



FUTURE COMPUTING 2015

The Seventh International Conference on Future Computational Technologies and
Applications

ISBN: 978-1-61208-389-6

March 22 - 27, 2015

Nice, France

FUTURE COMPUTING 2015 Editors

Kendall E. Nygard, North Dakota State University - Fargo, USA

Dietmar Fey, Friedrich-Alexander-University Erlangen-Nürnberg, Germany

FUTURE COMPUTING 2015

Forward

The Seventh International Conference on Future Computational Technologies and Applications (FUTURE COMPUTING 2015), held between March 22-27, 2015 in Nice, France, continued a series of events targeting advanced computational paradigms and their applications. The target was to cover (i) the advanced research on computational techniques that apply the newest human-like decisions, and (ii) applications on various domains. The new development led to special computational facets on mechanism-oriented computing, large-scale computing and technology-oriented computing. They are largely expected to play an important role in cloud systems, on-demand services, autonomic systems, and pervasive applications and services.

The conference had the following tracks:

- Computing technologies
- Computational intelligence strategies
- Large-scale computing strategies

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the FUTURE COMPUTING 2015 technical program committee, as well as the numerous reviewers. The creation of such a 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 effort to contribute to FUTURE COMPUTING 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the FUTURE COMPUTING 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope FUTURE COMPUTING 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of future computational technologies and applications. We also hope that Nice, France provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

FUTURE COMPUTING 2015 Chairs

FUTURE COMPUTING 2015 Advisory Chairs

Cristina Seceleanu, Mälardalen University, Sweden

Hiroyuki Sato, The University of Tokyo, Japan

Miriam A. M. Capretz, The University of Western Ontario - London, Canada

Kendall E. Nygard, North Dakota State University - Fargo, USA

Vladimir Stantchev, SRH University Berlin - Institute of Information Systems, Germany

Wail Mardini, Jordan University of Science and Technology, Jordan

Alexander Gegov, University of Portsmouth, UK

FUTURE COMPUTING 2015 Industry/Research

Francesc Guim, Intel Corporation, Spain

Wolfgang Gentzsch, The UberCloud, Germany

Noboru Tanabe, Toshiba Corporation, Japan

FUTURE COMPUTING 2015

Committee

FUTURE COMPUTING Advisory Chairs

Cristina Seceleanu, Mälardalen University, Sweden
Hiroyuki Sato, The University of Tokyo, Japan
Miriam A. M. Capretz, The University of Western Ontario - London, Canada
Kendall E. Nygard, North Dakota State University - Fargo, USA
Vladimir Stantchev, SRH University Berlin - Institute of Information Systems, Germany
Wail Mardini, Jordan University of Science and Technology, Jordan
Alexander Gegov, University of Portsmouth, UK

FUTURE COMPUTING 2015 Industry/Research

Francesc Guim, Intel Corporation, Spain
Wolfgang Gentzsch, The UberCloud, Germany
Noboru Tanabe, Toshiba Corporation, Japan

FUTURE COMPUTING 2015 Technical Program Committee

Jesús B. Alonso-Hernández, University of Las Palmas de Gran Canaria, Spain
Francisco Arcas Túnez, Universidad Católica San Antonio, Spain
Ofar Arieli, The Academic College of Tel-Aviv, Israel
Cristina Alcaraz, University of Malaga, Spain
Mohsen Askari, University of Technology Sydney, Australia
Panagiotis D. Bamidis, Aristotle University of Thessaloniki, Greece
Rudolf Berrendorf, Bonn-Rhein-Sieg University, Germany
Abdelhani Boukrouche, University of Guelma, Algeria
Alberto Cano, University of Córdoba, Spain
Miriam A. M. Capretz, The University of Western Ontario - London, Canada
Massimiliano Caramia, University of Rome "Tor Vergata", Italy
Jose M. Cecilia, Universidad Católica San Antonio, Spain
M. Emre Celebi, Louisiana State University, USA
Chin-Chen Chang, Feng Chia University, Taiwan, R.O.C.
Cheng-Yuan Chang, National United University, Taiwan
Wei Cheng, University of North Carolina, USA
Sung-Bae Cho, Yonsei University, Korea
Rezaul A. Chowdhury, Stony Brook University, USA
Rosanna Costaguta, Universidad Nacional de Santiago del Estero, Argentina

Zihua Cui, Taiyuan University of Science and Technology - Shanxi, China
Marc Daumas, INS2I & INSIS (CNRS), France
Isabel Maria de Sousa de Jesus, ISEP-Institute of Engineering of Porto, Portugal
Sérgio Deusdado, Polytechnic Institute of Bragança, Portugal
Leandro Dias da Silva, Federal University of Alagoas, Brazil
Qin Ding, East Carolina University, USA
Prabu Dorairaj, NetApp Inc, India
Marek J. Druzdzal, University of Pittsburgh, USA
Fausto Fasano, University of Molise, Italy
Dietmar Fey, Friedrich-Alexander-University Erlangen-Nuremberg, Germany
Francesco Fontanella, Università degli Studi di Cassino e del Lazio Meridionale, Italy
Félix J. García Clemente, University of Murcia, Spain
Miguel Garcia, University of Valencia, Spain
Teresa Garcia-Valverde, Universidad Católica San Antonio de Murcia, Spain
George A. Gravvanis, Democritus University of Thrace, Greece
Wolfgang Gentzsch, The UberCloud, Germany
Paul Gibson, Telecom Sud Paris, France
Debasis Giri, Haldia Institute of Technology, India
Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Victor Govindaswamy, Concordia University Chicago, USA
David Greenhalgh, University of Strathclyde Glasgow, UK
Michael Grottko, University of Erlangen-Nuremberg, Germany
Miguel Angel Guevara Lopez, IEETA - University of Aveiro, Portugal
Francisc Guim, Intel Corporation, USA
Hamid Harroud, Al Akhawayn University in Ifrane, Morocco
Daniela Hossu, University 'Politehnica' of Bucharest, Romania
Yongjian Hu, University of Warwick - Coventry, UK
Tee Sim Hui, Multimedia University, Malaysia
Saurabh Hukerikar, University of Southern California, USA
Jong-Gyu Hwang, Korea Railroad Research Institute (KRRRI), Korea
Woomin Hwang, Korea Advanced Institute of Science and Technology (KAIST), Korea
Muhammad Iftikhar, Universiti Malaysia Sabah (UMS), Malaysia
Sergio Ilarri, University of Zaragoza, Spain
Syed Mohammed Shamsul Islam, University of Western Australia, Australia
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Korhan Karabulut, Yasar University, Turkey
Jonghwa Kim, University of Augsburg, Germany
Georgios Kioumourtzis, Ministry of Citizen Protection and Public Order - Center for Security Studies, Greece
Scott A. Klasky, Oak Ridge National Laboratory (ORNL), USA
Michihiro Koibuchi, National Institute of Informatics, Japan
Zbigniew Kokosinski, Cracow University of Technology, Poland
Piotr Andrzej Kowalski, SRI Polish Academy of Sciences and Cracow University of Technology, Poland

Bartosz Krawczyk, Wroclaw University of Technology, Poland
Dalia Kriksciuniene, Vilnius University, Lithuania
Carlos León de Mora, Universidad de Sevilla, Spain
Pierre Leone, University of Geneva, Switzerland
Vitus Leung, Sandia National Labs, USA
Hongen Liao, Tsinghua University, China
Chun-Cheng Lin, National Chiao Tung University, Taiwan
Lan Lin, Ball State University, USA
Zihuai Lin, University of Sydney, Australia
Lu Liu, University of Derby, UK
Yu-lung Lo, Chaoyang University of Technology, Taiwan
Jay Lofstead, Sandia National Laboratories, USA
Stephane Maag, Telecom SudParis, France
Maciej Machulak, Cloud Identity Limited, UK
Olaf Maennel, Tallinn University of Technology, Estonia
Francesco Maiorana, University of Catania, Italy
Giuseppe Mangioni, University of Catania, Italy
Francesco Marcelloni, University of Pisa, Italy
José María Luna, University of Córdoba, Spain
Francesca Martelli, Institute for Informatics and Telematics (IIT) - Italian National Research Council (CNR), Italy
Gregorio Martinez, University of Murcia, Spain
Célia Martinie, ICS-IRIT - University Toulouse 3, France
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Rami Melhem, University of Pittsburgh, USA
Silvano Mignanti, Sapienza University of Rome / A.R.S. Sistemi / ValueUp S.r.l. / Prisma s.r.l. / Selex-ES, Italy
Nader F. Mir, San Jose State University, USA
Yehya Mohamad, Fraunhofer Institute for applied information technology (FIT), Germany
Ehsan Mousavi, University of Science and Technology - Tehran, Iran
Isabel Münch, German Federal Office for Information Security (BSI), Germany
Susana Munoz Hernández, Universidad Politécnica de Madrid, Spain
Andrés Muñoz Ortega, University of Murcia, Spain
Marco Mussetta, Politecnico di Milano, Italy
Kazumi Nakamatsu, University of Hyogo, Japan
Joan Navarro, La Salle - Universitat Ramon Llull, Spain
Isabel L. Nunes, Universidade Nova de Lisboa - Caparica, Portugal
Cyril Onwubiko, Research Series Ltd., UK
Gonzalo Pajares, University Complutense of Madrid, Spain
Panos M. Pardalos, University of Florida, USA
Przemyslaw Pocheć, University of New Brunswick, Canada
Aurora Pozo, Federal University of Paraná, Brazil
Prakash Ranganathan, University of North Dakota-Grand Forks, USA
Danda B. Rawat, Georgia Southern University, USA

Hendrik Richter, HTWK Leipzig, Germany
Carsten Röcker, Fraunhofer IOSB-INA, Germany
Ivan Rodero, NSF Center for Autonomic Computing / Rutgers the State University of New Jersey, USA
Marcos A. Rodrigues, Sheffield Hallam University, UK
Alfonso Rodriguez, University of Bio-Bio, Chile
Hiroyuki Sato, The University of Tokyo, Japan
Mitsuhisa Sato, University of Tsukuba, Japan
Robert Schaefer, AGH University of Science and Technology, Poland
Peter Schartner, Universität Klagenfurt, Austria
Elad Michael Schiller, Chalmers University of Technology, Sweden
Andrew Schumann, University of Information Technology and Management in Rzeszow, Poland
Cristina Seceleanu, Mdh, Sweden
Emilio Serrano, Polytechnic University of Madrid, Spain
Patrick Siarry, Université de Paris 12, France
Jurij Šilc, Jožef Stefan Institute, Slovenia
Georgios Sirakoulis, Democritus University of Thrace, Greece
Devan Sohier, Université de Versailles - St-Quentin-en-Yveline, France
Fengguang Song, Indiana University-Purdue University Indianapolis, USA
Mu-Chun Su, National Central University, Taiwan
Young-Joo Suh, Pohang University of Science and Technology (POSTECH), Korea
Vladimir Stantchev, SRH University Berlin - Institute of Information Systems, Germany
Anca-Juliana Stoica, Uppsala University, Sweden
Pierre Sutra, University of Neuchâtel, Switzerland
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Dan Tamir, Texas State University, USA
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Carlos M. Travieso-González, University of Las Palmas de Gran Canaria, Spain
Raquel Trillo Lado, University of Zaragoza, Spain
George Tsihrintzis, University of Piraeus, Greece
Teng Wang, Auburn University, USA
Alexander Wijesinha, Towson University, USA
Zhengping Wu, University of Bridgeport, USA
Chao-Tung Yang, Tunghai University, Taiwan R.O.C.
Yi-Hua Edward Yang, Xilinx Inc., USA
Hongji Yang, De Montfort University (DMU) - Leicester, UK
Lina Yao, University of Adelaide, Australia
Peng-Yeng Yin, National Chi Nan University, Taiwan
Zhibin Yu, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
Fernando Zacarias, Benemerita Universidad Autonoma de Puebla, Mexico
Ales Zamuda, University of Maribor, Slovenia
Marcelo Zanchetta do Nascimento, Federal University of Uberlândia (UFU), Brazil
Erliang Zeng, University of South Dakota, USA

Yong-Ping Zhang, Huawei Technologies Co., Ltd., China

Zhiyong Zhang, Henan University of Science and Technology, P.R. of China

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Ternary Arithmetic Pipeline Architectures Using Multi-bit Memristors <i>Dietmar Fey</i>	1
Bringing Colours to the Black Box - A Novel Approach to Explaining Materials for Evolution-in-Materio <i>Dragana Laketic, Gunnar Tufte, Stefano Nichele, and Odd Rune Lykkebo</i>	7
Is There Chaos in Blobs of Carbon Nanotubes Used to Perform Computation? <i>Stefano Nichele, Dragana Laketic, Odd Rune Lykkebo, and Gunnar Tufte</i>	12
Specification and Verification of Garbage Collector by Java Modeling Language <i>Wenhui Sun, Yuting Sun, Zhifei Zhang, Jingpeng Tang, Kendall Nygard, and Damian Lampl</i>	18
From Personal Computers to Personal Computing Networks: A New Paradigm for Computation <i>Mark Burgin, Rao Mikkilineni, and Giovanni Morana</i>	24
Assessment of MVAR Injection in Power Optimization for a Hydrocarbon Industrial Plant Using a Genetic Algorithm <i>Mohammed Abido, Muhammad Al-Hajri, and Mohamed Darwish</i>	31
Chemistry-inspired, Context-Aware, and Autonomic Management System for Networked Objects <i>Mahmoud ElGammal and Mohamed ElToweissy</i>	38
An Approach to Model Resources Rationalisation in Hybrid Clouds through Users Activity Characterisation <i>Giovanni Battista Barone, Vania Boccia, Davide Bottalico, Rosanna Campagna, Luisa Carracciolo, and Giuliano Laccetti</i>	48
Visualization Tool for Development of Communication Algorithms and a Case Study Using the K Computer <i>Syunji Yazaki, Ryohei Suzuki, Fumiyoshi Shoji, Kenichi Miura, and Hiroaki Ishihata</i>	54
Creating, Protecting and Sharing of Semantically-Enabled User-Orientated Electronic Laboratory Notebook in a Service Environment <i>Tahir Farooq and Zulkifly Mohd Zaki</i>	60

Ternary Arithmetic Pipeline Architectures using multi-bit Memristors

Dietmar Fey

Friedrich-Alexander-University Erlangen-Nürnberg (FAU)
 Department Computer Science 3, Chair for Computer Architecture
 Email: dietmar.fey@fau.de

Abstract—The paper addresses the possibilities exploiting the multi-bit storing features of memristor cells for balanced ternary signed-digit (SD) arithmetic pipelines realising addition, subtraction and multiplication. Different pipeline schemes using memristors as pipeline registers are shown in overview. Their functionality was verified on digital level by SystemC simulations and their performance is comparatively evaluated using analytic methods. In a ternary SD number system the digits of the operands can be assigned not only to 0 and 1 but also to minus 1. Such an SD representation has the great advantage that additions and subtractions can be carried out in $O(1)$ step, i.e., the run time is independent from the word length N . However, this requires a physical memory cell that allows to store reliably not only two but at least three states and in addition such a device has to offer fast access times and must be compatible with CMOS logic. All these constraints are fulfilled by memristors. Using memristor based pipeline registers offers different alternatives for implement fast arithmetic architecture circuits that differ from conventional ones. In particular, they allow to use more homogeneous pipelines for realising addition, subtraction and multiplication than current superscalar pipelines, which use for different operations also different pipeline paths. However, for the realisation of a homogeneous pipeline using memristors there are different possibilities. Those were evaluated in this paper concerning latency and hardware effort measured in number of required logic elements for computing and memristors for storing.

Keywords—Memristive computing; Ternary logic, Signed-digit arithmetic.

I. INTRODUCTION

Since the invention of a nano-device in the HP labs in 2008 by a team around Williams [1] that shows the specific behaviour prognosticated by Leon Chua in a paper of 1971 [2], and which was called by Leon Chua *memristor*, a lot of research launched in the community to use such new devices for digital and analogue logic operations. A memristor is an electronic device that can sustainably change its internal electrical resistance due to a change of an applied magnetic flux generated by electronic charge. Normally, this change of the magnetic flux and charge is generated indirectly by applying a voltage at the poles of a two-terminal memristor device to produce a current flow through it, if the memristor is connected to a circuit. Chua claimed, that if the current-voltage curve of such a device shows hysteresis behaviour and runs through the origin (see Figure 4), then it is a memristor ("if it's pinched, then it's a memristor"). Depending on the direction of the current the changing of the device's resistance can be made reversible. For example, it is possible to mimic the strength of a synaptic interconnect that is controlled by the time difference of two currents arriving at the two poles of the device to realise analogue neuromorphic processing

based on the spike time dependent plasticity (STDP) model [3]. Furthermore, complete memristor networks were proposed for digital processing. In such networks targeted changing of the memristors' resistance, called *memristance*, was induced in specific memristors to realise fundamental digital Boolean logic elements [4][5][6].

In some works, the research was also focused on the multi-bit storing feature of a memristor, i.e., the capability to store different resistance levels produced by periodic voltage excitations at the memristor poles. That multi-bit feature is also the topic in this paper. The idea, that was first published in [8], is to use memristors as elementary ternary storing cells for registers in a ternary arithmetic unit. This unit is based on a number system, in which the addition is accelerated by using a balanced ternary representation, i.e., each digit can be assigned to 1, 0, and -1 (in the following denoted as $\bar{1}$).

Based on that addition, further arithmetic circuits like a multiplication can be built up in a pipeline structure. Memristors are to use as multi-bit pipeline registers to store the ternary digits, called *trits*. There are different possibilities for the exact structure of such a pipeline, e.g., to process directly two numbers in signed-digit (SD) representation or a hybrid arithmetic structures operating an SD number and a binary number. All these alternatives cause different realisation efforts on the logic side, i.e., the number of used gates, and on the storage side, i.e., the number of necessary memristors and transistors for the interface circuitry to convert trits, stored in a single memristor cell, to bits and vice versa. In this paper, some of the different alternatives are shown and are evaluated among each other to select the best for a future memristor-based ternary computer.

The rest of the paper is structured as follows. In Section II, the fundamentals of a memristor and the possibility to store multi-bits in such a single physical storage cell are explained. In Section III, the principal Boolean equations and benefits of SD arithmetic are illustrated. In particular, it is shown why a ternary arithmetic unit is to favour. Afterwards in Section IV, computer arithmetic and memristor technology are brought together by presenting different solutions for arithmetic pipelines using memristor-based registers. These different proposals are evaluated in Section V, in terms of latency, bandwidth and hardware effort. Finally, Section VI summarizes the most important statements of the paper.

II. MEMRISTOR

The core of a memristor is a so-called transfer oxidation metal, like e.g., TiO_2 . This metal oxide layer is sandwiched with a ionized layer of the metal oxide, TiO_{2-x} , between two metal plates (see Figure 1). Inside this two-terminal

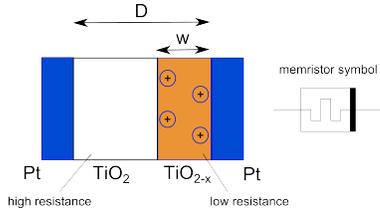


Figure 1. Structure of a two-terminal memristor cell

device electro-chemical processes of oxidation and reduction are taking place to move ions along the interface of the metal oxide and the ionized layer. This moving can be caused by an electrical field applied at the two metal poles. The change of voltage, $V(t)$, and current, $I(t)$, over time generates a magnetic flux, φ , and a charge, q , which both determine the memristance M of the device according to Leon Chua's assumption, expressed in (1) and (2).

$$d\varphi = Mdq \quad (1)$$

$$M(q(t)) = \frac{d\varphi/dt}{dq/dt} = \frac{V(t)}{I(t)} \quad (2)$$

This ion transfer is responsible for the sustainable changing of the resistive features of such an element. The electron depleted ionized layer (TiO_{2-x}) has a much lower resistance than the metal oxide layer. Both layers determine with different weights according to their lengths the resistance of the whole device (3). R_{OFF} is the resistance if the device is switched off, i.e., no ionized layer is available. R_{ON} holds for the contrast, i.e., the device contains of a complete ionized layer. Since the depletion zone can be shifted in both directions according to the direction, where the voltage is applied, the changing of the resistance is reversible. Therefore it can be memorized and this kind of reversible resistance with memory effect is called *memristance*.

$$R_{MEM}(x) = R_{ON} \cdot x + R_{OFF} \cdot (1 - x), \quad (3)$$

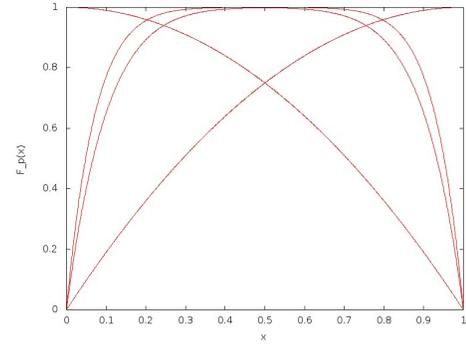
where $x = \frac{w}{D} \in (0, 1)$

The ratio of the length of ionized layer, w , to the total length of the two layers, D , is called the state variable x . The changing of this state variable in dependence on the time is expressed in (4),

$$\frac{dx}{dt} = \frac{\eta \cdot \mu_D \cdot R_{ON}}{D^2} I(t), \quad (4)$$

where μ_D corresponds to the drift mobility of the ions, η is equal to 1 or -1 , resp. The last parameter expresses the direction of the applied voltage causing either an expanding or a shrinking of the depletion zone.

The most elaborate presentation of the mathematical derivation of the behaviour of memristors based on ion transfer can be found in [11]. The paper derives a discretization of the differential equation (4), which is shown here by (5) to (8). These equations allow to calculate the changing of the parameters between two discrete time steps t_i and t_{i+1} .


