This requires an explicit and careful analysis of the pattern's Topology Fragment and Topology Transformation to cover all possible scenarios. Additional use cases must be identified to which the pattern can *not* be applied to additionally ensure the correctness of the pattern's Topology Fragment. Afterwards, the pattern is tested against these cases. The test is subdivided into two steps: (i) testing the Topology Fragment and (ii) testing the Topology Transformation. In the first step, the automated pattern is applied to different use cases which are not all suited for the pattern. Thus, some use cases match with the Topology Fragment, others not. The results of these matchmakings are then compared with the semantics of the original pattern or refinement, respectively, to verify the correct modelling of the Topology Fragment: the Automated Management Pattern must be applicable exactly to the same use cases as the original pattern or the refinement, respectively. In the second step, the Topology Transformation is executed on the correctly matching use cases. The resulting DASMs are then compared with (i) the *solution* section of the original pattern / refinement to verify if the declaratively specified tasks comply with the description and (ii) the *result* section to verify that the results are equal. This is possible as the created DASMs contain information about both solution and result: they describe the management tasks to be executed as well as the final application structure and partially the application's state after executing the workflow.

D. Refine declarative models for concrete use cases

Only for creating SAMPs: refine the resulting DASMs for concrete use cases in order to provide all missing information required to generate executable workflows.

If a pattern is semi-automated as SAMP, the DASMs resulting from the previous steps cannot be translated directly into executable workflows as the refinement is missing: the SAMP's Topology Transformation implements only the pattern's abstract solution and provides not all information required to generate the workflow. As a result, the resulting DASMs must be refined manually in this step for providing all required information. DASMs resulting from the application of AMIs are not affected.

E. Generate executable workflows and verify their correctness

Transform the final DASMs into the corresponding workflows, compare them with the solution described by the original pattern or refinement, and verify the result of their execution.

In the last step, the final DASMs resulting from the previous steps are transformed into the corresponding executable Management Workflows using the framework's Plan Generator. Then, the correctness of these imperative management description models are verified by two manual steps: (i) verifying the correctness of the generated workflow implementations and (ii) verifying the correctness of the final application states after executing the Management Workflows. In the first step, the implementation of the generated workflows are compared with the abstract *solution* of the pattern or its refined incarnation if the tested pattern is implemented as AMI. This last verification ensures that the finally executed management tasks including all technical details are correct. In particular, this step is required to ensure that the employed Management Annotations lead to correct workflows, e. g., that there are Management Planlets available to execute all the Management Annotations declared in the DASMs. In the second verification step, the generated Management Workflows are executed on the real running test applications and the results are compared with the *result* section of the original management pattern or the refined result if the tested automated pattern is implemented as AMI.

IV. Case Study and Validation

In this section, we validate the approach by a detailed case study that considers the automation of an existing management pattern using the proposed method. Due to the important issue of vendor lock-in in the domain of Cloud Computing, we automate a migration management pattern. First, we describe the most important facts of the original pattern and derive a refined idiom afterwards that enables migrating Java-based applications to the Amazon Cloud. The refined idiom is then implemented as Automated Management Idiom using the presented method.

The pattern to be automated is called *Stateless Component Swapping Pattern* [12] and originates from the Cloud Computing pattern language developed by Fehling et al. [1][12][13]. The pattern deals with the problem *"How can stateless application components that must not experience downtime be migrated?"*. The context observed is that for many business applications downtime is unacceptable, e. g., for customer-facing applications. Hence, its intent is migrating stateless applications from one environment into another transparently to the accessing human users or other applications. Therefore, the stateless application is active in both environments concurrently during the migration to avoid downtime. Here, "stateless" means that the application does not handle internal session state [12].