 Figure 2. Modelling non-linearity with Bioblek's window function for different parameters $p = 1, 5$, and 7 . Considered is also $I(t) < 0$, the curve starting at $F_p(0) = 1$.

$$M(x(t_i)) = R_{ON} \cdot (x(t_i)) + R_{OFF} \cdot (1 - x(t_i)), \quad (5)$$

$$I(t_{i+1}) = \frac{V(t_{i+1})}{M(x(t_{i+1}))}, \quad (6)$$

$$x(t_{i+1})/dt = \frac{\eta \cdot \mu_D \cdot R_{ON}}{D^2} I(t_i) \cdot F_p(x(t_i)), \quad (7)$$

$$x(t_{i+1}) = x(t_{i+1})/dt \cdot [t_{i+1} - t_i] + x(t_i) \quad (8)$$

The comparatively simple mathematical modelling assumes a linear drifting of the ions according to (4). However, real memristors show a different drift behaviour the nearer the ions are located at the boundaries of the device, where the drifting is slower. This non-linearity is modelled by a so-called window function F_p in (7). Different window functions are proposed in literature [9][10]. In this paper, the window function from Bioblek [10], see (9), was selected since it simplifies convergence issues at the device boundaries compared to other window functions. Figure 2 shows the graph for Bioblek's window function.

$$F_p(x) = 1 - [x - u(-I)]^{2p}, \quad (9)$$

where $u(I) = \begin{cases} 1, & \text{if } I \geq 0 \\ 0, & \text{if } I < 0 \end{cases}$

To investigate the memristor's behaviour more detailed in this paper a numerical simulation program was written in C to implement (5) to (9). The outputs of the simulation program are shown in the following figures. Figure 3 shows the applied input voltage over time. First two positive voltage oscillations are applied at the two terminals of the memristor device. Then, this process is reversed by changing the polarization. Figure 4 shows the I-U behaviour. The typical hysteresis loops for memristors are arising. After each voltage oscillation the level of memristance is changed and another hysteresis loop is valid.

Figure 5 displays the value of the state variable x . The multi-state or multi-bit feature of a memristor can be clearly detected. With each excited voltage oscillation the state variable x was shifted to another level. Even if the values are not completely constant and the transition between the levels is smooth each level corresponds to some amount to a discrete memristance level. The level starts not at zero, the initial value

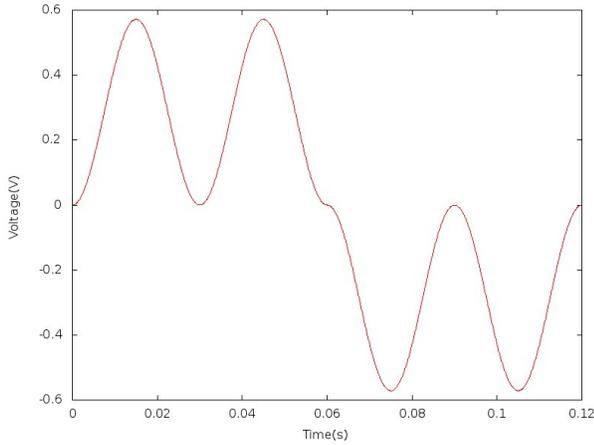


Figure 3. Input excitation with two voltage oscillations

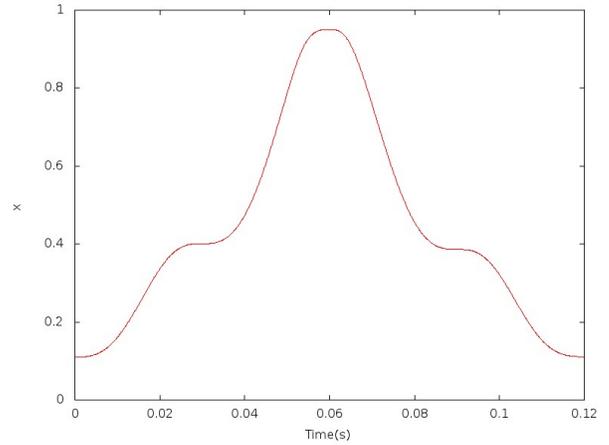


Figure 5. The course of the state variable x of a memristor cell

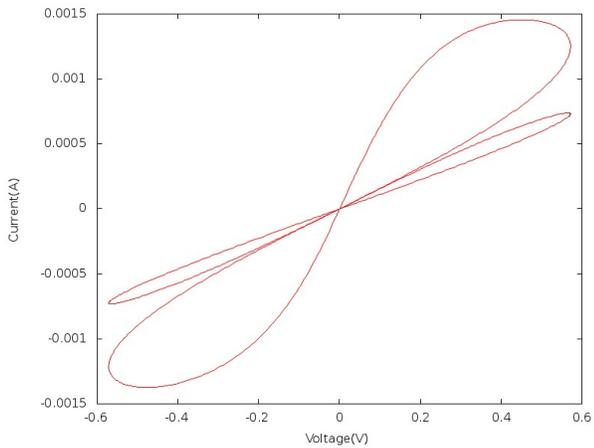


Figure 4. The memristor hysteresis loop, generated for $D = 41nm$, $x(0) = 0.11$, and $p = 7$. For each excited voltage oscillation another hysteresis behaviour is generated

for x was assumed as 0.11. It expresses an arbitrary assumed ratio between doped and undoped zone at the beginning determined by the memristor’s manufacturing process. Further, two levels were injected and then this process was withdrawn by the applied inverted voltage levels. Each of these three levels corresponds to one of the three signed digits, 1, 0, and $\bar{1}$, which are stored in one nano-sized physical cell. Such a cell is the base for pipeline registers used in the proposed SD arithmetic circuits proposed in the next section.

III. SD-ARITHMETIC

As already mentioned in the introduction the great advantage of arithmetic circuits using SD numbers is that the execution time of an addition has a complexity of $O(1)$, i.e., it is independent of the operands’ word length N [12]. Conventional adders which are executing binary operands like a carry-look-ahead adder or a conditional-sum adder show a run time complexity in $O(\log N)$, or in $O(N)$ like a ripple-carry adder. In order to avoid the generation of carry chains in an addition, a balanced number representation is used in an SD adder. That means that the number of negative digits is the same as the number of positive digits. Since in this

paper the focus is not on the computer arithmetic side, the functionality of an SD addition is explained by words, by presenting the basic formulae to execute a carry-free addition, and by examples to demonstrate the principle.

First, we investigate roughly the cost of an addition of two SD numbers. The addition is carried out in two steps. In the first step an intermediate sum vector, z , and a carry vector, c , is calculated. In the second step the two vectors, z and c , are post-processed to determine the final sum vector, s , which is also given in an SD number representation.

Adding two SD numbers requires a base $r \geq 3$. For the digits, sd_i , $0 \leq i < N$, of a N digit long SD number holds $sd_i \in \{-(r-1), -(r-2), \dots, 0, \dots, r-2, r-1\}$. Equation (10) shows the calculations that have to be performed in each digit position i to determine the vectors z and s for two SD input operands x and y . The term $c_i \cdot r_i$ in (10) is a so-called correction term. That correction term secures that $z_i \leq |r-2|$. This prevents that the possible addition of a carry bit will cause an additional carry to the left neighbored position. To compensate the addition/subtraction with $c_i \cdot r_i$ it is necessary that a carry bit, either +1 or -1, has to be added in the next left neighbored digit position $i + 1$.

$$z_i = x_i + y_i - c_i \cdot r$$

$$c_i = \begin{cases} +1, & \text{if } (x_i + y_i) > r - 2 \\ -1, & \text{if } (x_i + y_i) < -(r - 2) \\ 0, & \text{if } -(r - 2) \leq (x_i + y_i) \leq r - 2 \end{cases} \quad (10)$$

$$s_i = z_i + c_{i-1}$$

Table I shows an example calculation to a base $r = 3$ with a word length $N = 4$. The two SD numbers to add are $1 \cdot 3^3 + (-2) \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^0 = 12$ and $0 \cdot 3^3 + (-1) \cdot 3^2 + 1 \cdot 3^1 + 1 \cdot 3^0 = -5$. The correction term for the second and the first digit positions are +3 and -3, resp., since $x_2 + y_2 = -3 < -(r-2) = -1$ and $x_1 + y_1 = 2 > (r-2) = 1$. In Table I the carry vector, c_{i-1} , is already shifted to left for a better illustration of the final addition in (10). As expected the sum is $s = 0 \cdot 3^3 + 1 \cdot 3^2 + (-1) \cdot 3^1 + 0 \cdot 3^0 = 7$.

An evaluation of this addition scheme to a base $r = 3$ and a realisation with memristors yields the following statements.

TABLE I. ADDING TWO SD NUMBERS x AND y IN TWO STEPS.

	1	-2	1	0	= 12	x
	0	-1	1	1	= -5	y
step1	1	0	-1	1		z
	-1	+1	0	.		c_{i-1} shifted to left
step 2	0	+1	-1	1	= 7	s

For the required signed digits, sd_i , holds $sd_i \in [-2, 2]$ That means five levels are to store in a single physical memristor cell. That is not impossible in principle but there are two more states to realise than in a ternary computer making the interface circuitry to the memristor cells more difficult to realise. Furthermore a larger effort for the coding is necessary. Three bits are required to code the five possible values, yielding $\binom{8}{5} \cdot 5! = 6720$ different possibilities to assign five 3-bit long code words for $x_i, y_i, z_i, 0 \leq i \leq 2$, to cover the values $\{-2, \dots, +2\}$. Table II looks like nearly as the function table from which the Boolean logic is to derive that allows calculating the sum code bits.

TABLE II. Scheme for function table to determine (10).

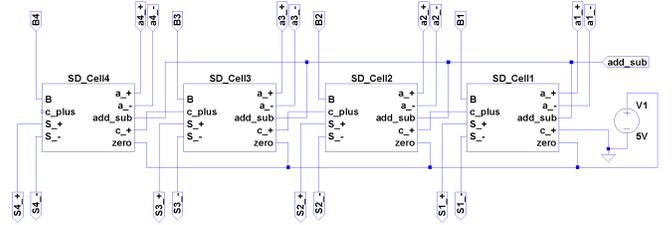
x_2	x_1	x_0	y_2	y_1	y_0	2 bits for $r \cdot c_i$	z_2	z_1	z_0
5 entries						0, 1 or -1
for $[-2, +2]$						0, 1 or -1
...						0, 1 or -1
6720 possibilities						0, 1 or -1
for 25 entries					

To the best knowledge of the author it is not known if a minimal coding for such a function exists. However, it is to expect that the gate logic is much higher compared to an alternative operation mode, which is preferred in this paper.

This refers to an arithmetic that allows to operate a SD number, SD , with a binary number, B . For such an operation the coding for the SD number producing a minimum gate effort is known. Already in the 1990s Muller and Duprat [13] published the efficient coding scheme shown in Table III for the positive and the negative channel of a SD number, SD . This allows ternary coding, what in addition simplifies the realisation of the interface to store the SD value in a memristor cell since only three states are to distinguish. Furthermore the necessary gate logic, shown in (11), to calculate the intermediate sum bits, z_i , the carry bits, c_i , and the positive and negative channel of the sum, s , is comparatively simple. An example calculation for the addition of a SD number, a , and a binary number, B , presents Table IV. The shown bits for z_i, c_i, SD_i^+ , and SD_i^- are calculated by applying all the equations given in (11) in each digit position. The technique to avoid the carry propagation in this case is to generate in the intermediate sum vector only 0's and -1's, and in the carry vector only 0's and +1's. This excludes that in the second step two positive or negative 1's will meet in one digit position to produce a further carry.

TABLE III. Digit coding of a binary SD number.

SD^+	SD^-	Value for SD
0	0	0
0	1	$\bar{1}$
1	0	1
1	1	not defined


 Figure 6. Schematic for a 4-digit SD adder that can add/subtract a SD number a and a binary number B

$$\begin{aligned}
 c_i^+ &= SD_i^+ \vee (B_i \wedge \overline{SD_i^-}) \\
 z_i^- &= (SD_i^+ \vee SD_i^-) \oplus B_i \\
 s_i^+ &= z_i^- \wedge c_{i-1}^+ \\
 s_i^- &= \overline{c_{i-1}^+} \wedge z_i^-
 \end{aligned} \tag{11}$$

A detailed analysis how the Boolean equations from (11) can be derived and how they are mapped onto an SD adder using memristors as in-/outputs can be found in [13][8]. There is also shown a detailed verification of the functioning of the SD adder by SPICE simulations [8]. A schematic of that SD adder is shown in Figure 6. In each module denoted as SD cell the equations shown in (11) are carried out on the SD number a , and the binary number B for word length of $N = 4$. The signal $addsub$ determines if an addition or a subtraction is carried out. A subtraction can be easily reduced to an addition according to (12). The inverse of an SD number can be simply determined by exchanging the negative with the positive channel. The signal $zero$ clears the binary input, if it is set to logical 0, i.e., an addition is carried out with zero. That is necessary in case of a multiplication if the multiplier bit is zero. If $zero$ is set to 1, the corresponding bit of the second operand B is passed. The output sum, s , as well as the input, a , are SD numbers, which have a positive and a negative part and can be stored in memristor cells. Details how this adder is operating can be found in [8]. There is also included a comparison that proves the performance benefits of an SD adder compared to existing methods like ripple-carry or carry-look-ahead adders. In this paper, the focus is laid on how combinations of such SD adders can be used for a much easier operation of two SD numbers, which is reduced to subsequent series of additions of a SD number and a binary number. The price for that reduction is only a small higher latency compared to the direct addition of two SD numbers shown in Table I.

TABLE IV. Example of a carry-free addition for two positive integers, A and B, using binary coded SD numbers.

a	.	0	$\bar{1}$	0	1	= (-3) ₁₀
B	.	1	0	0	1	= (9) ₁₀
z	0	$\bar{1}$	$\bar{1}$	0	0	
c	1	0	0	1	.	
s	1	$\bar{1}$	$\bar{1}$	1	0	= (16 - 8 - 4 + 2) ₁₀ = (6) ₁₀

$$SD - B = -(-SD + B) \tag{12}$$

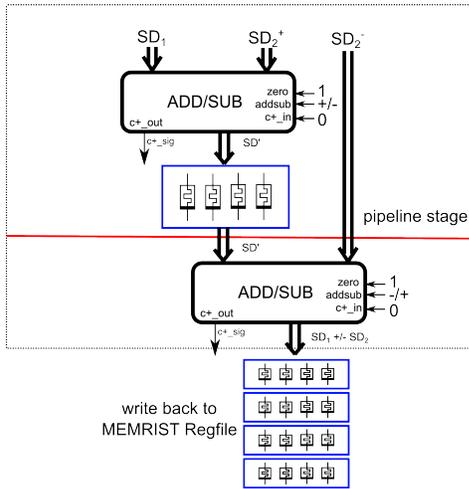


Figure 7. Schematic for adding/subtracting two SD numbers using memristors as pipeline register

IV. MEMRISTOR BASED PIPELINE STRUCTURES

An addition between two SD numbers, SD_1 and SD_2 , can be decomposed in two steps, in which in the first step SD_1 is processed with the positive channel of SD_2^+ and in the second step the negative channel is subtracted from the result of the addition (13). In the case of a subtraction the signs of the operators signs have to be inverted (+ becomes - and vice versa).

$$SD_1 \pm SD_2 = ((SD_1 \pm SD_2^+) \mp SD_2^-) \quad (13)$$

Figure 7 shows the schematic for a corresponding arithmetic circuit. Two of the addsub modules shown in Figure 6 are necessary with an appropriate interface circuitry, not shown in Figure 7, to store the first result in a memristor register. This allows to pipeline the calculation before the second addsub module comes to execution.

Figure 8 shows a scheme for a multiplication of a SD number a with a binary number B . A multiplication of operands with N digits is reduced back to N subsequent additions/subtractions. Each addsub module forms a pipeline with memristors to store the partial product digits. In the first three steps the product digits, $P[0]$ to $P[3]$, are produced before in the last pipeline step the final product digits $P[4]$ to $P[7]$ appear at the output of the last addsub module and all digits can be stored in a memristor register file. The digits of the SD operand $a[i]$ determine in each pipeline stage, i , if either an addition ($a[i] = 0$) or a subtraction ($a[i] = 1$) has to be carried out with the partial product and the binary number B , or if in case $a[i] = 0$ no addition takes place.

Multiplication of two SD numbers, $SD_1 \cdot SD_2$ can be traced back to a hybrid operation scheme that executes a SD number and a binary number (14) as it was done for the addition/subtraction, see Figure 9. First, two intermediate products, $Prod_1$ and $Prod_2$, are generated with two concurrently operating circuits from Figure 8. Afterwards these two products, given in SD number representation, are subtracted with a two step circuit analogue to the addition/subtraction shown in Figure 7. If no multiplication has to be carried out

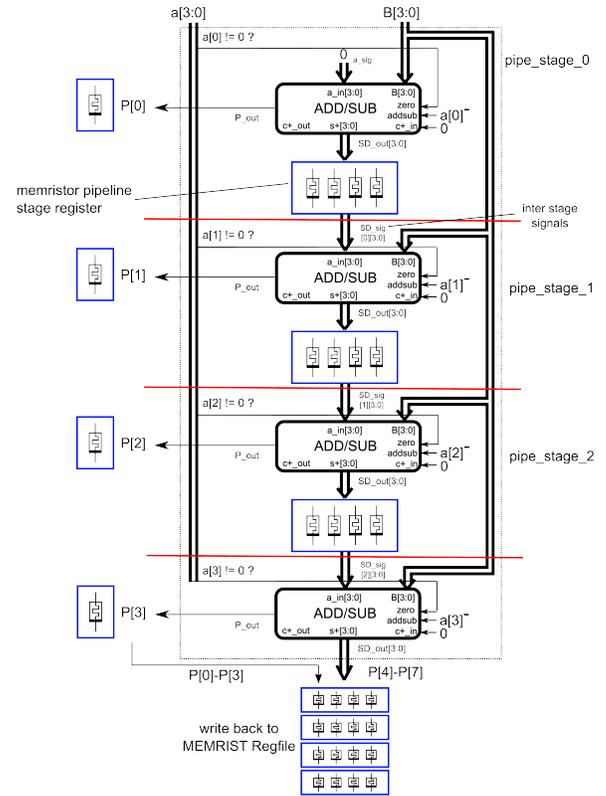


Figure 8. Memristor-based pipelining for the multiplication of a SD number, a , and a binary number B

the two multiplier pipelines can be used for two concurrent additions/subtractions. Therefore, Figure 9 represents the core of a future memristor multi-threading architecture, which the author intends to realise with largely homogeneous pipeline structures.

$$SD_1 \cdot SD_2 = (SD_1 \cdot SD_2^+) - (SD_1 \cdot SD_2^-) \quad (14) \\ = Prod_1 - Prod_2$$

V. EVALUATION OF MEMRISTOR-BASED PIPELINE STRUCTURES

The functionality of all proposed pipeline architectures was verified by simulations in SystemC. In this section the pipeline architectures are compared with regard to latency, expressed in gate delays, achievable throughput, and necessary hardware effort, which is measured in numbers of necessary transistors and memristors (Table V).

First, the add/sub pipeline from Figure 7 is analysed. If Δ corresponds to one gate delay, according to (11) an add-sub module requires 4Δ plus the time, t_{conv} , for converting the digital signals by a number of voltage oscillations (see Figs. 3 and 5) to store the SD adder outputs in multi-bit memristor cells at the end of the first pipeline stage. In [8], $N \cdot 62$ transistors were determined for the used SD adder, shown in Figure 6. Additionally, the number of transistors for the conversion, $trans_{conv}$, is to add for each of the N memristors. An adder that adds directly two SD numbers using

TABLE V. Evaluation of the memristor-based pipeline architectures. Time to write back to MEMRIST regfile is not considered

	latency	# memristors	# transistors
Addition/Subtraction	$4\Delta + t_{conv}$	N	$N \cdot (124 + trans_{conv})$
SD-binary multiplier	$N \cdot (4\Delta + t_{conv})$	N^2	$N^2 \cdot (62 + trans_{conv})$
SD-SD multiplier	$(N + 2) \cdot (4\Delta + t_{conv})$	$2N^2 + 4N$	$(2N^2 + 4N) \cdot (62 + trans_{conv})$

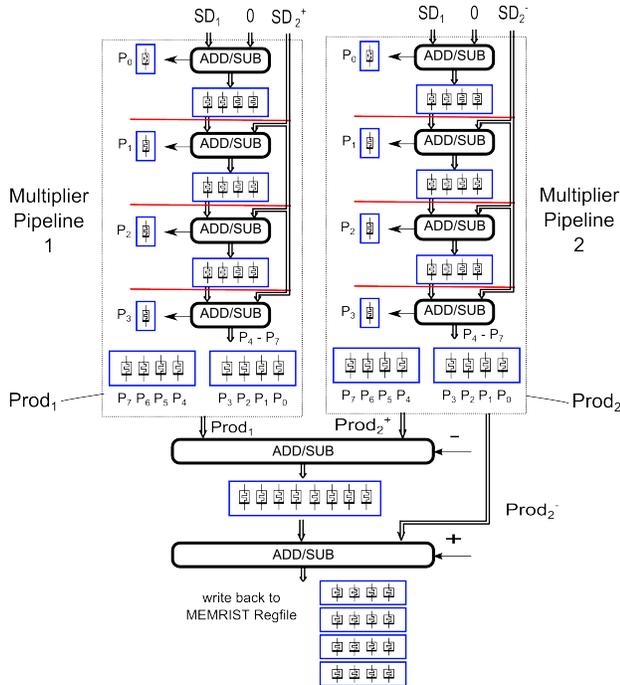


Figure 9. Memristor-based pipelining for the multiplication of two SD numbers with a digit length of four.

a base $r = 3$ would show a latency of only 2Δ . However, it is to expect that this requires much more transistors for logic and conversion, since five and not three multiple memristor states have to be stored. Therefore, the conclusion is drawn that the double latency time of a SD-binary adder is acceptable.

Counting the number of memristors and addsub modules of the SD-binary multiplier in Figure 8 for the case $N = 4$ and deducing from that the generic case yields the numbers shown in Table V for latency and hardware effort. The SD-SD multiplier consists of two multipliers and two further SD adders with $2N$ digit length leading to the values entered in the last row of Table V. The throughput for all architectures is equal to the reciprocal of the latency in one stage, $1/(4\Delta + t_{conv})$. Access times for memristors in the range of $0.3ns$ make realistic to achieve a clock frequency of more than $1GHz$.

VI. CONCLUSION

Memristors and their capabilities to store multi-bits in a single cell allow the set-up of fast CMOS compatible carry-free arithmetic circuits. The presented analysis shows that the effort for the realisation of ALUs with the proposed hybrid SD-binary multiplier pipeline approach and pipeline memristors is quite moderate and to prefer versus a pure SD-SD adder arithmetic. The half latency time of the hybrid solution is

offset against the pure SD approach that would cost much more hardware. Future work requires the design of a corresponding control unit to feed the pipeline with a stream of instructions and a detailed analysis how to realise efficiently an interface circuit to the memristors.

ACKNOWLEDGMENT

The author thanks Jonathan Martschinke who wrote the numerical simulation program of memristor's behaviour that was used to generate Figs. 3 to 5.

REFERENCES

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. Stanley Williams, The missing memristor found. *Nature*, vol. 453, 2008, pp. 80–83.
- [2] L. Chua, Memristor-the missing circuit element. *Circuit Theory, IEEE Transactions on*, vol. 18, no. 5, September, 1971, pp. 507–519.
- [3] B. Linares-Barranco and T. Serrano-Gotarredona, Memristance can explain spike-time-dependent plasticity in neural synapses. *Nature Precedings*, March, 2009, hdl:10101/npre.2009.3010.1.
- [4] S. Kvatinsky, A. Kolodny, U. C. Weiser, and E. G. Friedman, Memristor-based IMPLY logic design procedure. *Proc. 29th IEEE International Conference on Computer Design (ICCD)*, October, 2011, pp.142-147.
- [5] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, Memristor-based MRL Memristor ratioed logic. *Proc. 13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, August, 2012, pp. 1-6.
- [6] I. Vourkas and G. Ch. Sirakoulis, Memristor-based combinational circuits: A design methodology for encoders/decoders. *Microelectronics Journal*, vol. 45, no. 1, January, 2014, pp. 59–70.
- [7] L. Zheng, S. Shin, and S.-M. S. Kang, Memristor-based ternary content addressable memory (mTCAM) for data-intensive computing. *Semiconductor Science and Technology*, vol. 29, no. 10, 2014, 104010 (10pp).
- [8] D. Fey, Using the multi-bit feature of memristors for register files in signed-digit arithmetic units. *Semiconductor Science and Technology*, vol. 29, no. 10, 2014, 104008 (13pp).
- [9] Y. N. Joglekar and S. J. Wolf, The elusive memristor: properties of basic electrical circuits. *Eur. J. Phys.*, vol.30, no. 661, January, 2009, pp. 1-24, arXiv:0807.3994 v2 [cond-mat.meshall].
- [10] Z. Biolek, D. Biolek, and V. Biolkov, SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, vol. 18,no. 2, June, 2009, pp. 210-214.
- [11] N. R. McDonald, R. E. Pino, P. J. Rozwood, and B. T. Wysocki, Analysis of dynamic linear and non-linear memristor device models for emerging neuromorphic computing hardware design. *IEEE IJCNN*, 2010, pp. 1-5.
- [12] A. F. González and P. Mazumder, Redundant arithmetic, algorithms and implementations. *INTEGRATION, the VLSI journal*, vol. 30, no. 1, 2000, pp. 13-53.
- [13] J. Duprat and J.-M. Muller, Fast VLSI implementation of CORDIC using redundancy. in E.F. Depreterre, and A.-J. van der Veen (eds.): *Algorithms and Parallel VLSI architectures*, Elsevier, vol. B, June, 1991, pp. 155-164.

Bringing Colours to the Black Box —

A Novel Approach to Explaining Materials for Evolution-in-Materio

Dragana Laketić, Gunnar Tufte, Stefano Nichele, Odd Rune Lykkebo

Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway

Emails: (dragana.laketic, gunnar.tufte, nichele, lykkebo)@idi.ntnu.no

Abstract—The work presented in this paper is done within the NASCENCE (NAAnoCSale Engineering for Novel Computation using Evolution) project which investigates computational properties of nanomaterials and unconventional computing paradigms which can be applied to such materials in order to achieve material computing. The paper presents a novel approach to modelling computations in one of the nanomaterials considered in the project - Single-Walled Carbon Nanotubes (SWCNTs)-polymer nanocomposites. Our belief is that the presented approach is more suitable for the experiments within NASCENCE. It takes inspiration from the principles of dynamical hierarchies and can be related to some known cellular computing architectures. Our motivation is given as well as some initial simulation results based on a simplified model built according to the proposed approach. The results show that the proposed approach captures well the conductivity dependence on the concentration of carbon nanotubes and varying electric potential in the material.

Keywords—Unconventional computing; Evolution-in-Materio; carbon nanotubes; SWCNT-polymer nanocomposites.

I. INTRODUCTION

Computing devices based on silicon (Si) technology are not likely to meet the needs for the extent of computations needed by humankind in the future. Novel technologies need to be found which will provide for the growing scales of computations but also for the demands such as energy efficiency and similar. In that it is not only sufficient to find a promising replacement for Si, i.e., some novel material which can perform computations; it is also necessary to discover new computing paradigms better suited to novel technologies.

The NASCENCE project [1] investigates both. Materials considered so far are coated gold nanoparticles and nanocomposites made of SWCNTs and polymer molecules but also, recently, SWCNTs / Liquid Crystal (LC) dispersions. Methods of material manipulation for achieving computation are explained within Evolution-in-Materio (EIM) [2][3] whereby Evolutionary Algorithms (EA) are employed to search for the solution of a computational task in materio.

Exploitation of computational properties of a material requires a good understanding of its properties and behaviours when exposed to certain type of excitation. A good understanding of the material properties is best tested in a model of the material. If a model produces responses similar to the experiments in the lab, then model is sufficiently good to make good predictions of the material behaviour in simulation thereby saving experimental time and resources.

However, to make a model good enough for the CNT-based materials used in NASCENCE is not an easy task. This is due to the setup of the experiments but also the bulk of material itself: there is no tidy alignment of CNTs, rather there is a blob of nanocomposite where Van der Waals' forces keep SWCNTs in bundles which, sustained by polymer molecules, stretch in all directions. Further, there is no 'neat' electric field as is the case, for example, in the CNT-gate stretching from source to drain electrode in thin film transistors [4][5]. In NASCENCE experiments, electric field is the result of the voltages on a number of electrodes which are immersed into a blob, thick film of SWCNT-polymer nanocomposite. As existing models and simulation methods fail to serve the purpose, a novel approach is needed.

This paper presents one of the approaches considered within the project. In NASCENCE, a material under experiments is treated as a black box - some signals are brought to it and some signals are read as a response. If the aim is to model what takes place in the material, we need to unbox it and look inside searching for the physical processes which lead to material computing. Different levels of details can be accounted for when addressing the physics at the bottom of the computations produced by the material. The approach presented in this paper is based on two paradigms borrowed from complex systems: dynamical hierarchies [6][7] and cellular computing [8][9].

The paper is organised as follows. Section II gives a brief overview of the computing materials and EIM. Section III presents the approach to modeling SWCNT-based nanocomposites. Section IV shows simulation results based on a simple model and, finally, Section V provides a discussion of a proposed approach and some directions for future work.

II. MATERIALS THAT COMPUTE

The notion is not novel. Ferrous sulphate which discriminates between the frequencies [10], frequency discriminator in FPGA tissue [11], pattern recognition in a bucket of water [12], robot controllers in liquid crystal [13] - are just a few examples, some of which have been known for as long as half a century. Beside finding suitable materials, another challenge is which method of material manipulation to use in order to achieve useful computation. One possible way is to use a bottom-up approach and let an EA search the space of possible solutions. EIM [2] uses computations performed by the material when stimulated by some signals from outside material which change

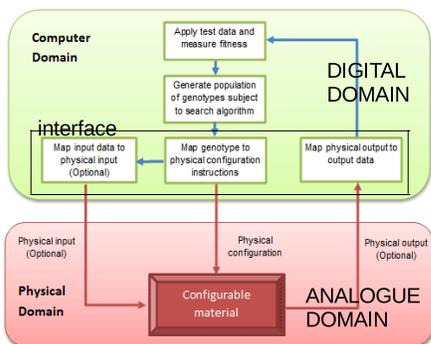


Figure 1. Overview of the Evolution-in-Materialio, taken and adapted from [14]. See Section II for explanation.

according to an EA. These signals configure the material so that it performs a certain computation. Therefore they are called *configuration signals*. EA is run on a computer where configuration (signals) of the material is represented as a genotype which undergoes evolutionary changes until material response corresponds to a desired computation. Figure 1 shows an overview of this process.

Configuration and input signals as well as fitness value are processed by the computer, i.e., they are digital while actual computations which happen in the material are analogue due to the physics exploited for computation. Figure 1 shows this cross-domain between digital and analogue worlds. Instructions which configure the material are generated by the computer (digital) and are subject to an EA which runs on a computer. Material response (analogue) is read and converted by interface so that it can be tested for fitness by the computer.

EIM is used in NASCENCE [1]. In a typical experiment, material is treated as a black box. We assume nothing about what is inside the sample of material when searching for solutions for some computational task. A more detailed description of the process used in the experiments can be found, for example, in [15][16].

In NASCENCE, it has already been shown that materials can solve computational problems [14][16]. However, having a problem solved is not enough. We would like to learn more about the possibilities that lie in the material. If the travelling salesman problem is successfully solved for 9 and 10 cities [14], can we expect the same blob of material to be equally successful at solving the problem for larger numbers of cities? Can we say more about problem scalability in some other way than purely by running exhaustive runs of experiments?

To get more knowledge about the computing properties of the material, a way to go is to construct a model of the material. A good model means a good understanding of the underlying laws of physics which govern behaviour of the system under investigation. A model also shows how well we understand the material - if a model is good enough, when used in simulations it will produce behaviours similar to those observed in the lab when we run experiments. The more we know about the material, the better can we make use of its properties for computations while saving experimental resources.

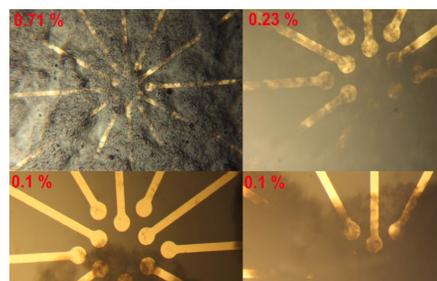


Figure 2. Optical micrographs of SWCNT-polymer nanocomposites dispersed over gold electrodes with 25µm electrode gap. Different concentrations exhibit various coverage of gold electrodes.

III. DICING MATERIAL INTO CELLS

Many models of CNTs exist [17]–[21]. However, usually they are suited for a certain purpose, for example to model conductivity of CNT gates in thin film transistors [18]. Our goal is to model conductivity of CNTs used in NASCENCE experiments. The conductivity of CNTs is due to the percolation paths they form and can be observed when CNTs are placed in an electric field. It changes either with the change of electric field or with the change of percolation paths if nanotubes are movable. In case of the samples currently under investigations, SWCNTs are not movable, they are held at certain positions by polymer molecules. So, the only way that conductivity of the sample can change is by changing the electric field to which the sample is exposed.

Therefore, it can be said that the challenge faced when modelling blobs of SWCNT-polymer nanocomposites used in NASCENCE experiments is multifaceted. The main two questions which need to be answered are:

- how to represent physics of SWCNT-polymer nanocomposites?
- how to represent electric field which changes according to the change of voltages on the electrode array?

Figure 2 shows micrographs of some of the samples used in NASCENCE experiments. Typically, it is a 20µL blob of SWCNT-polymer nanocomposite. Polymer is Poly(Methyl MethAcrylate) (PMMA) or Poly(Butyl MethAcrylate) (PBMA). The nanocomposite is dissolved in some solution to be dispersed over array of electrodes as a droplet and subsequently dried at high temperature leaving a thick film of nanocomposite spread over electrodes upon dissolvent’s evaporation. Electrodes are made of gold and can be placed in different arrangements - circular, square matrix - so that the distance between individual electrodes varies. Typically, it is several 10s of µm and so is the radius of individual electrodes.

Another feature which can be noticed from micrographs in Figure 2, is that the coverage of electrodes by nanotubes is not uniform. Some electrodes are more in contact with nanotubes than the others and some may even remain with no contact at all. Also, this is very much dependent on the SWCNT concentration. A somewhat simplistic sketch of the material sample at hand can look as given in Figure 3, which simplifies the content of the sample together with the electrodes.

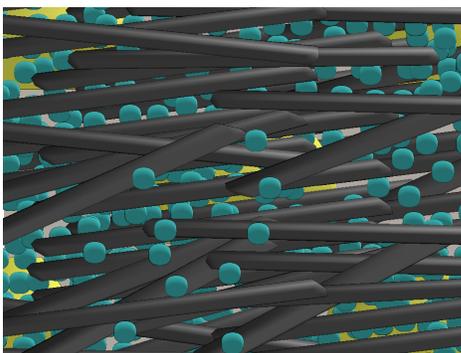


Figure 3. A sketch of the system used in experiments: SWCNT bundles are represented as grey sticks lying in all directions, polymer molecules are in a shape of balls and electrodes are golden yellow patches seen behind material.

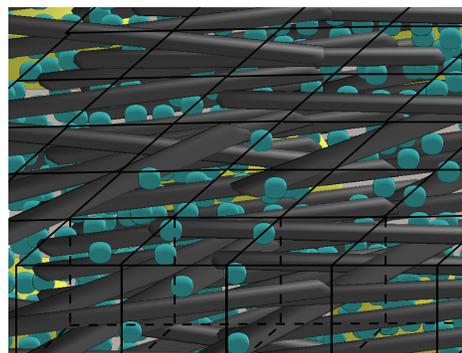


Figure 4. Example of the system viewed as a collection of cells.

A. *Abstracting away the details*

Behaviour of complex systems is the result of the dynamics of their constituent parts [22]. In [23], Herbert Simon named hierarchies as “one of the central structural schemes” that the architect of complexity uses. And, indeed, hierarchies, dynamical hierarchies [6][7] in particular, describe systems which are made of a number of components undergoing some dynamics and, in doing that, mutually interacting. Such behaviour leads to the emergence of novel units at higher hierarchical levels [24]. Hierarchical systems within which higher levels emerge as a result of the dynamics of lower level units have been studied from various approaches - philosophical, computational, information theoretical - that more as they carry the principles on which life itself emerges.

Interesting to notice is that dynamics at different levels happens at different rates [25]. The lower the level, the higher the rate of the dynamics [24] which further leads to averaging or “some selective loss of detail” as Pattee names it [26] or some filtering of the information from a lower to a higher level [27]. All the details of the dynamics at lower levels need not be fully known, rather some averaged, filtered information which describes it in sufficient detail.

B. *Cellular approach*

Another trait of complex systems is that their constituent parts operate in parallel. The behaviour or functionality of the system as a whole is the result of the operation of its parts but, as a rule, it cannot be simply represented as a sum of its parts, i.e., nonlinearity is inherent to the system. The related notion is *emergence* [28] which captures the property that “the whole is greater than the sum of its parts” and that nonlinearity is present.

Models of such systems are usually based on the representation of the parts of the system as cells. It is needed to find appropriate cell arrangement and some governing mechanism for the cell dynamics. This mechanism can be given in different forms - rules which describe how a cell transitions from one state to another - Cellular Automata (CA) [8] or some physics of the cell - Cellullar Neural Networks (CNN) [9].

C. *Our sample - what is in the cell?*

Now, let us take a second look at the sketch of a system shown in Figure 3. Let us try to imagine some 3-dimensional

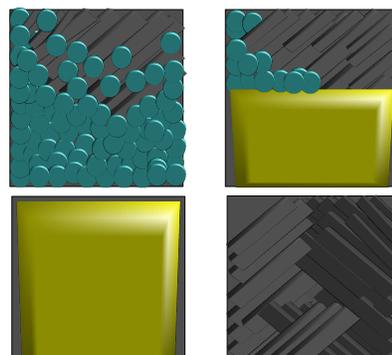


Figure 5. Four examples of what the cell sides may look like.

cells, in the simplest case let them be a one layer of cubes, as in Figure 4. Each cell contains a portion of the system which may contain SWCNTs and/or polymer molecules and/or golden electrodes. The cell dynamics within the system will be governed by the physics of its content. The cell interaction with its neighbouring cells will be governed by the sides where the cells touch. Figure 5 shows some examples of what the sides of the cell, in this case a cube, may look like. The top left figure shows an example with mainly polymer molecules and CNT bundles only in a right top corner; the top right an electrode with CNT bundles and a few polymer molecules; the bottom left an electrode; the bottom right only CNT bundles. If only polymer molecules are present, there will be no current flow between the cells as they are electrically isolators. If there are SWCNTs there, the current flow will be determined by the electrical properties of the SWCNTs - the percentage of metallic and semiconducting nanotubes, characteristic resistance per unit length, etc. If there is an electrode there, then the voltage it provides is known. Further, the sides of the cells may contain each of the elements to a certain percentage.

The behaviour of the cell is determined by the physics of the material the cell contains. When describing the physics that governs the cell dynamics, no detailed mechanisms of SWCNT-polymer nanocomposites behaviour are needed. Some level of abstraction may suffice so that interesting and useful behaviour is still captured although not all the segment and junction resistances and the currents in individual segments of percolation paths formed by SWCNTs are considered.

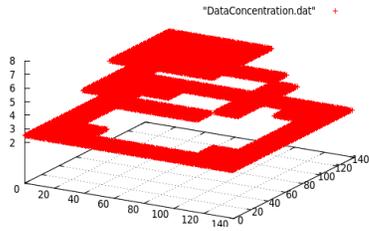


Figure 6. Plot of the concentration of SWCNTs in our model, see Section IV for explanation.

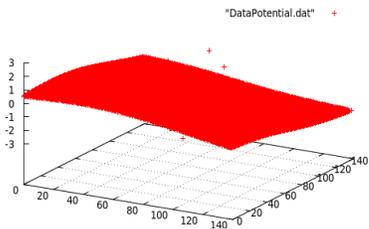


Figure 7. Plot of the electric potential resulting from the voltages on electrodes, stand-alone crosses correspond to the electrode voltages, see Section IV for further explanation.

IV. MODEL AND SIMULATION

Let us consider a very simple model which captures the properties described in Section III. A sample of material is given as a blob of SWCNT-polymer nanocomposites dropped over 16 electrodes placed in 4×4 array. Further, let us assume that such system has been diced into a number of cells, as in Figure 4. The cells are uniform and in a shape of a cube. Approximately $1.96 \cdot 10^4$ cells of $100 \times 100 \times 100 \mu m^3$ dimensions would account for a blob of $20 \mu L$ of nanocomposites typically used in our experiments. Let them be placed in a 140×140 grid of cells and position the array of electrodes in the centre of the grid.

Lab samples are characterised by the concentration of SWCNT. To make the model more realistic, let us assume that the concentration is not uniform for all the cells: the cells closer to the centre of the 140×140 grid have slightly higher concentration than the cells closer to the edges. Therefore, in our model, for a given concentration, we multiply this value with some weight which depends on the position of the cell within the grid, as shown in Figure 6, for a simple case where the concentration of 0.25 is multiplied by coefficients 1.5, 1 and 0.5 dependent on the cell position within the three 'rings' around the central cells which contain electrodes, 1.5 corresponding to the closest to the centre.

Electric field is the result of the voltages on electrodes. During EA runs, voltages on electrodes change thereby changing electric field as well. The electric field at certain location determines electric potential of the location. Without going into exact equations, let us assume that the value of the electric field at some position within the 140×140 grid is given as supersposition of the contributions from individual electrodes. Further, for each contribution, the value of the electric field

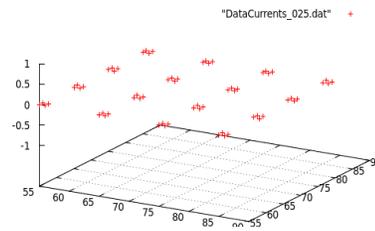


Figure 8. Plot of the simulated current for the SWCNT concentration 0.25% - only the cells which contain electrodes can conduct current but since no percolation paths are formed with other cells, there are no current flows.

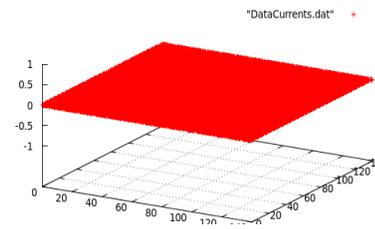


Figure 9. Plot of the simulated current for the SWCNT concentration 5% - all cells in the system conduct current.

is dependent on the distance of the cell from that particular electrode. Figure 7 shows a plot of the electric potential caused by the electric field which is determined in the described way for voltages on electrodes being some random numbers in the range $[0, 3]V$. For each location (i, j) in the grid, the potential is calculated as $v_{ij} = 0.01 \cdot \sum_{k=1}^{16} v_k \cdot \sqrt{(i - k_i)^2 + (j - k_j)^2}$, for an array of 16 electrodes in a 4×4 matrix.

The change of the states of the cells is governed by the physics of the material found in the cell. In our model, the cell state is described as the collection of the states of the cell sides $(v_i, i_i), i = 1, 2, \dots, 6$, the state of the cell side given as a pair of values (v_i, i_i) representing electric potential and current at its location within the grid. The choice to describe the state of the cell based on the states of its sides comes from the fact that conductivity of the sample is calculated based on the interactions between the neighbouring cells which can allow for the current conduction if sufficient concentration of CNTs is found at the cell boundaries so that conducting percolation paths can be formed.