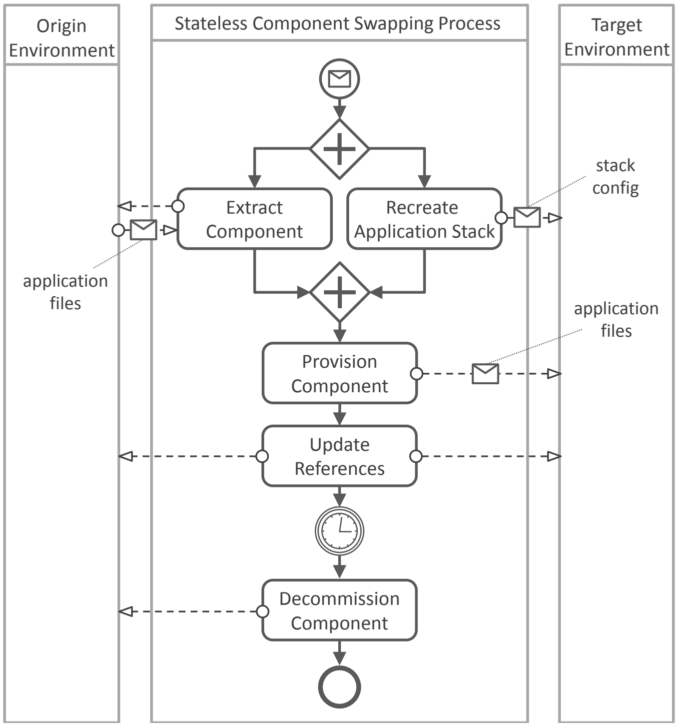


Figure 3. Abstract Stateless Component Swapping Process (adapted from [12]).

Figure 3 describes the pattern's solution as Business Process Model and Notation (BPMN) [14] diagram: first, the component to be migrated is extracted from the origin environment while the required application stack is provisioned concurrently in the target environment. After the new application stack is provisioned, a new instance of the component is deployed thereon while the original component is still active. When deployment has finished and the new component is running, all references pointing to the old component are updated to the new one and the original component gets decommissioned.

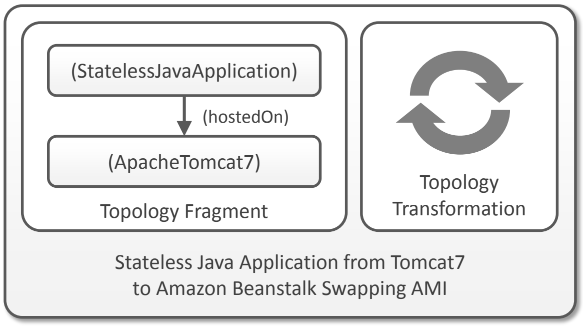

(StatelessJavaApplication)
(hostedOn)
(ApacheTomcat7)
Topology Fragment
Topology
Transformation
Stateless Java Application from Tomcat7
to Amazon Beanstalk Swapping AMI

Figure 4. Automated Management Idiom.

Since this case study aims at fully automating this management pattern, we implement it as Automated Management Idiom using the presented method. Therefore, we have to refine the pattern towards a more specific context to which it can be applied fully automatically in a generic manner, i. e., all applications that correspond to this context can be managed by applying the created AMI. In this case study, the refined context is defined by migrating a stateless Java application that is hosted on an Apache Tomcat 7 Webserver to Amazon's Cloud offering "Beanstalk", which provides a platform service (PaaS) for directly hosting Java applications. The pattern's core intent of migrating the application without downtime remains.

We now apply the presented method to automate the pattern for this refined context. In the first step, the kinds of applications to which the AMI can be applied must be defined. Therefore, we analyze the description of the pattern and the described refinement to model the Topology Fragment shown in Figure 4; according to the description, the automated pattern can be applied to all nodes of type "StatelessJavaApplication" that are connected by a relation of type "hostedOn" with a node of type "ApacheTomcat7". In the second step, we implement the pattern's solution as executable Topology Transformation. We first have to refine the pattern's abstract solution to the concrete use case considered in this study. Therefore, we (i) analyze the abstract solution process shown in Figure 3, (ii) transfer and refine the described information to our Java-based migration use case, and (iii) implement the Topology Transformation accordingly. We now step through this process and transfer each activity into our transformation implementation in consideration of the refinement. The transformation described in the following is implemented in a generic manner. It acts exclusively on the nodes and relations defined by the Topology Fragment or applies generic transformation rules that are not bound to a particular application. Therefore, it can be executed on all applications that match the pattern's Topology Fragment defined above. We explain the transformation directly on a real scenario that is depicted in Figure 5; on the left, there is the current ETG of a Java-based application that runs on a Apache Tomcat 7 installation hosted on a local physical server. The application implements a stateless Webservice that is publicly reachable via an internet domain. This Webservice shall be migrated to Amazon Beanstalk. As the defined Topology Fragment shown in Figure 4 matches this ETG, our refined pattern can be applied. All nodes and relations that are surrounded by dotted lines and Management Annotations are inserted by the transformation. The numbers in white circles represent the transformation steps.