Let us consider a simple case where the physics is determined purely by the concentration of the SWCNTs in the cell and some probability that the side of the cell will conduct. In particular, a random number from a uniform distribution is multiplied by the concentration of SWCNTs in the cell and if such product is greater than the midrange of the distribution interval, the current may flow and is set to 1, otherwise it is set to 0. Figures 8 and 9 show the conductance of the material for two corner cases when there is no current as no percolation paths are formed due to a low concentration of SWCNTs and when concentration of SWCNTs is so high that current flows across the whole of material.

V. DISCUSSION/CONCLUSION AND FUTURE WORK

In this paper, initial work on modelling conductivity of SWCNT-polymer nanocomposites has been presented. The novelty of the approach lies in it being based on two paradigms: dynamical hierarchies and a cellular computation model. Principles of hierarchies are used to find a suitable approach to abstracting away details of the physics at a very low level. A cellular computation model accounts for the interactions between units of material - cells.

Initial sketch of the model shows that when it is used in simulations, it yields the results which are intuitively expected: higher concentrations of SWCNTs in nanocomposites lead to more percolation paths formed and consequently more current flows; different voltages on electrodes produce different electric fields in the material; the material behaviour is based on some underlying physics which can be abstracted to a certain level, etc.

However, this initial sketch is still far away from the model of the behaviour of the materials used in NAsCENCE experiments - it needs to be extended! We identify main directions for future extensions:

- representation of the physics which drives the behaviour of the cell
- type of the cell which is best suitable

Could some of the existing models for CNT thin film transistors be used like in [19]? Or some of the models currently investigated within the project (not published yet)? Or some even higher level description? Simple cubes as presented at this place are far from the real blob of material sample. Some other shapes, e.g., dodecahedron are being considered. Also, the non-uniformity of the cells is likely to be suitable for successful models regarding size and content of the cell.

We are looking forward to addressing these challenges.

ACKNOWLEDGMENT

The research presented in this paper has been funded within EC 7FP under grant agreement no. 317662

REFERENCES

- [1] H. Broersma, F. Gomez, J. Miller, M. Petty, and G. Tufte, "Nasence project: Nanoscale engineering for novel computation using evolution," *International Journal of Unconventional Computing*, vol. 8, no. 4, 2012, pp. 313–317.
- [2] J. Miller and K. Downing, "Evolution in materio: looking beyond the silicon box," in *Proceedings of the NASA/DoD Conference on Evolvable Hardware*. NASA/DoD, July 2002, pp. 167–176.
- [3] S. Harding, J. F. Miller, and E. A. Rietman, "Evolution in materio: Exploiting the physics of materials for computation," *Int J of Unconventional Computing*, 2008, pp. 155–194.
- [4] B. K. Sarker, N. Kang, and S. I. Khondaker, "High performance semiconducting enriched carbon nanotube thin film transistors using metallic carbon nanotubes as electrodes," *Nanoscale*, vol. 6, 2014, pp. 4896–4902.
- [5] D.-M. Sun, C. Liu, W.-C. Ren, and H.-M. Cheng, "A review of carbon nanotube- and graphene-based flexible thin-film transistors," *Small*, vol. 9, no. 8, 2013, pp. 1188–1205.
- [6] T. Lenaerts, D. Gross, and R. Watson, "Introduction to the workshop wdh2002," in *Proceedings of the Alife VIII workshop On the modeling of dynamical hierarchies*, D. Gross and T. Lenaerts, Eds. Sydney: University of New South Wales, 2002, pp. 37–44.
- [7] S. Rasmussen, N. Baas, B. Mayer, M. Nilsson, and M. Olesen, "Ansatz for dynamical hierarchies," in *Artificial Life*. Massachusetts Institute of Technology, 2001, pp. 329–353.
- [8] E. F. Codd, *Cellular Automata*, ser. Association for computing machinery, Inc. Monograph series. Academic Press, New York, 1968.
- [9] V. Cimagalli and M. Balsi, "Cellular neural networks, a review," in *Proc. 6th Italian Workshop on Parallel Architectures and Neural Networks*, Vietri sul Mare, Italy, E. Caianiello, Ed. World Scientific, 1993, pp. 12–14.
- [10] G. Pask, "Physical analogues to the growth of a concept," in *Mechanisation of Thought Process*, symposium proceedings, 1958, pp. 877–923.
- [11] A. Thompson, *Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*. Springer, 1996.
- [12] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *Advances in Artificial Life*, ser. Lecture Notes in Computer Science, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. Kim, Eds. Springer Berlin Heidelberg, 2003, vol. 2801, pp. 588–597.
- [13] S. Harding and J. F. Miller, "Evolution in materio : A real-time robot controller in liquid crystal," in *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, J. Lohn, D. Gwaltney, G. Hornby, R. Zebulum, D. Keymeulen, and A. Stoica, Eds. Washington, DC, USA: IEEE Press, 29 June-1 July 2005, pp. 229–238.
- [14] K. Clegg, J. Miller, K. Massey, and M. Petty, "Travelling salesman problem solved in materio by evolved carbon nanotube device," in *Parallel Problem Solving from Nature PPSN XIII*, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, Eds. Springer International Publishing, 2014, vol. 8672, pp. 692–701.
- [15] O. Lykkebø, S. Harding, G. Tufte, and J. Miller, "Mecobo: A hardware and software platform for in materio evolution," in *Unconventional Computation and Natural Computation*, ser. Lecture Notes in Computer Science, O. H. Ibarra, L. Kari, and S. Kopecki, Eds. Springer International Publishing, 2014, pp. 267–279.
- [16] A. Kotsialos, M. Massey, F. Qaiser, D. Zeze, C. Pearson, and M. Petty, "Logic gate and circuit training on randomly dispersed carbon nanotubes," *International Journal of Unconventional Computing*, vol. 5, no. 6, 2014, pp. 473–497.
- [17] M. S. Fuhrer, J. Nygrd, L. Shih, M. Forero, Y.-G. Yoon, M. S. C. Mazzoni, H. J. Choi, J. Ihm, S. G. Louie, A. Zettl, and P. L. McEuen, "Crossed nanotube junctions," *Science*, vol. 288, no. 5465, 2000, pp. 494–497.
- [18] A. Behnam and A. Ural, "Computational study of geometry-dependent resistivity scaling in single-walled carbon nanotube films," *Phys. Rev. B*, vol. 75, Mar 2007, p. 125432.
- [19] A. Behnam, J. Guo, and A. Ural, "Effects of nanotube alignment and measurement direction on percolation resistivity in single-walled carbon nanotube films," *Journal of Applied Physics*, vol. 102, no. 4, 2007, p. 044313.
- [20] D. A. Jack, C.-S. Yeh, Z. Liang, S. Li, J. G. Park, and J. C. Fielding, "Electrical conductivity modeling and experimental study of densely packed swcnt networks," *Nanotechnology*, vol. 21, no. 19, 2010, p. 195703.
- [21] W. S. Bao, S. A. Meguid, Z. H. Zhu, and M. J. Meguid, "Modeling electrical conductivities of nanocomposites with aligned carbon nanotubes," *Nanotechnology*, vol. 22, no. 48, 2011, p. 485704.
- [22] Y. Bar-Yam, *Dynamics of Complex Systems*. Westview Press, 1997.
- [23] H. Simon, "The architecture of complexity," in *Proceedings of the American Philosophical Society*, vol. 106, 1962, pp. 467–482.
- [24] S. N. Salthe, *Evolving Hierarchical Systems*. Columbia University Press, 1985.
- [25] H. A. Simon, "The architecture of complexity: Hierarchic systems," in *The Sciences of the Artificial (Third edition)*, 1996, pp. 183–216.
- [26] H. H. Pattee, "The physical basis and origin of hierarchical control," in *Hierarchy Theory*, H. H. Pattee, Ed. George Braziller Inc., 1973, pp. 71–108.
- [27] J. L. Lemke, "Opening up closure: Semiotics across scales," in *Annals of the New York Academy of Sciences*. Wiley Online Library, 2006, p. 100.
- [28] M. A. Bedau and P. Humphreys, *Emergence, Contemporary Readings in Philosophy and Science*. MIT Press, 2008.

Is There Chaos in Blobs of Carbon Nanotubes Used to Perform Computation?

Stefano Nichele, Dragana Laketić, Odd Rune Lykkebø, and Gunnar Tuft
Norwegian University of Science and Technology
Trondheim, Norway
email: {nichele, draganal, lykkebo, gunnart}@idi.ntnu.no

Abstract—We report on observations of the behavior in single-wall carbon nanotubes (SWCNTs) and polymer nanocomposites, which may indicate that the conductivity of this nanomaterial undergoes chaotic changes under certain circumstances. We present interesting material properties that can be used for (a) investigating non-linearity / chaotic behavioral properties of the material, (b) understanding the underlying physics that may be explored by evolution, and (c) identifying appropriate signal types (voltages, frequencies) suitable to stimulate the material and demonstrate a computationally rich behavior. The presented results show that the investigated material is particularly sensitive to specific frequencies, e.g., square waves, and suggest that different behavioral regimes may be achieved, e.g., uniform, stable, chaotic. This work is done within the NASCENCE project, whose goal is to find new materials suitable to perform computation.

Keywords- *Computation-in-Materio; Evolution-in-Materio; Evolvable Hardware; Chaos.*

I. INTRODUCTION

Traditional silicon-based computers are meticulously designed with a conventional top-down process. The miniaturization and engineering of such processors poses technological and economical challenges. In contrast, Evolution-in-Materio [1] [2] is a bottom-up approach where the intrinsic underlying physics of materials is exploited as computational medium. Different computational substrates have been previously explored [3] [4] [5]. The EU-funded NASCENCE project [6] investigates how nano-scale materials, e.g., carbon nanotubes / polymer composites, liquid crystals, and graphene, may be used and configured to produce computation. As such, the material blob is treated as a black-box and is interfaced to a traditional computer through a series of signals / wires which send inputs / configurations and read / interpret outputs. Evolutionary algorithms are the means of finding suitable configurations in order to “program” the material to solve a wanted function. Such a black-box hybrid approach has been shown successful for different problems, such as Traveling Salesman [7], logic gates [8], bin packing [9], machine learning classification [10], frequency classification [11], and function optimization [12]. At the current state of research, it is not clearly understood what the exploited underlying physical properties are and what is the best way of exploring

them, e.g., number of input / outputs, types of signals (static voltages, sinusoidal waves, square waves). The solved problems serve as proof of concept that such an approach may be competitive in terms of computational time, size, and energy consumption, but scaling-up to solve bigger instances requires a better understanding. In other words the black-box needs to be opened. The number of used input electrodes, configuration signals, etc., is related to what part of the search and solution space is available for evolution to be exploited on the material, which may be critical for the success of any kind of computational task.

There are several parameters that may have impact on evolvability and computational power. Those can be grouped in three macro-categories:

- **Intrinsic:** this is related to internal properties of the molecules that compose the material, for example the type of used particles and their composition. This decides the physical properties that may be available to be exploited, e.g., conductivity, charge, etc.
- **External/Environmental:** external stimuli that influence the material properties, such as current, temperature, light. Those can be of two types: controllable, e.g., evolved, or non-controllable. External inputs may have an impact and change the physical state of the material temporarily or permanently.
- **Construction:** those are properties that are decided when the system is built and cannot be changed afterwards. As such, they may be said to be both external, i.e., construction choices, and internal, as they influence the intrinsic physical properties of the material, e.g., concentration of molecules, type of nanotubes (metallic, semi-conducting), electrodes material, size and pitch.

As there are several variables to be explored, it is reasonable to acquire a better understanding of how the most relevant parameters impact the computational power of materials. The more we know about materials, the better they can be used for computations (or any other purpose we may want to use them for).

This paper, which serves as a work-in-progress report, is organized as follows: Section 2 describes the investigated material and hardware platform. Section 3 relates the work to dynamic complex systems and chaos. In Section 4, the experimental setup is described and Section 5 presents the

results and discussion. Section 7 concludes the paper together with planned future work.

II. MATERIAL AND HW PLATFORM

The current experiments are performed on a blob of single-wall carbon nanotubes (SWCNT) mixed with polybutyl methacrylate (PBMA) dissolved in anisole (methoxy-benzene). The material samples, supplied by Durham University, are prepared on 4x4 grids of gold micro-electrode arrays with pads of 50µm and pitch of 100µm. The preparation is done by dispensing 20µL of material onto the electrode area (CNT concentration 0.53% of weight). This is baked for 30min at 90C; the solvent dries out and leaves a “thick film”. The substrate is cooled slowly over a period of 1h. This process leaves a variable distribution of nanotubes across the electrodes. Carbon nanotubes are 30% conducting and 70% semi-conducting, while PBMA creates insulation areas within nanotube networks, which may allow non-linear current versus voltage characteristics.

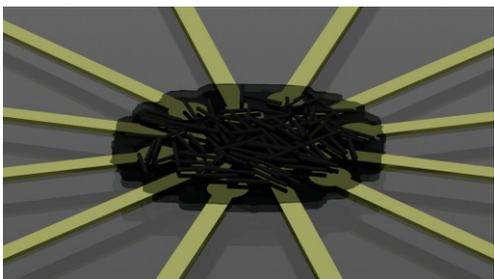


Figure 1. Sketch of nanotubes dispersed over electrodes.

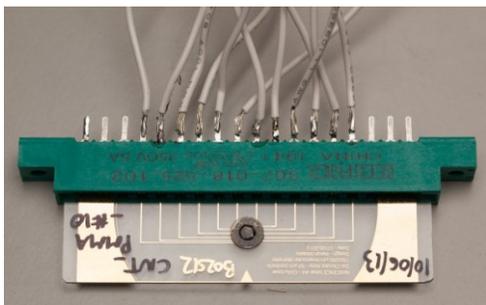


Figure 2. Electrode array slide plugged into connector.

A sketch of randomly dispersed nanotubes over gold electrodes on a glass slide is shown in Figure 1. The glass slide on which the material and electrodes are placed is plugged into a connector as represented in Figure 2. Such connector is attached to a custom built hardware board called Mecobo [13] (designed and built at the Norwegian University of Science and Technology), which is connected to a host PC via USB. Such platform can be accessed remotely over the internet by a Thrift server. Figure 3 shows a Mecobo board with a connected material slide. An overall block diagram of the system is presented in Figure 4. Mecobo serves as interface for computer-controlled evolution of configuration signals (analog or digital) to

stimulate the nanomaterial to solve computational problems. For more details on the Mecobo platform see [13].



Figure 3. Mecobo interface between nanomaterial and PC.

In a typical experiment, the overall goal is to solve a computational task. Output signals are sampled from the material by the board and returned to the host computer which executes an evolutionary algorithm and calculates the fitness. The maximum sampling frequency is 50KHz. The input signals may be static voltages or square waves ranging between 400Hz and 25MHz. As the material is treated as a black-box, the input stimuli are used to exploit the underlying physics of the material blob. On the other hand, if we want to unveil the computational power of materials, how to better exploit them, and how to use them to solve problems of scaled-up complexity, a different approach is needed. As such, in the experiments herein we connect material slides to a Hewlett Packard 33120A 15MHz function / arbitrary waveform generator (used as input) and an Agilent 54622D 100MHz mixed signal oscilloscope (used as output).

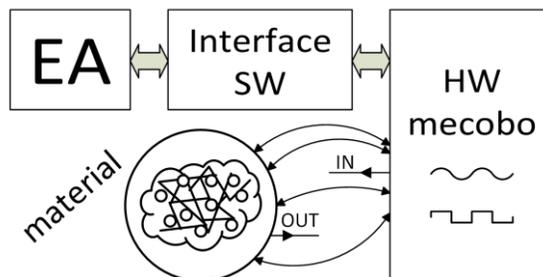


Figure 4. Experimental system block diagram.

III. COMPLEX DYNAMIC SYSTEMS AND CHAOS

Carbon nanotubes randomly dispersed in polymer solutions may be considered as a complex dynamic system where a huge number of tiny elements (nano-molecules)

interact at a local level and exhibit different emergent dynamics [14]. Such an emergent process cannot be understood by describing one of its parts alone but must be considered together with the massively parallel interactions among its parts. The idea of complex dynamic systems is connected with the notion of “edge of chaos” [15] [16], which may indicate maximum complexity and computational power. Computation may be said to occur in the vicinity of a phase transition between order, i.e., little dynamics / information processing and high memory / structure preservation, and chaos, i.e., no memory and plenty of dynamics. It may be argued that in order to have any complex computation, a balance between order and chaos needs to exist, i.e., edge of chaos. Computation at the molecular level, i.e., computation-in-materialio, may be able to produce considerably rich dynamics as the very essence of the material physics is exploited. This may allow to abstract computational properties and analyze the dynamics of the investigated substrates as trajectories and attractor basins [17].

IV. EXPERIMENTAL SETUP

Evolving solutions to computational problems in the material requires interfacing to a computer (EA). Such interface is typically provided by the Mecobo board. However, when it comes to understanding the underlying properties of the material and relative responses, it may be necessary to use specific electronic test instruments, i.e., oscilloscopes and signal generators.

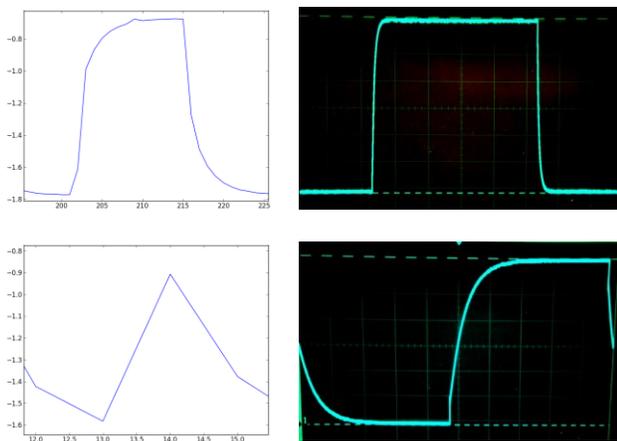


Figure 5. Mecobo (left) vs. oscilloscope (right) at 1KHz (top) and 10KHz (bottom). Input $\pm 3.3V$, duty cycle 50%.

Figure 5 presents the same square wave ($\pm 3.3V$, duty cycle 50%) at a frequency of 1KHz (top) and 10KHz (bottom), sampled with Mecobo (left) at a frequency of 25KHz and probed with the oscilloscope (right). It is visible that the material may show a charge / discharge transition, but the level of details is partially lost with Mecobo. As the operating input frequency may range between 400Hz and 25MHz, while the maximum sampling rate is 50KHz, it is clear that physical properties of the material need to be

investigated with an oscilloscope. On the other hand, when solutions ought to be discovered by an evolutionary algorithm, Mecobo acts as an interface, being able to produce different inputs, i.e. analog and digital, on different pins and read outputs at the same time.

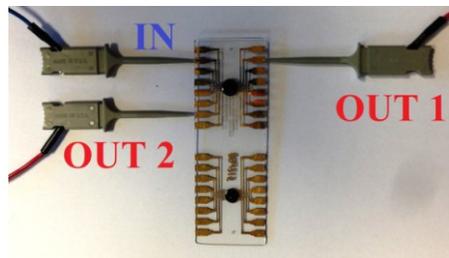


Figure 6. Schematics without Mecobo.

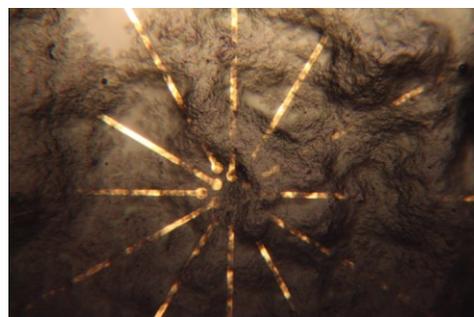


Figure 7. SEM image of gold electrode arrays with different coverage of nanotubes. Adopted from [8].

The chosen experimental setup is shown in Figure 6, where the input probe (from the signal generator) is placed on pin #2 (IN) and the two output probes (to the oscilloscope) are connected to pins #9 (OUT1) and #7 (OUT 2). The input / output pins have been chosen as there is the same electrode distance between gold pads within the material blob. Coverage of gold electrodes with randomly dispersed nanotubes varies and some of the electrodes may even be left with little or no coverage, as visible in the Scanning Electron Microscope (SEM) image in Figure 7.

One of the expected results is to achieve different responses on different output pins, as the material from a macroscopic point of view has the following properties:

- Varying capacitance between pin pairs;
- Varying inductance between pin pairs;
- Varying induced magnetic field, inducing current in other parts of the material;
- Varying resistance between pin pairs;
- Varying metal/semiconductor junctions, e.g. different coverage of carbon nanotubes on gold probes.

V. RESULTS AND DISCUSSION

Even if a single square wave input signal is used, the resulting output shows a rich variety of behaviors, more than

what may be achieved by a single static voltage or by a sinusoidal wave. As such, square waves may be better suited to penetrate the material and exploit the nanotube sub-networks, which may be particularly sensitive to different frequencies.

Figure 8 presents the experimental results. In particular, Figures 8(a) show several snapshots of the material response on two different pins at different frequencies, ranging from 1KHz (Figure 8a1) to 14MHz (Figure 8a12). At 1KHz the signals may seem similar (a1), where the material charges-up and subsequently discharges, but at a zoomed-in resolution (a2) a voltage spike is visible on the second probe which is not present on the first probe. This is better visible at 5KHz (a3), 30KHz (a4) and 100KHz (a5), where it is possible to notice that on the rising front there is a sudden voltage increase/drop, as the material behavior is “supercapacitor-like”. Starting from 500KHz (a6), which is also zoomed-in (a7), the second probe signal is similar to a square wave (most of the harmonic frequencies are passed) while the first probe acts more like a filter. In both cases, there is a resonance phase which results in a deterministic yet semi-chaotic waveform. This may be the effect of some conducting sub-networks in the material that are enabled at specific frequencies and disabled at others. At 2, 5 and 8.5 MHz the measured voltage decreases while frequency increases. At 10MHz (a11) a strange phenomenon is observed where both signals (it is more noticeable on the first) have a voltage increase, probably due to a feedback-effect (some frequencies are fed again into the material by some nanotube sub-networks). At 14MHz (a12) the signal on the second probe is sinusoidal, i.e., only one harmonic frequency is present. As such, it may be concluded that with a single square wave input it is possible to observe a rich variety of behaviors while the frequency spectrum is traversed.

As the system produces uniform, stable and semi-chaotic behaviors, it is of particular interest to visualize input-output responses and output-output relations, as to better understand traversed trajectories and attractors. For this purpose, XY plots are shown in Figure 8b, where OUT1 is plotted against OUT2 and Figure 8c, where IN is plotted against OUT1. In Figure 8b1, some orbits are present at 30KHz. Similar orbits are visible at 60KHz (b2) and 100KHz (b3), moving towards opposite corners to those where the impulse is. After each impulse, there is a semi-chaotic orbit which relaxes before the next impulse arises, as the semi-chaotic behavior is annihilated by the lack of energy in the material, until the next impulse. This may suggest that chaotic behavior may be present, yet particularly difficult to observe.

Finally, we present XY plots between input and output, which represent the phase space of the system (input-output pin pair). Figure 8c1 is registered at 350KHz. Several oscillating orbits are present, which are zoomed-in at 2MHz (c2). Same effect is observed for frequencies up to 5MHz (c3) while for frequencies around 10MHz and higher we observe a hysteresis loop which may indicate that some saturation may have been reached in the material. Some sort of non-linearity seems present, which is always a good indicator that the system may achieve chaotic behavior.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented interesting properties of nanocomposites of single-wall carbon nanotubes and polymers. Such material blobs are investigated within the NASCENCE projects as one of possible novel computational substrates. Previous work has presented proof of concept for the solution of several problems, where the material blob was used as a black-box and interfaced to a computer running EAs through a custom developed board. Our approach was intended for investigating different behavioral responses when the material is stimulated with square wave voltages at different frequencies, with the goal of detecting non-linear and chaos-like behaviors. As such, signal generator and oscilloscope’s probes were connected to the material gold electrodes. This was done to understand underlying physical responses and suitable signal frequencies able to show computationally rich behaviors. The work in progress results are promising as a single square wave signal has been shown to produce a variety of responses dependent on pins and frequencies. Phase space plots have been presented with orbiting trajectories alternated to relaxed transients. Responses that may be exploited for evolution have been identified at rather low frequencies, in ranges below MHz.

As future work we will continue our investigation to search for interesting responses and new methods for their analysis, in particular with the goal of determining whether chaotic behavior is present. The presence of chaos may be considered a benefit of the used computational materials, as the described theory on the computation at the “edge of chaos” may be plausible for computation at the molecular level, where the low physics of material are exploited.

Another future direction is the modeling of different materials at different abstraction levels. Currently, cellular automata (CA) models of materials are considered. Such an approach has the advantage of representing different computational behaviors as trajectories and attractor basins, allowing measurements on computational complexity and computational power of materials.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP/2007-2013) under grant agreement number 317662.

REFERENCES

- [1] J. Miller, S. Harding, and G. Tufte, “Evolution-in-materio: evolving computation in materials,” *Evolutionary Intelligence*, vol. 7, no. 1, 2014, pp. 49–67.
- [2] J. Miller and K. Downing, “Evolution in materio: Looking beyond the silicon box,” in *The 2002 NASA/DoD Conference on Evolvable Hardware*, A. Stoica, J. Lohn, R. Katz, D. Keymeulen, and R. S. Zebulum, eds., Alexandria, Virginia, Jet Propulsion Laboratory, California Institute of Technology, IEEE Computer Society, 15-18 July 2002, pp. 167–176.
- [3] A. Thompson, “Hardware evolution - automatic design of electronic circuits in reconfigurable hardware by artificial evolution,” CPHC/BCS distinguished dissertations, 1998.

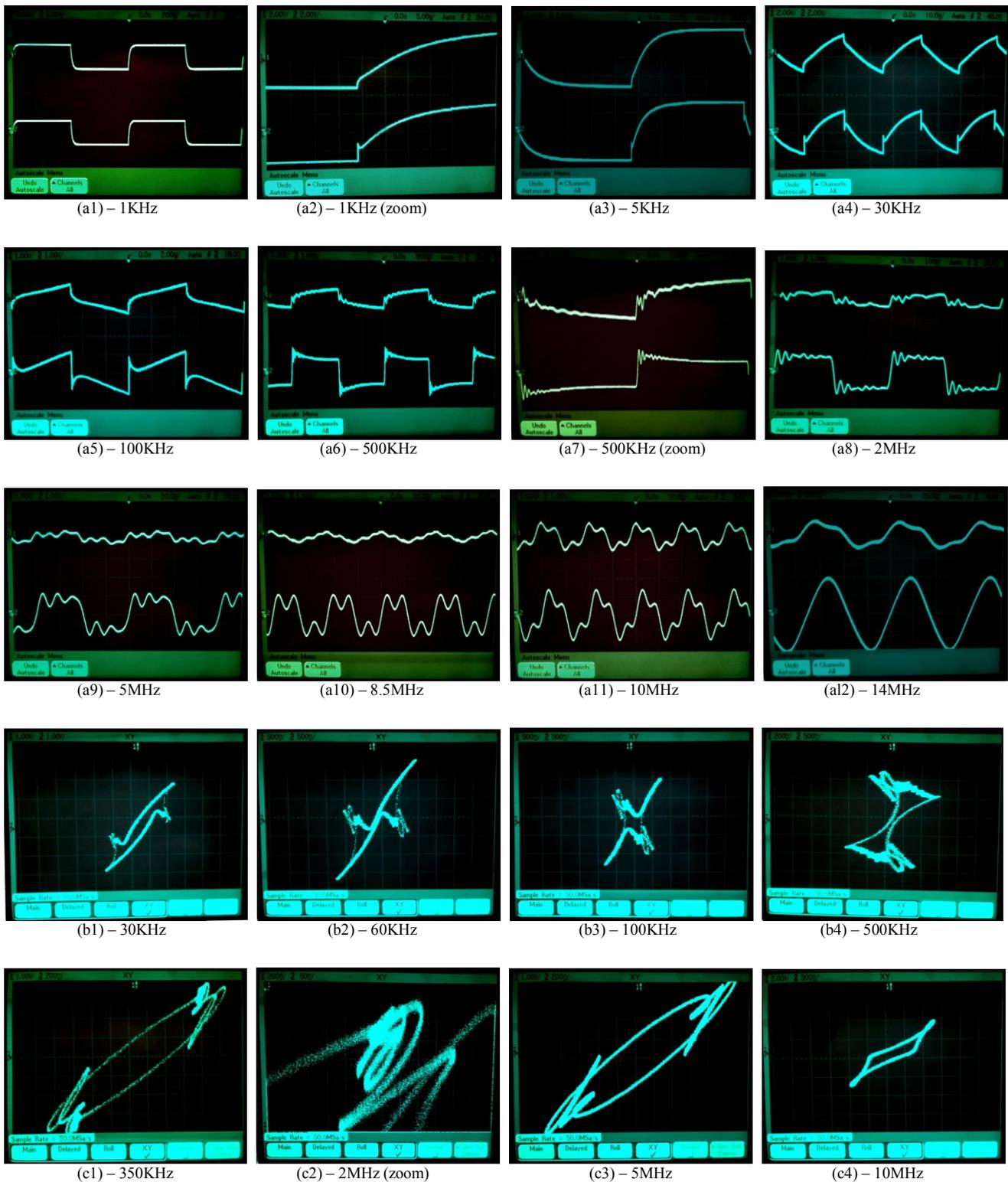


Figure 8. Oscilloscope screenshots.
 (a) Voltage responses on 2 different pins with input square wave at different frequencies.
 (b) XY plots, X (OUT1) is plotted against Y (OUT2) at different frequencies.
 (c) XY plots, X (IN) is plotted against Y (OUT1) at different frequencies.

- [4] S. Harding and J. Miller, "Evolution in materio : A real-time robot controller in liquid crystal," in Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware, J. Lohn, D. Gwaltney, G. Hornby, R. Zebulum, D. Keymeulen, and A. Stoica, eds., Washington, DC, USA, IEEE Press, 29 June-1 July 2005, pp. 229–238.
- [5] P. Cariani, "To evolve an ear: epistemological implications of Gordon Pask's electrochemical devices," *Systems Research*, vol. 10, no. 3, 1993, pp. 19–33.
- [6] H. Broersma, F. Gomez, J. Miller, M. Petty, and G. Tufte, "Nascence project: nanoscale engineering for novel computation using evolution," *International journal of unconventional computing*, vol. 8, no. 4, 2012, pp. 313–317.
- [7] K. Clegg, J. Miller, K. Massey, and M. Petty, "Travelling salesman problem solved "in materio" by evolved carbon nanotube device," in *Parallel Problem Solving from Nature - PPSN XIII*, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, eds., vol. 8672 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 692–701.
- [8] A. Kotsialos, K. Massey, F. Qaiser, D. Zeze, C. Pearson, and M. Petty, "Logic gate and circuit training on randomly dispersed carbon nanotubes," *International journal of unconventional computing*, vol. 10, September 2014, pp. 473–497.
- [9] M. Mohid et al., "Evolution-in-materio: Solving bin packing problems using materials," in *The 2014 IEEE Conference on Evolvable Systems - ICES*, IN PRESS, IEEE Computer Society, 2014.
- [10] M. Mohid et al., "Evolution-in-materio: Solving machine learning classification problems using materials," in *Parallel Problem Solving from Nature - PPSN XIII*, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, eds., vol. 8672 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 721–730.
- [11] M. Mohid et al., "Evolution-in-materio: A frequency classifier using materials," in *The 2014 IEEE Conference on Evolvable Systems - ICES*, IN PRESS, IEEE Computer Society, 2014.
- [12] M. Mohid et al., "Evolution-in-materio : solving function optimization problems using materials," in *Computational Intelligence (UKCI) : 2014 14th UK Workshop on*, 8-10 September 2014, Bradford, UK ; proceedings. (D. Neagu, M. Kiran, and P. Trundle, eds.), IEEE, September 2014, pp. 1–8.
- [13] O. R. Lykkebø, S. Harding, G. Tufte, and J. Miller, "Mecobo: A hardware and software platform for in materio evolution," in *Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014*, London, ON, Canada, July 14-18, 2014, Proceedings, pp. 267–279.
- [14] S. Stepney, "The Neglected Pillar of Material Computation," *Physica D: Nonlinear Phenomena*, vol. 237, July 2008, pp. 1157–1164.
- [15] C. Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation," *Phys. D*, vol. 42, June 1990, pp. 12–37.
- [16] M. Mitchell, J. Crutchfield, P. Hraber, "Dynamics, computation, and the "edge of chaos": A re-examination," in *Complexity: Metaphors, Models, and Reality*, Addison-Wesley, 1994, pp. 497–513.
- [17] S. Nichele and G. Tufte, "Trajectories and attractors as specification for the evolution of behaviour in cellular automata," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, IEEE, 2010, pp. 1–8.

Specification and Verification of Garbage Collector by Java Modeling Language

Wenhui Sun, Yuting Sun, Zhifei Zhang
 Department of Computer Science and Technology
 Beijing Jiaotong University
 Beijing, China
 {whsun1, ysun, zhfzhang}@bjtu.edu.cn

Jingpeng Tang
 Department of Computer Science
 Utah Valley University
 Oren, Utah
 JTang@uvu.edu

Kendall E. Nygard, Damian Lampl
 Department of Computer Science
 North Dakota State University
 Fargo, ND, USA
 {kendall.nygard,damian.lampl}@ndsu.edu

Abstract— The Java garbage collector effectively avoids some security holes and improves the utilization rate of resources. Guaranteed reliability of the garbage collector is a challenge due to the complexity of the interaction between the collector and the user program; the highly abstracted garbage collector algorithms cannot reflect the real implementation details. System complexities have allowed dynamic analysis based on Design by Contract (DBC) to become an important method for ensuring software quality. Java Modeling Language (JML) inherits all the advantages of contractual design, and became a behavior interface specification language for Java. JML can be used to regulate module behavior and detailed design of Java programs. In this paper, we discuss the JML specifications for the functional requirements of the garbage collector in Hoare-style. This approach can improve the reliability and correctness of the software system in the extent of real environments and run-time.

Keywords- design by contract, Java Modeling Language, garbage collector.

I. INTRODUCTION

A large number of existing research works have addressed validating various types of garbage collection algorithms. However, these works focus on highly abstract algorithms with little work on implementations. Birkedal, Torp-Smith, and Reynolds gave an informal proof of a copying garbage collector [1]. Russinoff mechanically verified an incremental garbage collector under abstracted memory and a user program without actual environmental and implementation details. Validated algorithms are not equivalent to executable programs [2]. Lin, Chen, and Li verified the incremental stop-the-world mark-sweep garbage collector, which is more complex [3]. However, the interactions between a user program and garbage collector do not typically exist for a stop-the-world garbage collector.

Java Garbage Collector is an important component of a software system that can effectively avoid dangling pointer bugs, memory leaks, double free bugs, and can improve memory utilization [4]. Multi-threading makes possible for a

concurrent incremental garbage collector. However, compared to a stop-the-world garbage collector, it is more complex and the reliability issue is more challenging. Java is an object-oriented language with inheritance, polymorphism, and dynamic binding properties. The program execution is no longer simply based on static typing and must now accommodate dynamic typing, meaning the type will not be known until execution time. This will introduce extra complexity to ensure program correctness. If the garbage collector process produces errors or exceptions, the user program will run into unpredictable consequences. Therefore, ensuring the reliability of the garbage collector is extremely important.

Many practitioners utilize Design by Contract (DBC) to improve software quality. Java Modeling Language (JML) is a DBC implementation in Java, and is also a precise formal specification language for Java programs [5][6]. JML can accurately describe functional requirements and generate efficient testing cases which can avoid ambiguity and inaccuracies caused by natural language [7]. Formal interface specifications written in JML can also encourage automated testing.

This paper discusses the validation and verification of an incremental mark-sweep garbage collector. The security interaction between the garbage collector and the user program is accurately described by applying JML precondition, postcondition, and invariants in Hoare-style logic. The JML runtime assertion will automatically perform formal verification to ensure the correctness of the garbage collector. This study focuses on real environment memory objects. The JML specification covers both normal and abnormal behavior, which can accurately describe the real-time environment. The assertions are runtime execution, thus they can effectively handle polymorphic, inheritance and dynamic binding for object-oriented software. In our approach, program execution is not only the result of a function generation process, but also an assertion checking process. This approach can improve correctness and

reliability for the garbage collector, quickly position errors, and handle abnormal behavior during collection.

The main contributions of this paper are listed as follows:

(1) Using JML to verify the incremental mark-sweep garbage collector. JML, a Hoare-style syntax for pre- and postconditions and invariants, is a DBC implementation in Java. If the inputs meet the requirements, we should get the expected outputs. In more detail, if we take the parameters as inputs and returns as the outputs, then the responsibility of the caller (client) is to ensure that the correct parameters are provided, while the obligation of the supplier is to ensure the correct results are returned.

(2) Verifying the write barrier of the garbage collector using JML. The verification can avoid incorrect operation due to memory access and modification by the user program, to improve the correctness of the interaction between the garbage collector and the user program.

(3) Improve simplicity and understandability of the program function code. By separating the original program function code from the DBC checking code using JML, the program function code no longer mingles with DBC code block, thus avoiding unnecessary confusion. Also, the postcondition failure can easily locate errors. Improvement of algorithm reliability, as well as code understandability, can be achieved.

The rest of the paper is organized as follows: Section 2 describes JML and examples. Section 3 describes the garbage collector and write barrier algorithm using JML. Section 4 establishes the capabilities of the garbage collector in detail. Finally, Section 5 points out the conclusion and future work.

II. INTRODUCTION TO JML

A. Features of JML

JML specifications are written as Java annotation comments in the source files, and can be compiled with any Java compiler. These specifications are more abstract without logic implementation and thus can increase the modularity and accuracy of the source code [7][8]. By using DBC ideas, JML inherits all of its advantages, and is an excellent specification language:

(1) Documentation. JML provides semantics to formally specify interface, behavior, and detailed design. Java modules with JML specifications can be compiled with any Java compiler, making JML well suited for documenting reusable components, libraries, and frameworks [8].

(2) Clear Obligation. Pre- and postconditions separate the obligation. A precondition error indicates that the user's input does not meet the conditions, while a postcondition error indicates the procedural methods do not meet the requirement [9][10].

(3) High Efficiency. Since each execution of pre- and postconditions checks will consume resources, JML can turn off these checks to avoid unnecessary consumption of resources. This mechanism can decrease the cost of debugging and testing.

(4) Modular Reasoning. JML is abstract so that by reading the formal specification of a method its function is understood with no need to go inside other referenced methods (JML modular reasoning). JML modularity brings the benefits of easy understanding, but shields the details. The user will cannot understand the contents due to the lack of corresponding information.

In addition, the quantifier, specification inheritance, and pre-process can make the specification more accurate. Java modules with JML specifications can describe a method or class's anticipated behavior, without affecting the normal code while compiling. This can provide an early detection of incorrectness to improve the security of a system. Finally, Java modules with JML specifications can be compiled unchanged with any Java compiler. Various verification tools, such as a runtime assertion checker and the Extended Static Checker (ESC/Java) are available to aid the development. If the program does not implement the specification, JML throws an unchecked exception to explain that the program violates the specification.

B. JML Syntax and Examples

JML is a behavioral interface specification language for Java modules. JML provides semantics to describe the behavior of a Java module, preventing ambiguity with the module designers' intentions. Developers use JML to write classes and interfaces in the form of specifications. Each of the methods and interfaces written in accordance with the functional requirements is a JML formal specification. Developers must consider specific context of the system within which the method is running. The more precise the specification is, the more correctness will be achieved.

The example below illustrates JML usage and how it ensures reliability of a program. Assume class CustomerManager can manage all customer information of class BasicCustomerDetails. CustomerManager provides users with the add() method to create new customers. When a clear requirement of the add() method is available, we can develop JML specification as follows:

```

/*@ public normal_behavior
   @ public invariant count>=0;
   @ requires !theManager.isActive(theCustomer);
   @ assignable theManager;
   @ ensures
   @   theManager.count==old(theManager.count+1);
   @   theManager.isActive(theCustomer);
   @*/
public void add(BasicCustomerDetails theCustomer);

```

Figure 1. JML example.

JML invariant assertion count is always greater than or equal to zero. Class CustomerManager's invariant count is

true under all circumstances. In line 3, the keyword requires starts the precondition followed by a precondition assertion, @requires !theManager.isActive(theCustomer);

The precondition has to be true; otherwise, the caller is not able to call this method. This shows that in order to legally call add() to add theCustomer, theCustomer.id should be inactive. This will be asserted during runtime. Keyword assignable can modify the variable theManager. Keyword ensures introduces the postcondition which should be true after the execution, otherwise there are errors in the implementation of the add() method. In this case, the postcondition includes two assertions, @theManager.count==\old(theManager.count+1); @theManager.isActive(theCustomer);

The first assertion ensures count is incremented by 1. The expression \old indicates that the count value is the value before calling add(). The second assertion indicates the customer ID is now active. The pre- and post-conditions are specified as:

- (1) If the customer ID is already active, the same customer cannot be added;
- (2) Increasing the customer count will make the customer ID active.

Violation of either one or both will be considered as illegal and prohibited. If the add() method implementation did not follow JML specification, we would get error debugging feedback like the following:

By reading the debug feedback, we can get the following information:

- (1) The application is stopped in an object;
- (2) The object is theManager of class CustomerManager;
- (3) The error occurred when calling the add() method;
- (4) The error is a violation of a precondition of add();
- (5) The violation is isActive;
- (6) The BasicCustomerDetails object (theCustomer) caused the error when passed as an argument;
- (7) The call sequence causes the problems: The changeCustomer method of the CustomerManagerUif class (Customer Manager user interface) calls the add() method of the CustomerManager class;

In summary, we conclude that the CustomerManagerUif class's changeCustomer method is the problem: it is trying to add() an activated ID of the BasicCustomerDetails object, which is illegal.

In supporting the design by contract without slowing down the program execution, the contract testing can be manipulated by turning it on or off according to customer need.

III. MARK-AND-SWEEP GARBAGE COLLECTION

A. Garbage Selection Algorithm

The mark-and-sweep algorithm is based on tracing through the working memory, which includes a mark phase and sweep phase. In the mark phase, the collector does a tree traversal of the entire “root set”, marking all reachable objects, while the remaining memory cells are unreachable. During the sweep phase, unreachable objects are returned to the free list. The most notable disadvantage is that the entire system must be suspended during collection, also known as a stop-the-world event. In order to avoid this halting interruption, we adopted an interleaved garbage collector and user program which is called incremental collection. In our approach for JML specification, we also adopted tri-color marking, which divides the heap node into black, gray, and white sets. The tri-color method can be performed “on-the-fly”, without halting the system for significant time periods.

- (1) The black set is the set of reachable objects that the garbage collector has visited and all their referenced objects.
- (2) The gray set is the set of reachable objects the garbage collector has not visited; or, visited but not all their referenced objects; or, the reference relationship has been changed by the user program.
- (3) The white set is the set of unreachable objects the garbage collector has not yet visited. At the end of the tracking phase, these are the garbage.

The garbage collection process is divided into mark, sweep, and idle phases. During the mark phase, each object in memory has a flag (a single bit) reserved for garbage collection and a stack data structure to achieve the tri-color abstraction: 1) a marked object not in the stack is considered black; 2) a marked object in the stack is considered gray; 3) an unmarked object not in the stack is white. Although additional data structures needed for the mark phase will increase the memory space required for the collector, they will also shorten the time used for marking survived objects in the stack. If the stack is not empty, every time a gray object is visited, the garbage collector will mark all the non-black objects that are referenced by the current object to gray, and mark the current object to black. This process will continue until the number of visited objects meet the thresh value, and go to the sweep phase. In the sweep phase, not only the white objects are recycled, but also all the black objects are marked white for the next round before entering the idle phase. Once the empty space in the stack is less than the thresh value, a new mark phase is started again.