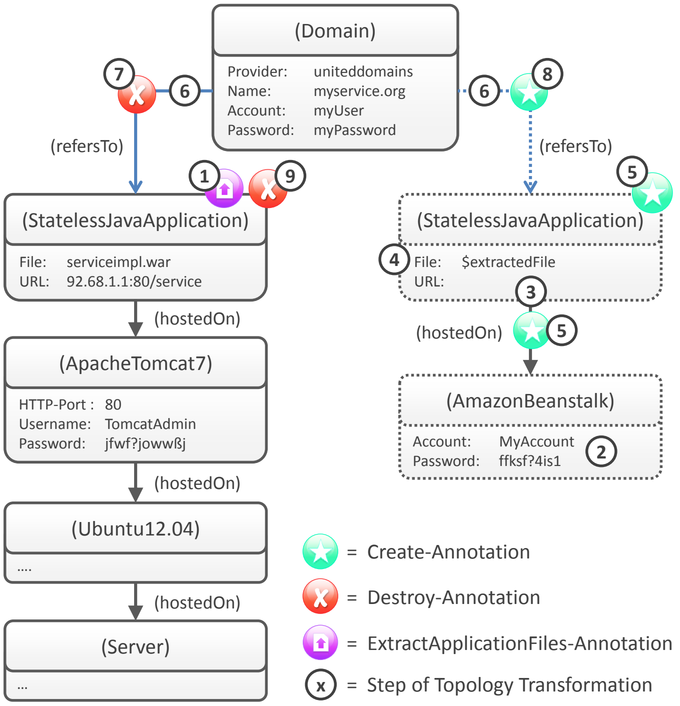


Figure 5. DASM that results from applying the created AMI.

First, the component to be migrated has to be extracted and the new application stack needs to be created. Therefore, the transformation (1) attaches a *ExtractApplicationFiles-Annotation* to the "StatelessJavaApplication" node and (2) inserts a new node of type "AmazonBeanstalk" to the DASM. As the Beanstalk node provides "Account" and "Password", the transformation requests these properties as input parameters and writes them directly to the node. We need no Management Annotations on this node since the Beanstalk service is always running. The *ExtractApplicationFiles-Annotation* is configured to export the Java files of the application to a location that is stored in a variable "$extractedFile", which is used in the next step. Afterwards, the transformation (3) inserts a new node of type "StatelessJavaApplication" and a "hostedOn" relation to the Beanstalk node, (4) specifies the files to be deployed using the "$extractedFile" variable, and (5) attaches *Create-Annotations* to the new node and the new relation. To update the references of the old component, we first (6) copy all incoming and outgoing relations except its "hostedOn" relation and replace the old component node by the new node, (7) attach *Destroy-Annotations* to the old relations, and (8) attach *Create-Annotations* to the new relations. Then, the transformation (9) attaches a *Destroy-Annotation* to the old component that specifies to undeploy the application from the local Webserver. To avoid downtime, the execution order of some annotations must be defined, e. g., creating and destroying the "refersTo" relations must be done concurrently by one planlet whereas the old component must not be decommissioned before the new one is ready. These orders are specified in the last step. The resulting DASM is complete and ready to be translated into the corresponding executable Management Workflow. In the following Test Phase, similar use cases are taken and the AMI gets applied to them. The resulting DASMs, as well as the generated workflows, are then analyzed for correctness.

V. Related Work

We applied the method to automate several management patterns, e. g., the *Stateless Component Swapping Pattern* [12] and the *Update Transition Process Pattern* [13] (published in Breitenbücher et al. [2][8]). In Nowak et al. [15], we automated the *Green Compensation Pattern* [16] to reduce the CO_2 emission of virtual machine-based business processes.

Fehling et al. [17][18] present a (i) step-by-step process for the traceable identification of Cloud patterns, a (ii) pattern format, and a (iii) pattern authoring toolkit that can be used to support the identification process. Despite these works focus mainly on Cloud architecture patterns, the core concepts can be adapted and used to create management patterns and idioms that can be automated afterwards using the presented method. Fehling et al. [17] also show how the identified architectural Cloud patterns can be applied using an existing provisioning tool. However, they consider only application provisioning and do not consider the automation of management patterns.

Reiners et al. [19] present an iterative pattern formulation approach for developing patterns. They aim for documenting and developing knowledge from the very beginning and continuously developing findings further. From this perspective, patterns are not just final artifacts but are developed based on initial non-validated ideas. They have to pass different phases until they become approved patterns. In contrast to our work that focuses on automating patterns, this iterative pattern formulation approach can be used to develop management patterns that capture problem and solution in natural text. Thus, the approaches are complementary: the formulation approach can be used to create management patterns that are automated afterwards using our method. Our pattern automation helps testing the captured knowledge in each phase of the iterative process to validate the pattern's correctness and suitability.