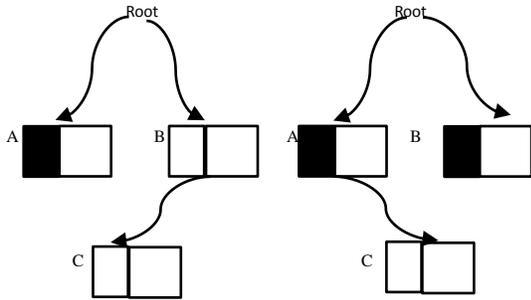


Figure 2. User programs violate garbage collection.

B. The Write Barrier

The role of the write barrier is to prevent error caused by a changed reference graph from the user program. Figure 2 shows the user program has changed pointer A to point to object C, and we do not know if there are other references to object B. If the objects B and C at the end of the mark phase are still white, then the garbage collector must ensure they are marked, otherwise they will be treated as garbage in the sweep phase. In this case, an object needs to be protected by a write barrier. Otherwise, there must be an object that is still reachable by the user program but is marked white. Thus, two conditions must be met at marking phase:

- (1) A reference to a white object is written to a black parent object, and this is the only reference to this white object.
- (2) The original reference to the white object is eliminated.

An object is retained if one or both of the conditions failed. We used Dijkstra’s algorithm for color updating. Every time a reference to a white object is created, regardless of the color of its parent, this white object is marked gray. When the collector traverses the heap, there will be no reference from a black object to a white object. For a reachable white object, there must be a path from gray to white. When the collector traverses the stack, condition 1 will fail. This is the solution for the communication between the user program and the collector.

IV. GARBAGE COLLECTOR JML SPECIFICATION

The JML specification for the garbage collector is discussed in this section. Assume the *pointerSet* is the memory set for the garbage collector, including white, gray, black, and free sets. The *freeList* is a linked list used to store the recycled idle objects. The *stack* together with the flag bit is used for marking the objects. The detailed JML specification of the garbage collector is as follows:

```

/*@ require p.getAddr()>=ST&&p.getAddr()<=ED;
   @ assignable p.color,stack;
   @ ensures p.getColor()==Color.BLACK;
   @ stack.peek()==p;
   @ stack.count==old(stack.count+1);
   @ also
   @ requires p.getAddr()<ST|&p.getAddr()>ED;
   @ assignable \nothing;
   @ signals_only IllegalArgumentException;
   @*/
public void markField( /*@non_null @*/ Pointer p);

```

Figure 3 JML specification for the markField.

The *markField* function has a constraint that the *non_null* parameter passed is not an empty pointer. If the pointer address is not within the address space managed by the collector, *assignable \nothing* cannot modify the stack, otherwise it is painted gray. According to the coloring, set *p* flag and push to the stack. The function returns the top element, *stack.peek()* which is *p*, and the number of elements in the stack increases by 1, which is *old(stack.count)* plus 1.

```

/*@requires phase==Phase.MARK&&stack.count==0;
   @ assignable Phase;
   @ ensures phase==Phase.MARK;
   @ \not_modified(stack);
   @ \not_modified(pointSet);
   @ \not_modified(freeList);
   @also
   @ requires phase==Phase.MARK&&stack.count>0;
   @ (\forallall Pointer p;
   @ pointerSet.contains(p)&&p.getcolor()==Color.BLACK;
   @ p.accessible(root));
   @ invariant NumMark>=MARKNUM&&NumMark<=MARKNUM;
   @ assignable Phase,stack,pointSet,color,pointSet.Field;
   @ ensures (numMark==MARKNUM&&phase==Phase.MARK);
   @ (\forallall Pointer p;
   @ pointerSet.contains(p)&&p.getcolor()==Color.BLACK;
   @ p.getField1().getcolor()==Color.BLACK&&p.getField1()==stack.peek()&&
   @ p.getField2().getcolor()==Color.BLACK&&p.getField1()==stack.peek());
   @*/
public void mark();

```

Figure 4. JML specification for the mark function.

The precondition of the *mark()* function is that garbage collection is in the mark phase, and the operations are dependent on the status of the stack. When the stack is empty, that means no nodes need to be visited, and no operations need to be done, thus the garbage collection goes to the sweep phase directly. If the stack is not empty, the mark stage is started. The assumption is that all the black and gray objects are reachable by the root. Each time a top element (a black node) is popped, all its referenced objects need to be marked gray, and the number of marked objects increases until reaching the threshold value. In the mark phase, the pop and push operations modify the stack, the flag, and the address space, while other variables that are not declared in the *assignable* remain unmodified. This ensures that the mark process did not modify the user information, idle list, or the current sweep position in the main memory.

```

/*@ invariants numSwept>=0&&numSwept<=SWEEPNUM;
@ invariants sweepCur>=START&&sweepCur<=END;
@ requires phase==Phase.SWEEP&&START<=p.getAddr()<=END;
@ assignable freeList.pointSet.color;
@ ensures numSwept=SWEEPNUM;
@ (\forallall Pointer p;START<=p.getAddr()<=sweepCur&&
@ \old (p.getColor()==Color.BLACK);p.getcolor==WHITE)
@ (\forallall (Pointer p;sweepCur<=p.getAddr()<=END&&
@ \old (p.getColor()==Color.BLACK);p.getcolor==BLACK)
@ (\forallall (Pointer p;START<=p.getAddr()<=sweepCur&&
@ \old(p.getColor()==Color.BLACK);p.getcolor==WHITE&&
@ freelist.getlast()==p&&p.freeField<==>TRUE);
@also
@ requires phase==Phase.SWEEP&&p.getAddr()>=END;
@ ensures phase==phase.IDLE;
@*/
public void sweep();

```

Figure 5. JML specification for the sweep function.

Similar to the *mark()* function, the precondition for the *sweep()* function is that garbage collection is in the sweep phase. Its task is to recycle a certain number (*SWEEPNUM*) of garbage. The *invariant* keyword describes, during sweep the *numSwept* (already swept objects) and *sweepCur* (current sweep address) should both be within the valid range. Changing in *\old(p.getColor)* (before call) and

p.getColor (after call) means all the objects before *sweepCur* are swept. For the objects after *sweepCur*, *p.getColor* remaining unchanged means the objects were not visited. By setting the second identification bit *p.freeField* we can check whether the garbage is in the free list. The assertion *Freelist.getlast () == p* verifies if it is indeed recycled in the free list. Finally, all the visited black objects are marked white for the next round sweep.

Garbage collector calls the corresponding functions according to *phase* status. Whether to start the next round of mark-sweep is based on the number of idle objects (*FREENUM*). The garbage collector constantly monitors the memory to make judgments. When the size of the free list, *freelist.size*, is less than the threshold, the *start_marking* function sets *phase* to mark status. The *sweepCur* starts scanning from low address and marks the root node to mark phase. Its JML specification is as follows:

```

/*@ requires freelist.size()<FREENUM&&phase==Phase.IDLE;
@ ensures sweepCur==START;
@ phase=Phase.MARK;
@ root.color==Color.BLACK;
@ root==stack.peek();
@ stack.size=\old (stack.size+1);
@*/
public void startMarking();

```

Figure 6. JML specification for start_marking().

In order to make the garbage collector and the user program interact properly, the *allocate()* function needs not only to allocate space to the user program from the free list, but also needs to avoid treating objects as garbage in the unswept address segment. After assigning space to the user program (*\result* is assigned starting address), the length of the free list and the number of the idle objects (*numfree*) will both be reduced. Marking objects after *sweepCur* black can ensure the unprocessed objects are not treated as garbage.

```

/*@ requires freelist.size()>0;

@ assignable freelist.pointSet.color;

@ ensures \result=\old(freelist.getFirst());

@      freelist.size()=\old(freelist.size()-1);

@      numfree=freelist.size();

@      (\result.getAddr()<sweepCur;||(\result.getAddr())>=sweepCur&&

@      \result.getColor==Color.BLACK;

@*/

public void allocate ( );

```

Figure7. JML specification for allocate function

The *assignable* constraint in the write barrier only changes the color of the node, while data in memory is not changed before or after the execution of write barrier. As long as there is a pointer from black to white in the stack, the white are marked gray to ensure the user program will not interfere with the execution of the garbage collector.

```

/*@ requires phase=Phase.MARK;

@ assignable stack.pointSet.color;

@ ensures (\exist Iterator it; PointerSet.iterator()>=START/(sizeof)(Pointer)

@      &&PointerSet.iterator()<=END/(sizeof)(Pointer)&&

@      \old(it.getColor==Color.BLACK)&&\old(it.next().getColor==Color.WHITE);

@      it.next().getColor==Color.BLACK&&stack.count=\old(stack.count+1)&&

@      stack.peek()==it.next());

*@/

public void djikstraStroe(Pointer field,Pointer val);

```

Figure 8. JML specification for djikstraStore().

The examples shown above illustrate that the JML specification can efficiently specify the pre- and postconditions for the mark, sweep, and write barriers of the garbage collector. Based on the system requirements, the JML specification can be applied to the entire garbage collector to improve the correctness and reliability.

V. CONCLUSION AND FUTURE WORK

We discussed the JML specification for the interaction between the garbage collector and the user program. The assertion is based on DBC pre- and postconditions in Hoare-style logic. This study focuses on real environment memory objects without abstraction, which is more reliable to some extent. The JML specification covers both normal and

abnormal behavior which can accurately describe the real-time environment. Runtime execution of the assertions is more suitable for object-oriented software. In our approach, program execution is not only the result of a function generation process, but also an assertion checking process. This approach can improve correctness and reliability for the garbage collector, quickly position errors, and handle abnormal behavior during collection. For the future, we will focus more on DBC implementation in JML, improve accuracy for describing various types of garbage collectors, and their implementation on generational concurrent garbage collectors.

REFERENCES

- [1] L. Birkedal, N. Torp-Smith, and J. Reynolds, "Local reasoning about a copying garbage collector," Proc. 31st ACM Symp On Principles of Prog .Lang. pp 220-231, 20014.
- [2] D. Russinoff, "A mechanically verified incremental garbage collector," Formal Aspects of Computing, vol. 6, pp 359-390, 1994.
- [3] L Chun-xiao, Y. Chen and Li Long, "Garbage collector verification for proof-carrying code," Journal of Computer Science and Technology vol 22, pp. 426-437, 2007.
- [4] M. Ben-Ari, "Algorithms for on-the-fly garbage collection," ACM Transactions of Principles on Programming Languages and Systems, vol. 6, pp. 333-344, 1984.
- [5] G. Leavens, A. Baker, and C. Ruby, "Preliminary Design of JML: A Behavioral Interface Specification Language for Java," ACM SIGSOFT Software Engineering Notes, vol 31, pp. 1-38, 1999.
- [6] G. Leavens, A. Baker and C. Ruby, "JML: A Notation for Detailed Design.," in Behavioral Specifications for Businesses and Systems, Chapter 12, H Kilov, B Rumpe and W Harvey, Eds. Kluwer, pp. 175-188, 1999.
- [7] G. Leavens and Y. Cheon, "Design by Contract with JML," [Online]. Available from: <http://www.eecs.ucf.edu/~leavens/JML/jmldbc.pdf>. Accessed 3/12/2014.
- [8] G. Leavens, E Poll, C. Clifton, Y. Cheon, C Ruby, D Cok, J. Kiniry, P. Chalin, D. Zimmerman and W. Dietl. "JML Reference Manual," (DRAFT), [Online]. Available from: <http://www.jmlspecs.org/OldReleases/jmlrefman.pdf>. Accessed 3/12/2014.
- [9] L. Burdy, Y. Cheon, D. Cok, M. Ernst, J. Kiniry, G. Leavens, K. Rustan M. Leino and E Poll, "An overview of JML tools and applications," International Journal on Software Tools for Technology Transfer, vol. 7, pp. 212-232, 2005.
- [10] R. Mitchell, J. McKim and B. Meyer, Design by contract, by example, Redwood City, Addison Wesley, 2002.

From Personal Computers to Personal Computing Networks

A New Paradigm for Computation

Mark Burgin
UCLA
Los Angeles, USA
mburgin@math.ucla.edu

Rao Mikkilineni and Giovanni Morana
C³DNA
Santa Clara, USA
{rao,giovanni}@c3dna.com

Abstract— Transition from mainframe computers to personal computers marked a new important step in computer technology. Here, we suggest a new transition from personal computers to personal computing networks that, as proven in scientific literature, they can be more powerful and efficient in computation. An efficient tool for personal computing networks is the Distributed Intelligent Managed Element (DIME) network architecture, which extends the conventional computational model of information processing networks, allowing improvement of the efficiency and resiliency of computational processes. This approach is based on organizing the process dynamics under the supervision of intelligent agents that, knowing the intent of the underlying process, is able to optimize its execution. In this paper, we will discuss about main ideas, structural features and tentative applications of personal computing networks and will explain why the DIME network architecture is suitable to build them.

Keywords- *personal computer; personal computing network; Oracle; DIME network architecture; structural operation; connectivity; modularity.*

I. INTRODUCTION

After their creation, electronic computers existed in the form of a mainframe computer where the end user's requests are filtered through operating staff, or a time sharing system in which one large processor is shared by many individuals. A new important step in computer technology followed with the transition from mainframe computers to personal computers intended for interactive individual use, as opposed to time sharing access to mainframe computers.

Efficiency of personal computers has grown very fast and contemporary personal computers are more powerful than gigantic mainframe computers, which existed in the past. However, people need to solve more and more complex computational problems. To do this, engineers build more and more efficient and fast computers, develop new architectures such as Grid and Cloud Computing [1] [2] [3] while programmers write more and more complex distributed software systems [4] [5] [6] [7].

Here we suggest a new computational paradigm, which presuppose the transition from personal computers to personal computing networks (PCN), a set of distributed resources provisioned on demand (often provided by different service providers) that may have global reach using private and public clouds. The goal of PCN is to solve

computational problems of the user with a specific goal with appropriate resources throughout the computation life-cycle to maintain optimal or desired availability, performance, security, compliance and cost.

As it is proved in [8], computation using networks of computers can be more powerful and efficient than computation using an individual computer. Being an efficient environment for concurrent computations, this approach will unleash a revolution of new possibilities in computer technology and applications of computers.

An efficient tool for PCNs is the distributed intelligent managed element (DIME) network architecture [9] [10] [11] [12] [13], which extends the conventional computational model of information processing networks, allowing improvement of the efficiency and resiliency of computational processes. This approach is based on organizing the process dynamics under the supervision of intelligent agents that, knowing the intent of the underlying process, is able to optimize its execution.

The DIME network architecture (DNA) utilizes the DIME computing model with non-von Neumann parallel implementation of managed Turing machines with a signaling network overlay adding cognitive elements to evolve super recursive information processing, for which it is proved that they improve efficiency and power of computational processes.

The aim of this paper is to explain why the DIME network architecture is suitable to build PCNs. The paper is organized as follows. Section II of this paper, describes the main ideas, structural features and tentative applications of PCNs. Section III describes how it is possible to build PCNs with assured distributed resources throughout the computation life-cycle based on the DIME network architecture. In Section IV, we apply theoretical models to study properties of PCNs, while in Section V, some conclusions are considered and directions for future work are suggested.

II. PERSONAL COMPUTING NETWORKS

Here we review the main ideas, structural features and tentative applications of PCNs.

There are different approaches to the PCN architecture. The goal of a PCN is to allow the user to work with a network of computers in a similar way to working with a single computer. The simplest solution is to form a

computing network and to provide access to each of its computers for the user.

However, to interact with a personal distributed network of computers at the same time is an unmanageable task. That is why to achieve this goal, it is necessary to have a special machine, which on one hand, provides an efficient interface for the user, while, on the other hand, manages functioning of all computers, called *basic network computers*, in the PCN. We call such a machine the *network Oracle* because it has to be more powerful and have more information than the basic computers in the PCN.

There are different modes how the network Oracle can manage the basic network computers. Here we consider three prime modes of such a management:

1. Information supply
2. Control
3. Supervision

Definition 1. *Information supply* of the network Oracle is the function of the Oracle in Turing machines where the computing machine from time to time goes to the Oracle to get information in the form of data that the Oracle already has.

Definition 2. When the network Oracle *A controls* functioning of the basic computers from its network, *A* monitors these computers all the time changing, if necessary, functioning of any of these computers.

Definition 3. When the group Oracle *A supervises* functioning of the basic computers from its network, *A* interacts with these computers from time to time, verifying if the functioning of any of them is correct, providing necessary data and allocating instructions for forthcoming work.

Possible applications are:

A. Information search.

Each basic network computer from a PCN uses a specific search engine for its task. For instance, one computer uses Google, another utilizes Yahoo while the third one applies Microsoft's Bing. After each search cycle, the network Oracle collects the most relevant results from all basic computers, excluding repetitions and reorganizing data. Then it transmits the results to the user and assigns new tasks to the basic network computers. In such a way, the user receives more relevant and organized information.

When it is necessary to find information about different objects, e.g., different terms, each basic network computer can explore only one object. As a result, the PCN performs search in a parallel mode decreasing time of the search.

B. Computer simulation.

Each basic network computer from a PCN uses a specific simulation algorithm or/and simulation technique for its task. After each simulation cycle, the network Oracle collects the obtained results from all basic computers, excluding repetitions, eliminating irrelevant information and reorganizing data. Then, it transmits the results to the user and assigns new tasks to the basic network computers. In

such a way, the user receives more relevant and organized information.

When it is necessary to simulate different systems, e.g., air currents and ocean currents, each basic network computer can simulate only one system. As a result, the PCN performs simulation in a parallel mode decreasing time of the simulation.

III. DIME NETWORK ARCHITECTURE AS A BASIS FOR PERSONAL COMPUTING NETWORKS

The DIME network architecture introduces three key functional constructs to enable process design, execution and management to improve both resiliency and efficiency of computing networks:

1. Machines with an Oracle
2. Blue-print or policy managed fault, configuration, accounting, performance and security monitoring and control
3. DIME network management control overlay over the managed Turing Oracle machines

A. Machines with an Oracle

Executing an algorithm, the DIME basic processor *P* performs the {read \Rightarrow compute \Rightarrow write} instruction cycle or its modified version the {interact with external agent \Rightarrow read \Rightarrow compute \Rightarrow interact with external agent \Rightarrow write} instruction cycle. This allows the external agent to influence the further evolution of computation, while the computation is still in progress. We consider three types of agents:

- (a) A DIME agent.
- (b) A human agent.
- (c) An external computing agent.

In a PCN, we use DIME with several basic processors. The DIME agent plays the role of the network Oracle in its PCN, while the basic DIME basic processors function as the basic computers from this network.

It is assumed that a DIME agent knows the goal and intent of the algorithm (along with the context, constraints, communications and control of the algorithm) the DIME basic processor is executing and has the visibility of available resources and the needs of the basic processor as it executes its tasks. In addition, the DIME agent also has the knowledge about alternate courses of action available to facilitate the evolution of the computation to achieve its goal and realize its intent. Thus, every algorithm is associated with a blueprint (analogous to a genetic specification in biology), which provides the knowledge required by the DIME agent to manage the process evolution. An external computing agent is any computing node in the network with which the DIME unit interacts.

The Distributed Intelligent Managed Element

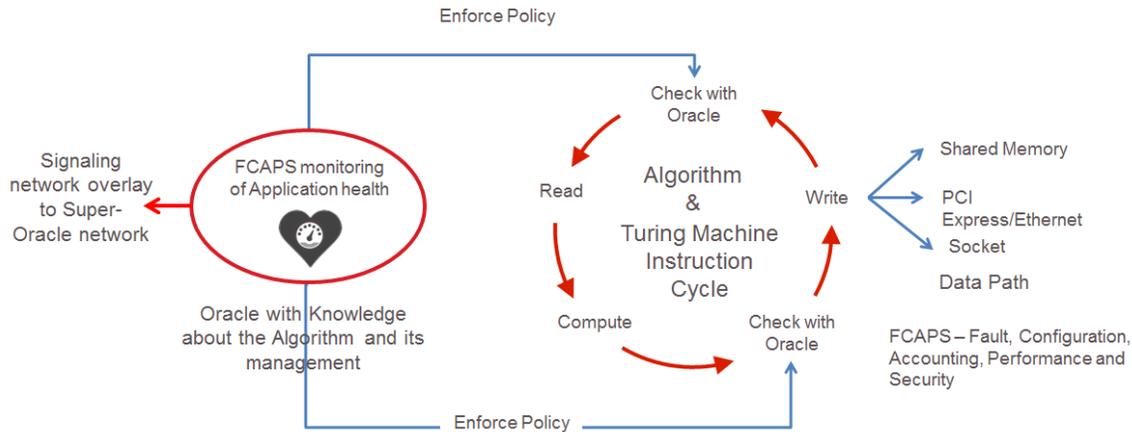


Figure 1. The distributed intelligent managed element (DIME) is a managed Turing Oracle Machine endowed with a signaling network overlay for super recursive policy based DIME network management

B. Blue-print or policy managed fault, configuration, accounting, performance and security monitoring and control

The DIME agent, which uses the blueprint to configure, instantiate, and manage the DIME basic processor executing the algorithm uses concurrent DIME basic processors with their own blueprints specifying their evolution to monitor the vital signs of the DIME basic processor.

It also implements various policies to assure non-functional requirements such as availability, performance, security and cost management while the managed DIME basic processor is executing its intent. Figure 1 shows the DIME basic processor (executing its “Algorithm & Turing Machine Instruction Cycle”) and its DIME agent (performing “FCAPS monitoring of Application health”). The DIME agent extends the basic processor capability adding the “Check with Oracle” operation: this allows the DIME agent to infuse to the underlying basic processor new knowledge, stored as the blueprint [15] and coming from both local information, collected by the DIME agent itself, and global information, received from the network of oracles through the signaling channel.

C. DIME network management control overlay over the managed Turing Oracle machines

In addition to read/write communication of the DIME basic processor (the data channel), other DIME basic processors communicate with each other using a parallel signaling channel. This allows the external DIME agents to influence the computation of any managed DIME basic processor in progress based on the context and constraints. The external DIME agents are DIMEs themselves. As a result, changes in one computing element could influence the evolution of another computing element at run time without halting its Turing machine executing the algorithm. The

signaling channel and the network of DIME agents can be programmed to execute a process, the intent of which can be specified in a blueprint. Each DIME basic processor can have its own Oracle managing its intent, and groups of managed DIME basic processors can have their own domain managers implementing the domain’s intent to execute a process. The management DIME agents specify, configure, and manage the sub-network of DIME units by monitoring and executing policies to optimize the resources while delivering the intent.

Figure 2 shows the DIME network architecture implementation for a process with different hardware, functions and an evolving structure used to attaining the intent of the process. This architecture has following benefits from current architectures deploying virtual machines to provide cloud services such as self-provisioning, self-repair, auto-scaling and live-migration:

1. Using DNA same cloud services can be provided at application and workflow group level across physical or virtual servers. The mobility of applications comes from utilization of the policies implemented to manage the intent through the signaling network overlay over the managed computing network. Applications are moved into static Virtual Machines or physical servers with given service levels provisioned.

2. Scheduling, monitoring, and managing distributed components and groups with policies at various levels decouple the application/workflow management from underlying distributed infrastructure management systems. The vital signs (CPU, memory, bandwidth, latency, storage IOPs, throughput and capacity) are monitored and managed by DIMEs, which are functioning similar to the Turing Oracle Machines.

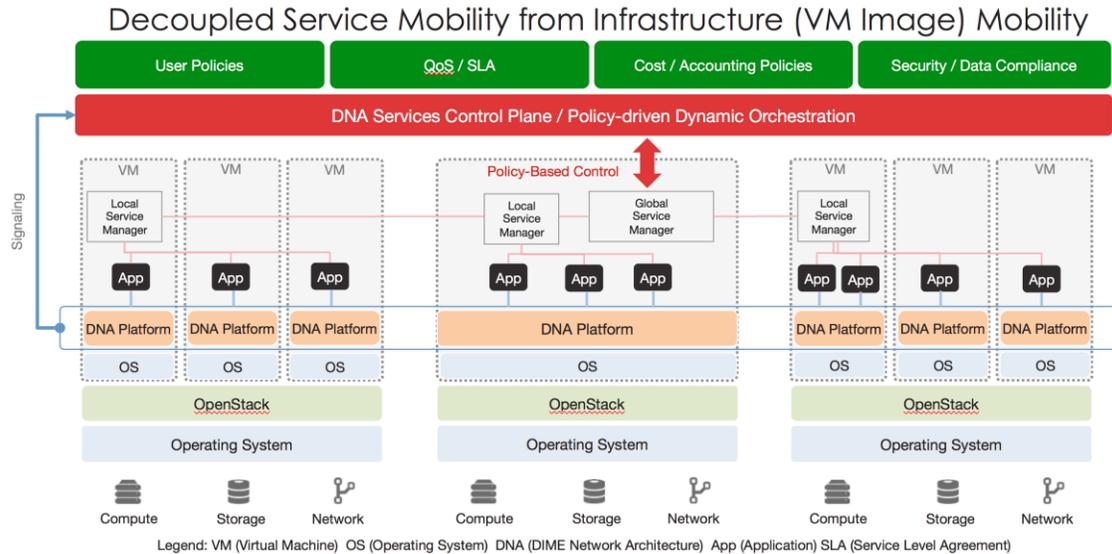


Figure 2. A managed application network with super recursive global, local and process level policy management with a signaling control network overlay

While implementing the monitoring and management of the DIME agent, the DIME network monitors and manages its own vital signs and executing various policies to assure availability, performance and security. At each level in the hierarchy, a domain specific task or workflow is executed to implement a distributed process with a specific intent. In Figure 2, each web component has its own policies and the group has the service level policies that define its availability, performance and security. Based on policies, the elements are replicated or reconfigured to meet the resource requirements based on monitored behavior.

In essence, the DIME computing model infuses sensors and actuators connecting the DIME basic processor with the DIME agent to manage the DIME basic processor and its resources based on the intent, interactions and available resources. Higher level policy managers are used to configure, monitor and manage the intent of a network of lower level managed basic processors.

The DIME network architecture has been successfully implemented using both Linux and Parallax [9][10][14], an operating system designed to support natively the DIME-based approach. More recently, a product based on DIME network architecture was used to implement auto-failover, auto-scaling, and live-migration of a web based application deployed on distributed servers with or without virtualization [15].

The mobility of the applications is provided by using the Oracle interrupt to control the down-stream processes. The global knowledge of down-stream process activity allows the higher level Oracles to reason and affect changes to the down-stream process dynamics. In addition to read/write communication of the basic automaton (the data channel), the Oracles manage different basic automata communicating with each other using a parallel signaling channel. This allows the external Oracles to influence the computation of any managed basic automaton in progress based on the

context and constraints just as a Turing Oracle is expected to do.

The Oracle uses the blueprint to configure, instantiate, and manage the automaton and executing the algorithm. Utilization of concurrent automata in the network with their own blueprints specifying their evolution to monitor the vital signs of the DIME basic automaton and to implement various policies allows the Oracle to assure non-functional requirements such as availability, performance, security and cost management, while the managed DIME basic automaton is executing its task to achieve its goal and realize its intent.

The external Oracles represent DIME agents, allowing changes in one computing element influence the evolution of another computing element at run time without stopping its basic automaton executing the algorithm. The signaling channel and the network of the Oracles can be programmed to execute a process whose intent itself can be specified in a blueprint. Each basic automaton can have its own Oracle managing its intent, and groups of managed basic automata can have their own domain managers implementing the domain's intent to execute a process. The management Oracles specify, configure and manage the sub-network of DIMEs by monitoring and executing policies to optimize the resources while delivering the intent. The DIME network implementing the Oracles is itself managed by monitoring its own vital signs and executing various FCAPS (i.e. Fault, Control, Accounting, Performance and Security) policies to assure availability, performance and security.

An Oracle is modeled by an abstract automaton that has higher computational power and/or lower computational complexity than the basic automaton it manages. For instance, the Oracle can be an inductive Turing machine, while the basic automaton is a conventional Turing machine.

End-to-end Service visibility

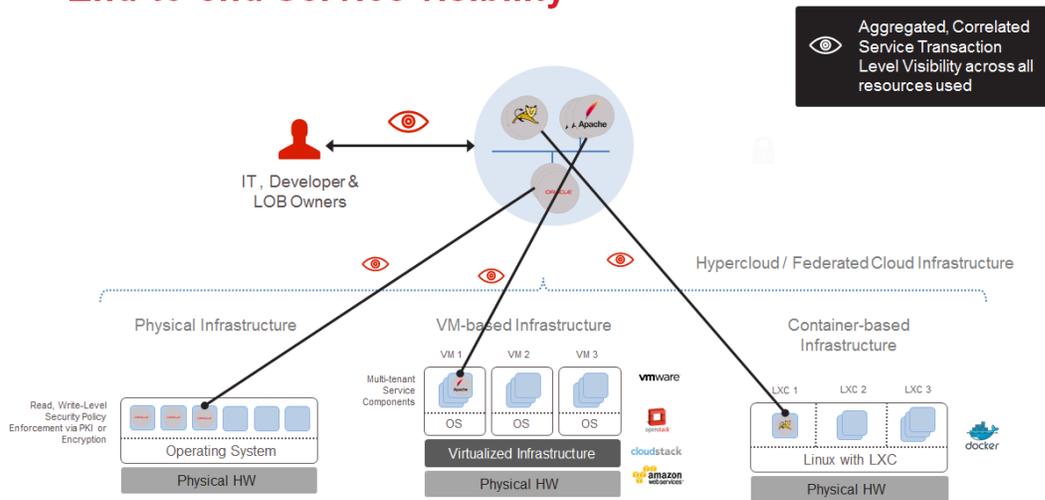


Figure 3. A Web application suite deployed using DIME network architecture with policies to monitor and manage the service availability, performance and security using auto-scaling, auto-fail-over and live migration

It is proved that inductive Turing machines have much higher computational power and lower complexity than conventional Turing machine [16][17][18][19].

DIME agents possess a possibility to infer new data and knowledge from the given information. Inference is one of the driving principles of the Semantic Web, because it will allow us to create software applications quite easily. For the Semantic Web applications, DIME agents need high expressive power to help users in a wide range of situations.

To achieve this, they employ powerful logical tools for making inferences. Inference abilities of DIME agents are developed based on mathematical models of these agents in the form of inductive Turing machines, limit Turing machines [16] and evolutionary Turing machines [18][20][21][22].

Figure 3 shows a DNA workflow of a web application running on a physical infrastructure that has policies to manage auto-failover by moving the components when the vital signs being monitored at various levels are affected. For example if the virtual machine in the middle server fails, the service manager at higher level detects it and replicates the components in another server on the right and synchronizes the states of the components based on consistency policies. A similar schema, described in details in another work [15], is adopted for handling (live) migration and (auto) scaling-out.

Figure 4 shows, in details, how a network of Oracles can be deployed in order to allow the user/developer of the application to have full control over an application. A user (a developer of a web service in this case) can provision resources by defining the intent of the global Oracle and use

multiple Oracles at various levels to implement, monitor and manage the life-cycle of each component of the web application. The DIME network architecture allows components to be developed and composed into services, deploy them, monitor the resources and their behavior and take corrective actions to fulfill the intent.

In conclusion, the PCNs distinguish themselves with the following properties:

1. On-demand service provisioning with required resources,
2. Auto-failover based on policies,
3. Stateful or stateless application migration and
4. End-to-end service visibility and control to assure availability, performance and security.

Users can create application components and compose them into service workflows and execute them on available distributed resources with dynamic service assurance.

IV. CAPABILITIES OF PRIVATE COMPUTING NETWORKS

Let us assume that it is possible to model the group Oracle and each computer from its group by a Turing machine. Results from [16] allow us to prove the following result.

Theorem 1. If the network Oracle A works in the recursive mode and controls functioning of basic network computers in the recursive mode, then it is possible to model the network functioning by a Turing machine.

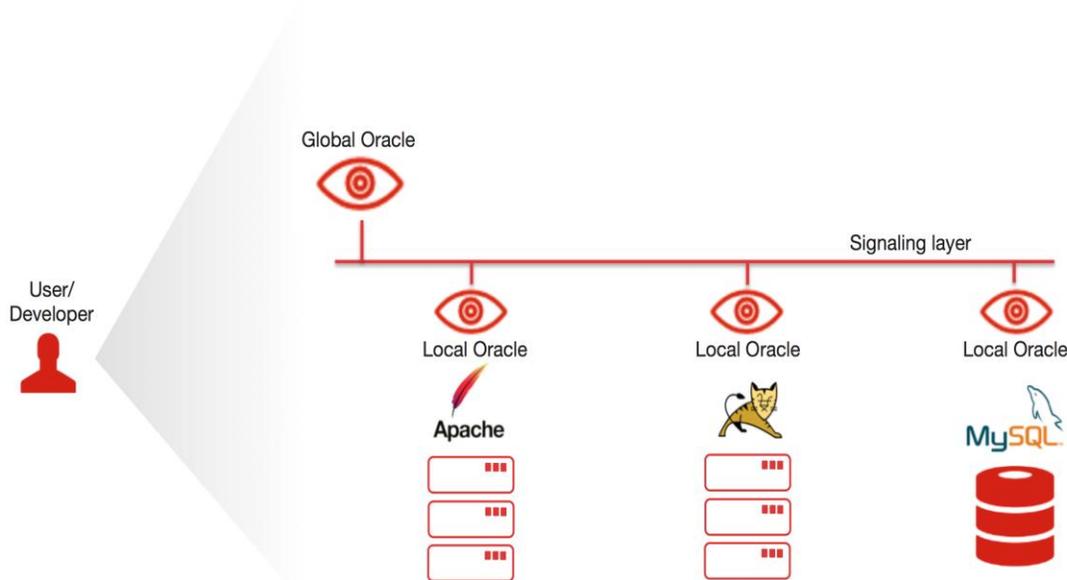


Figure 4. PCN for a Web Application

Results from [16] allow us to prove the following results.

Theorem 2. If the network Oracle *A* working in the recursive mode supervises functioning of basic network computers, then the supervised network can be more powerful than any Turing machine even if all basic network computers function in the recursive mode and *A* does not have noncomputable information.

Theorem 3. If the network Oracle *A* has recursively non-computable information, then the supervised network can be more powerful than any Turing machine.

V. CONCLUSION

Three innovations are introduced, namely, the parallel monitoring of vital signs (CPU, memory, bandwidth, latency, storage IOPs, throughput and capacity) in the DIME, signaling network overlay to provide run-time service management and machines with Oracles in the form of

DIME agents. This allows interruption for policy management at read/write in a file/device allow self-repair, auto-scaling, live-migration and end-to-end service transaction security with private key mechanism independent of infrastructure management systems controlling the resources and thus, provide freedom from infrastructure and architecture lock-in. The DIME network architecture puts the safety and survival of applications and groups of applications delivering a service transaction first using secure mobility across physical or virtual servers. It provides information for sectionalizing, isolating, diagnosing and fixing the infrastructure at leisure. The DIME network architecture therefore makes possible reliable services to be delivered on even not-so-reliable infrastructure. Modeling this

architecture by grid automata allows researchers to study properties and critical parameters of semantic networks and provides means for optimizing these parameters. Future work will investigate specific predictions that can be made from the theory for a specific DIME network execution and compare the resiliency and efficiency using both recursive and super-recursive implementations.

ACKNOWLEDGMENT

Rao Mikkilineni and Giovanni Morana thank the team from C³ DNA Inc., for their continued support in implementing the DIME network architecture.

REFERENCES

- [1] R. Buyya, C. S. Yea, S. Venugopala, J. Broberga and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, Vol. 25, no. 6, pp. 599-616, 2009.
- [2] J. Varia, *Cloud Computing: Principles and Paradigms*, Wiley Press, 2011, Best Practices in Architecture.
- [3] M. Rahman, R. Ranjan, R. Buyya and B. Benatallah, *A Taxonomy and Survey on Autonomic Management of Applications in Grid Computing Environments, Concurrency and Computation: Practice and Experience*, Volume 23, Number 16, Pages: 1990-2019, ISSN: 1532-0626, Wiley Press, New York, USA, November 2011.
- [4] S. Jha, et al. "Distributed computing practice for large - scale science and engineering applications." *Concurrency and Computation: Practice and Experience* 25.11 (2013): 1559-1585.
- [5] D. Thain, T. Tannenbaum, and M. Livny. "Distributed computing in practice: The Condor experience." *Concurrency and computation: practice and experience* 17.2 - 4 (2005): 323-356.

- [6] A.D. Kshemkalyani, M. Singhal, *Distributed Computing: Principles, Algorithms, and Systems*, ISBN-13: 9780521876346, Cambridge University Press (2008).
- [7] O. Babaoglu, K. Marzullo, *Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms*, *Distributed Systems*, ACM Press (editor S.J. Mullender), Chapter 4, 1993.
- [8] M. Burgin, *Algorithmic Control in Concurrent Computations*, in *Proceedings of the 2006 International Conference on Foundations of Computer Science*, CSREA Press, Las Vegas, June, 2006, pp. 17-23 .
- [9] R. Mikkilineni, *Designing a New Class of Distributed Systems*. New York: Springer, 2011.
- [10] R. Mikkilineni, G. Morana and I. Seyler, "Implementing Distributed, Self-Managing Computing Services Infrastructure using a Scalable, Parallel and Network-Centric Computing Model." In *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*, ed. M. Villari, I. Brandic and F. Tusa, pp. 57-78 2012.
- [11] R. Mikkilineni, "Architectural Resiliency in Distributed Computing," *International Journal of Grid and High Performance Computing (IJGHPC)* 4. 2012 doi:10.4018/jghpc.2012100103. [retrieved: February, 2015]
- [12] R. Mikkilineni, G. Morana, D. Zito, and M. Di Sano, "Service Virtualization Using a Non-von Neumann Parallel, Distributed, and Scalable Computing Model," *Journal of Computer Networks and Communications*, vol. 2012, Article ID 604018, 10 pages, 2012. doi:10.1155/2012/604018 [retrieved: February, 2015].
- [13] R. Mikkilineni, "Going beyond Computation and Its Limits: Injecting Cognition into Computing." *Applied Mathematics* 3 (2012): 1826.
- [14] R. Mikkilineni, A. Comparini and G. Morana, *The Turing O-Machine and the DIME Network Architecture: Injecting the Architectural Resiliency into Distributed Computing*, In *Turing-100. The Alan Turing Centenary*, (Ed.) Andrei Voronkov, *EasyChair Proceedings in Computing*, Volume 10, 2012.
- [15] R. Mikkilineni and G. Morana, "Infusing Cognition into Distributed Computing: A new approach to distributed datacenters with self-managing services" *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2014 23rd IEEE International Conference, June 2014, pp.131-136.
- [16] M. Burgin, *Super-recursive Algorithms*, New York: Springer, 2005.
- [17] M. Burgin, *Interactive Hypercomputation*, in *Proceedings of the 2007 International Conference on Foundations of Computer Science (FCS'07)*, CSREA Press, Las Vegas, Nevada, USA, 2007, pp. 328-333.
- [18] M. Burgin, "Reflexive Calculi and Logic of Expert Systems", in *Creative processes modeling by means of knowledge bases*, Sofia, (1992) pp. 139-160 .
- [19] J. P. Crutchfield and M. Mitchell, "Evolution of Emergent Computation" *Computer Science Faculty Publications and Presentations*. Paper 3. 1995
http://pdxscholar.library.pdx.edu/compsci_fac/3 [retrieved: February, 2015].
- [20] M. Burgin, "Inductive Turing machines," *Notices of the Academy of Sciences of the USSR*, 270 N6 (1983) pp. 1289-1293 (translated from Russian).
- [21] M. Burgin and E. Eberbach, "On Foundations of Evolutionary Computation: An Evolutionary Automata Approach," in *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies* (Hongwei Mo, Ed.), IGI Global, Hershey, Pennsylvania, 2009, pp. 342-360 .
- [22] M. Burgin and E. Eberbach, "Evolutionary Automata: Expressiveness and Convergence of Evolutionary Computation," *Computer Journal*, v. 55, No. 9 (2012) pp. 1023-1029.

Assessment of MVAR Injection in Power Optimization for a Hydrocarbon Industrial Plant Using a Genetic Algorithm

M. T. Al-Hajri
Computer & Electronic Eng. Dept.
Brunel University
Uxbridge, United Kingdom
e-mail: muhammad.al-hajri@brunel.ac.uk

M. A. Abido
Electrical Engineering Department.
King Fahad University (KFUPM)
Dhahran, Kingdom of Saudi Arabia
e-mail: mabido@kfupm.edu.sa

M. K. Darwish
Computer & Electronic Eng. Dept.
Brunel University, U.K.
Uxbridge, United Kingdom
e-mail: mohamed.darwish@brunel.ac.uk

Abstract- In this paper, a genetic algorithm (GA) is considered for assessing the effect of Million Volt-Ampere Reactive (MVAR) power injection via shunt capacitors in optimizing the electrical power loss for a real hydrocarbon industrial plant. The subject plant electrical system consists of 275 buses, two gas turbine generators, two steam turbine generators, large synchronous motors, and other rotational and static loads. The minimization of power losses objective is used to guide the optimization process, and, consequently, the injected power in the grid is monitored. First, the optimal locations of MVAR injection will be identified using a voltage stability index. The potential of power loss optimization with and without MVAR injection versus the base case will be discussed in the results. The results obtained demonstrate the potential and effectiveness of the proposed approach to optimize the power consumption in both scenarios (with and without MVAR injection). Also, in this paper a cost appraisal for the potential daily, monthly and annual cost saving in both scenarios will be addressed.

Keywords- genetic algorithm, power loss optimization, electrical submersible pump, hydrocarbon facility, British thermal unit (BTU), millions of standard cubical feet of gas (MMscf).

I. INTRODUCTION

Since 2009, an environment of urgency was created to deal with the exponential increase of the domestic energy used in the kingdom of Saudi Arabia. All stockholders since then are working together in many initiatives sponsored by the government of the kingdom to address the optimization and the efficiency improvement of energy in all utility sectors, especially, the electrical generation sector. The kingdom international commitment to reduce the CO₂ emission was another driver for energy optimization in the kingdom. Optimizing the oil usage for electrical generation is also for the benefits of the oil producing countries. The optimization of the energy sector will support the development of downstream petrochemical industries and other very promising industries. In addition, the need of shaping the high annual rate increase of energy demand becomes a major concern for most of the developing countries. For example, in Saudi Arabia, the annual electric demand increase is around 8% [1]. All these pressing critical issues push many countries to develop nationwide strategies

for enhancing the electricity generation efficiency, reduce loss and invest in the renewable energy development.

In light of the aforementioned challenges and others, GA was addressed in literature for optimizing the electrical power

system loss. Optimizing the power loss of virtual IEEE system models, improving the performance of the GA by adapting different crossover and mutation techniques and creating a hybrid GA by combining it with other techniques, such as the swarm particles and Fuzzy logic, were among the many techniques addressed in literature. None of the previous studies addressed the application of GA in optimizing the power loss of real hydrocarbon facility with small system footprint, shorter lines, large machines, combined cycle's generation and large load. [2]-[9].

Generally, there are three approaches to solve the real power loss optimization problem by optimizing the reactive power flow. The first approach applies sensitivity analysis and gradient-based optimization algorithms by linearizing the objective function and the system constrains around operating points [10]. The gradient-based methods are usually subjected to be trapped in local minima which makes the obtained solution not optimal. Moreover, sensitivity factors calculation is a time consuming process. The second approach uses a nonlinear programming technique [11]. This approach has many disadvantages such as long execution time, insecure convergence properties and algorithmic complexity. The third approach utilizes heuristic methods to search the solution space for the optimal solution. This approach is promising as it can overcome the possibility of trapping in local minima [9].