Falkenthal et al. [20] present an approach that enables reusing concrete implementations of patterns by attaching them as so called *Solution Implementations* directly to the patterns they originate from. The approach can be used to create workflows that implement a management patterns solution for a certain use case as Solution Implementation that is linked with the original pattern. However, a manual implementation of the corresponding workflows requires a lot of management expertise for handling the technical complexity of refinement [2]. In addition, such management workflows are typically tightly coupled to particular application structures and are not able to provide the flexibility of Topology Transformations that may analyze the whole application topology to ensure a correct specification of the Management Annotations to be performed.

VI. Conclusion and Future Work

In this paper, we presented a method that enables automating existing management patterns using the Management Planlet Framework. We showed that the method enables analyzing and implementing existing management patterns in a generically applicable fashion. The method provides a structured means to create and test Automated Management Patterns that guides developers in transforming natural text into automated routines. To validate the presented approach, we conducted a detailed case study that shows how a Cloud Application Management Pattern for application migration can be automated using our method. In future work, we plan to investigate how the method can be used to automate architectural patterns to support the development and initial provisioning of Cloud applications, too.

Acknowledgment

This work was partially funded by the BMWi project Cloud-Cycle (01MD11023).

References

[1] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, "Pattern-Based Development and Management of Cloud Applications," *Future Internet*, vol. 4, no. 1, pp. 110–141, March 2012.

[2] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Automating Cloud Application Management Using Management Idioms," in *PATTERNS 2014*. IARIA Xpert Publishing Services, May 2014, pp. 60–69.

[3] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do internet services fail, and what can be done about it?" in *USITS 2003*. USENIX Association, June 2003, pp. 1–16.

[4] A. B. Brown and D. A. Patterson, "To Err is Human," in *EASY 2001*, July 2001, p. 5.

[5] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and J. Wettinger, "Integrated Cloud Application Provisioning: Interconnecting Service-Centric and Script-Centric Management Technologies," in *CoopIS 2013*. Springer, September 2013, pp. 130–148.

[6] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Pattern-based Runtime Management of Composite Cloud Applications," in *CLOSER 2013*. SciTePress, May 2013, pp. 475–482.

[7] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning of Cloud Applications," in *SECURWARE 2013*. Xpert Publishing Services, August 2013, pp. 86–95.

[8] U. Breitenbücher *et al.*, "Policy-Aware Provisioning and Management of Cloud Applications," *International Journal On Advances in Security*, vol. 7, no. 1&2, 2014.

[9] T. Binz, C. Fehling, F. Leymann, A. Nowak, and D. Schumm, "Formalizing the Cloud through Enterprise Topology Graphs," in *CLOUD 2012*. IEEE, June 2012, pp. 742–749.

[10] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "Automated Discovery and Maintenance of Enterprise Topology Graphs," in *SOCA 2013*. IEEE, December 2013, pp. 126–134.

[11] F. Leymann and D. Roller, *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, 2000.

[12] C. Fehling, F. Leymann, S. T. Ruehl, M. Rudek, and S. Verclas, "Service Migration Patterns - Decision Support and Best Practices for the Migration of Existing Service-based Applications to Cloud Environments," in *SOCA 2013*. IEEE, December 2013, pp. 9–16.

[13] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer, January 2014.

[14] OMG, *Business Process Model and Notation (BPMN), Version 2.0*, Object Management Group Std., Rev. 2.0, January 2011.

[15] A. Nowak, U. Breitenbücher, and F. Leymann, "Automating Green Patterns to Compensate CO_2 Emissions of Cloud-based Business Processes," in *ADVCOMP 2014*. IARIA Xpert Publishing Services, August 2014.

[16] A. Nowak, F. Leymann, D. Schleicher, D. Schumm, and S. Wagner, "Green Business Process Patterns," in *PLoP 2011*. ACM, October 2011.

[17] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An Architectural Pattern Language of Cloud-based Applications," in *PLoP 2011*. ACM, October 2011.

[18] C. Fehling, T. Ewald, F. Leymann, M. Pauly, J. Rütschlin, and D. Schumm, "Capturing Cloud Computing Knowledge and Experience in Patterns," in *CLOUD 2012*. IEEE, June 2012, pp. 726–733.

[19] R. Reiners, "A Pattern Evolution Process - From Ideas to Patterns," in *Informatiktage 2012*. GI, March 2012, pp. 115–118.

[20] M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, and F. Leymann, "From Pattern Languages to Solution Implementations," in *PATTERNS 2014*. IARIA Xpert Publishing Services, May 2014, pp. 12–21.