This paper considers an existing real life hydrocarbon central processing facility electrical power system model for assessing the potential of system loss minimization using the GA for two scenarios: without and with MVAR injection. In section 2 of the paper the problem will be formulated as optimization problem with equality and inequality constrains. Also, the voltage stability index will be introduced. In section 3, the GA will be employed to solve this problem. In section 4, the paper study scenarios will be developed. Finally, in section 5 the results technically and economically will be evaluated.

II. PROBLEM FORMULATION

The problem formulation consists of three parts: the development of the objective functions, the identification of the system electrical constrains to be met; equality and inequality constrains; and the calculation of all load buses stability index (L-Index).

A. Problem Objective Functions

The objective function is to minimize the real power loss J_1 (P_{Loss}) in the transmission and distribution lines. This objective function can be expressed in term of the power follow loss between two buses i and j as follows:

$$J_1 = P_{Loss} = \sum_{k=1}^{nl} g_k [V_i^2 + V_j^2 - 2 V_i V_j \cos(\delta_i - \delta_j)] \quad (1)$$

Where nl is the number of transmission and distribution lines; g_k is the conductance of the k^{th} line, $V_i \angle \delta_i$ and $V_j \angle \delta_j$ are the voltage at end buses i and j of the k^{th} line, respectively [12] [13].

The real power injected (PR_{Inject}) in the utility grid at Bus# 1 was monitored as J_1 evolves. It is expected that PR_{Inject} will be maximized since it is inversely proportional to J_1 ; a decrease in the J_1 results in an increase in PR_{Inject} .

B. Problem Equality and Inequality Constrains

The system constrains are divided into two categories: equality constrains and inequality constrains [9][13]. Details are as follows:

B.1 Equality Constrains

These constrains represent the power load flow equations. The balance between the active power injected P_{Gi} , the active power demand P_{Di} and the active power loss P_{li} at any bus i is equal to zero. The same balance apply for the reactive power Q_{Gi} , Q_{Di} , and Q_{li} . These balances are presented as follows:

$$P_{Gi} - P_{Di} - P_{li} = 0 \quad (2)$$

$$Q_{Gi} - Q_{Di} - Q_{li} = 0 \quad (3)$$

The above equations can be detailed as follow:

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)] = 0 \quad (4)$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)] = 0 \quad (5)$$

where $i = 1, 2, \dots, NB$; NB is the number of buses; P_G and Q_G are the generator real and reactive power, respectively; P_D and Q_D are the load real and reactive power, respectively; G_{ij} and B_{ij} are the conductance and susceptance between bus i and bus j , respectively.

B.2 Inequality Constrains

These constrains represent the system operating constrains posted in Table III and they are as follow:

- Generator and synchronous motor voltages; V_G and V_{Synch} ; their reactive power outputs; Q_G and Q_{Synch} .
- The transformers taps.
- The load buses voltages V_L .

Combining the objective function and these constrains, the problem can be mathematically formulated as a nonlinear constrained single objective optimization problem as follows:

Minimize J_1

Subject to:

$$g(x,u) = 0 \quad (6)$$

$$|h(x,u)| \leq 0 \quad (7)$$

where:

x : is the vector of dependent variables consisting of load bus voltage V_L , generator reactive power outputs Q_G and the Synchronous motors reactive Power Q_{Synch} . As a result, x can be expressed as

$$x^T = [V_{L1} \dots V_{LN}, Q_{G1} \dots Q_{GN}, Q_{Synch1} \dots Q_{SynchNSynch}] \quad (6)$$

u : is the vector of control variables consisting of generator voltages V_G , transformer tap settings T , and synchronous motors voltage V_{Synch} . As a result, u can be expressed as

$$u^T = [V_{G1} \dots V_{GN}, T_1 \dots T_{NT}, V_{Synch1} \dots V_{SynchNL}] \quad (8)$$

g : are the equality constrains.

h : are the inequality constrains.

C. Voltage Stability Index (L-Index)

The L indicator varies in the range between 0 (the no load case) and 1, which corresponds to voltage collapse. This indicator uses the bus voltage and network information provided by the power flow program to measure the stability of the system. The L indicator can be calculated as given in [14]. For a multi-node system

$$I_{bus} = Y_{bus} \times V_{bus} \quad (9)$$

By segregating the load buses (PQ) from generator buses (PV), (8) can be written as:

$$\begin{bmatrix} I_L \\ I_G \end{bmatrix} = \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} \begin{bmatrix} V_L \\ V_G \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} V_L \\ I_G \end{bmatrix} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{bmatrix} I_L \\ V_G \end{bmatrix} = \begin{bmatrix} Z_{LL} & F_{LG} \\ K_{GL} & Y_{GG} \end{bmatrix} \begin{bmatrix} I_L \\ V_G \end{bmatrix} \quad (11)$$

where

V_L, I_L are load buses voltages and currents

V_G, I_G are Generator buses voltages and currents

H_1, H_2, H_3, H_4 are submatrices generated from Y_{bus} Partial Inversion

$Z_{LL}, F_{LG}, K_{GL}, Y_{GG}$ are submatrices of H-matrix

Therefore, a local indicator L_j can be worked out for each node j similar to the line model

$$L_j = \left| \mathbf{1} - \frac{\sum_{i \in \alpha_L} F_{ji} V_i}{V_j} \right| \quad (12)$$

For a stable situation the condition $L_j \leq 1$ must not be violated for any of the nodes j . Hence, a global indicator L describing the stability of the whole system is given by:

$$L_{max} = \text{MAX}_{j \in \alpha_L} \left| \mathbf{1} - \frac{\sum_{i \in \alpha_L} F_{ji} V_i}{V_j} \right| \quad (13)$$

where α_L is the set of load buses and α_G is the set of generator buses.

III. THE PROPOSED APPROACH

A. Electrical System Model Data Collection

The research electrical models system parameters were gathered and categorized in tables to be ready for developing the simulation model of the system. The gathered parameters include the followings:

- Generators type, voltage and capacity, including active and reactive capacity curves reflecting the operation limitations such as stator and rotor thermal limitations.
- The Generators BTU/kW equation and cost equation.
- Utility power system parameters (swing bus); bus voltage and short circuit MVA.
- System buses voltage constrains.
- Lines parameters, including the lines resistance, reactance, capacitance, length and voltage.
- Transformers parameters including primary voltage, secondary voltage, voltage taps, size and impedance.
- The large synchronous motor parameters, including active and reactive power curve reflecting the operation limitations such as stator and rotor thermal limitations.
- The large induction motor and the electrical submersible pumps (ESPs) parameters such as the active and reactive power demands.

- The lumped load Thousand Voltage-Ampere (KVA) rating. All loads except the motor rated > 5000 Horse Power (HP) and the ESP are modeled as lumped load.

B. Optimal Locations of the Shunt Capacitors

The L index described in (13) was employed to identify the most sensitive load buses with regard to voltage stability. These most sensitive load buses were selected for shunt capacitors connection.

C. Generic Algorithm Implementation

The implementation of the developed GA technique can be summarized in the following steps:

- Generate initial populations of chromosomes; each chromosome consists of genes and each of these genes represents either transformer tap settings, synchronous motors voltages, the generators voltages or shunt capacitors MVAR values.
- Assign fitness to each chromosomes as follows;
 - Use the Newton-Raphson method to calculate the real power losses for each population [15].
 - Identify if the voltage constrains are satisfied.
 - Identify if the Synchronous machines (generators and motors) capacity limitations are met.
 - Assign fitness values to the populations that meet the voltage constrains; the population best power loss value (J_1) divided by the base case power loss value.
 - Assign penalty values to those populations that do not meet the voltage constrains; constant value (0.05).
- Identify the best population with its associated chromosomes that has the best objective function value and store it.
- Identify the chromosomes parents that will go to the mating pool for producing the next generation via the Random Selection method. This method works by generating two random integer numbers (each represents a chromosome). Then, these two randomly selected chromosomes fitness values are compared and the one with the better fitness value will go into the mating pool. This randomly selected chromosomes mechanism will be repeated until the population in the mating pool equals to the initial chromosomes population [16].
- Perform genes crossover for the mating pool parents via the Simple Crossover method [16]. In this method, the offspring chromosomes are generated by establishing a vertical crossover position for parent's chromosomes and then crossover their genes.
- Perform gene mutation for the mating pool parents after they have been crossed over; the Random Mutation method was implemented [16]. In this method, the offspring chromosomes genes are mutated to new ones randomly from the genes domain.

- 7) Go to Step #2 and repeat the above steps with the new populations generated from the original chromosome parents after being crossed over and mutated.
- 8) Each time, identify the best population and compare its fitness value with the stored one; if it is better (meeting the objective function), replace the best chromosomes with the new ones.
- 9) The loop of generation is repeated until the best population with its associated chromosomes, in terms of minimum real power loss, is identified or the maximum number of generations is met. The flow chart of the proposed approach implemented is shown in Figure 1.

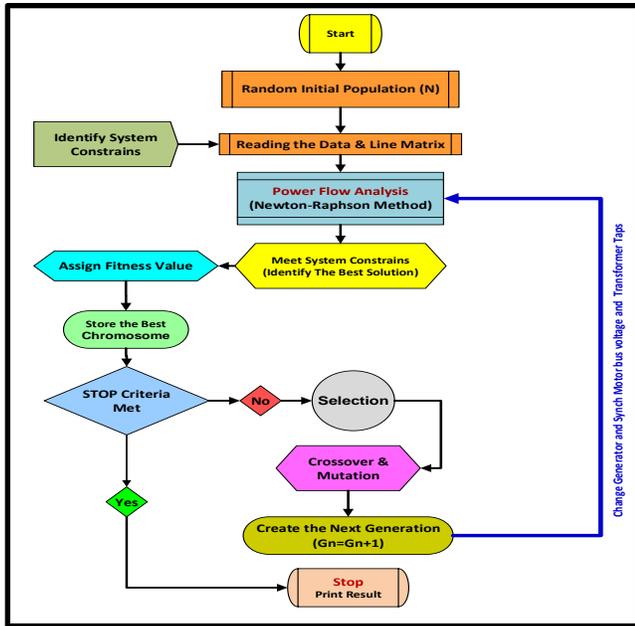


Figure 1. The GA algorithm evolution process flowchart

IV. STUDY SCENARIOS

In this paper, three scenarios were studied: the base case scenario (system as usual), the optimal case scenario without MVAR injection, and the optimal case scenario with MVAR injection. In the optimal cases the best system parameters (chromosomes) that meet the minimum objective function (J_1) are obtained.

A. Base Case Scenario

Normal system operation mode was simulated to be benchmarked with the two optimal scenarios. Following are some of the normal system operation mode parameters:

- 1) The utility bus and generators terminal buses were set at unity p.u. voltage.
- 2) All the synchronous motors were set to operate very close to the unity power factor.
- 3) All downstream distribution transformers and the captive synchronous motors transformers; off-load tap changers; were put on the neutral tap.

- 4) The causeway substations main transformers taps were raised to meet the very conservative voltage constrains at these substations downstream buses; ≥ 0.95 p.u. Refer to Table I below.

TABLE I
THE SELECTED FEASIBLE TRANSFORMERS TAPS VALUE

Substation Number	Transformer Tap
Causeway Substation#1	+3 (1.019 p.u.)
Causeway Substation#2	Neutral (1.0 p.u.)
Causeway Substation#3	+3 (1.019 p.u.)
Main Substation Transformers	+1 (1.006 p.u.)

B. Optimal Case Scenario without MVAR injection

To optimize the elevation process time the unfeasible transformers tap values (genes) were not selected. In other words, the genes values were limited to certain taps around the neutral taps out of the all taps full range; ± 16 taps. Table II below posts the selected range of the transformers tap values and the percentage of the voltage change for each tap.

TABLE II
THE SELECTED TRANSFORMER TAP FEASIBLE GENES VALUE

Description	Upper Tap	Lower Tap
Main Transformers	+8 (0.625%)	-4 (0.625%)
Causeway Main Transformers	+8 (0.625%)	-3 (0.625%)
Captive Motors/Distribution Transformers	+1 (2.5%)	-1 (2.5%)
Generator Step-up Transformers	+5 (1.25%)	-4 (1.25%)

An initial 200 populations of feasible chromosomes (individuals) which meet both the buses voltage and synchronous machine reactive power constrains were identified. These feasible populations are associated with the first optimal scenario; without MVAR injection. The feasible populations with their associated chromosomes were subject to the GA evolutionary process of 20 generations guided by the objective function J_1 . The PR_{inject} was monitored as J_1 evolved. The GA process was set with 90% crossover probability and 10% mutation probability. The system parameters and the objective function value associated with the optimal solution of this scenario were identified.

C. Optimal Case Scenario with MVAR injection

The evolutionary process was optimized via the same method employed in the second case scenario. Another initial 300 populations of feasible individuals were identified in for the second optimal scenario; with MVAR injection. In this scenario, MVAR shunt capacitors are connected to the preselected buses; refer to Table IV. In this case, the MVAR chromosomes are extended to include MVAR injection considered as control variables. The feasible populations with their associated chromosomes were subject to 20 generation of GA evolutionary process. The crossover and

mutation probability were set equal to those in the scenario without MVAR injection: 90% and 10%, respectively.

V. RESULTS AND DISCUSSIONS

The results from the three scenarios, base case, without MVAR injection and with MVAR injection, will be analyzed in two categories: the system parameters analysis and the economic analysis.

A. System Parameters Analysis

The hydrocarbon facility simplified electrical system model, which is studied in this paper, is shown in Figure 2.

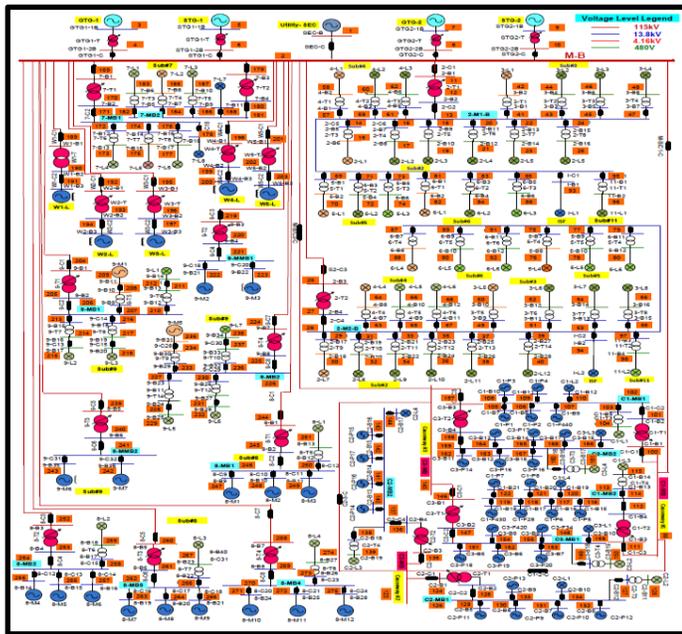


Figure 2. Simplified electrical system of the hydrocarbon processing facility

The system inequality constrains are posted in Table III.

TABLE III
SYSTEM INEQUALITY CONSTRAINS

Description	Lower Limit	Upper Limit
GTG Terminal Voltage (V_{GTG})	90%	105%
STG Terminal Voltage (V_{STG})	90%	105%
GTG Reactive Power (Q_{GTG}) Limit	-62.123 MVAR	95.72 MVAR
STG-1 Reactive Power (Q_{STG}) Limit	-22.4 MVAR	20.92 MVAR
STG-2 Reactive Power (Q_{STG}) Limit	-41.9 MVAR	53.837 MVAR
Captive Synch. Motors Terminal Voltage	90%	105%
Synch. Motors Terminal Voltage (V_{Synch})	90%	105%
Causeway downstream Buses Voltage	95%	105%
All Load Buses Voltage	90%	105%
Main Transformer Taps	+16 (+10%)	-16 (-10%)
Generators Step-Up Transformer Taps	+8 (+10%)	-8 (-10%)

Base on the substation load buses stability index rank, the selected buses for shunt capacitors connection, together with the potential MVAR values, are posted in Table IV.

TABLE IV
THE SELECTED BUSES FOR MVAR INJECTION

Substation Number	Bus Number	Potential MVAR
Substation#2	13 and 28	[8 8.5 9 9.5 10]
Causeway Substation#1	102 and 113	[1.5 2 2.5 3 3.5]
Causeway Substation#2	124 and 137	[2 2.5 3]
Causeway Substation#3	148 and 159	[2 2.5 3]

The evolution of the objective function (J_1) and PR_{Inject} values over the GA process is captured in Figure 3. The benchmark for the system real power loss and the injected power in the grid is demonstrated in Figure 4. There are 0.202 Million Watts (MW) and 0.203 MW reduction in the system loss between the base case, the no MVAR and with MVAR optimal cases sequentially. The same amount of MW were injected in the grid for both scenarios.

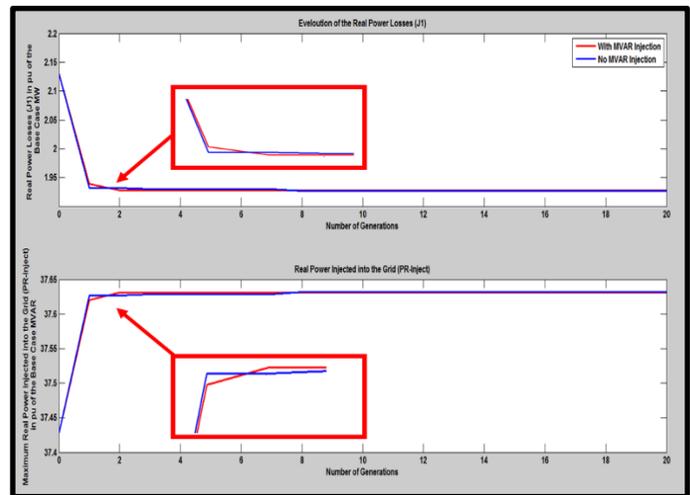


Figure 3. J_1 and PR_{Inject} value convergent for 10 generations

The system for the two optimal cases demonstrates an improvement in the system buses p.u. voltage profile, which increases the robustness of the system (Figure 5).

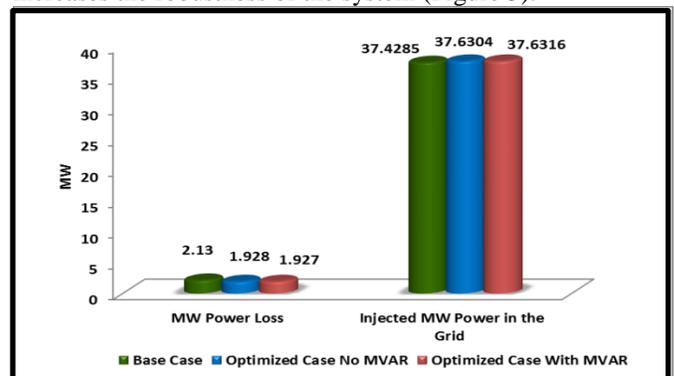


Figure 4. System power loss and injected power benchmark

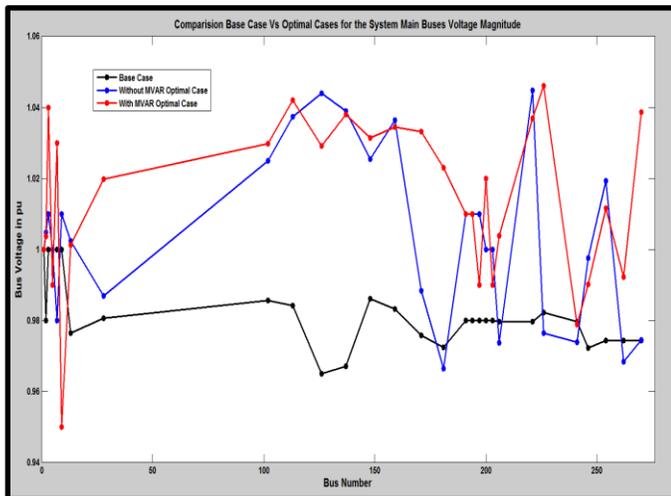


Figure 5. The main system buses' voltage benchmark

The paper shows there is not much power loss reduction associated with MVAR injection scenario compared to the no MVAR injection scenario [17] due to the system stiffness and its small footprint.

B. Economic Analysis

The avoided cost due to the optimization of the system power loss is demonstrated in Figure 6 at daily, monthly and annual bases. The annual cost avoidance based on natural gas cost of \$3.5 per MMscf is around \$60,300/year and \$60,400/year for the no MVAR and with MVAR optimal cases sequentially when compared to the base case.

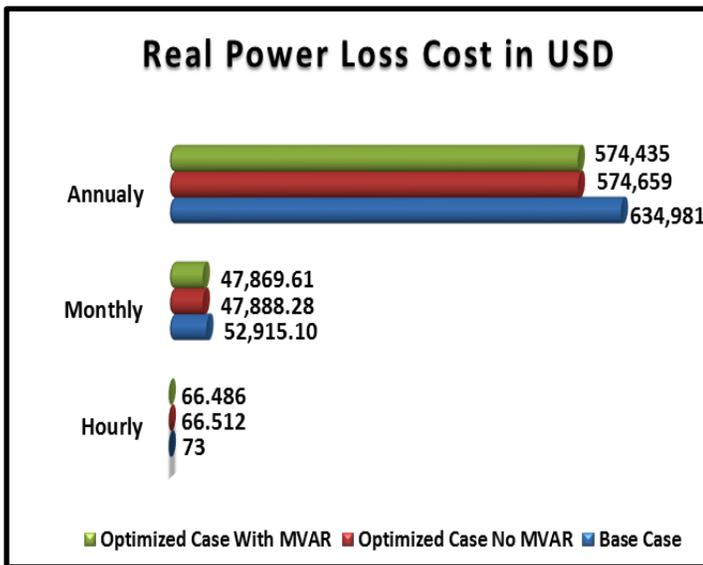


Figure 6. The system power loss cost

The revenue due to the power injection in the grid at both scenarios is shown in Figure 7.

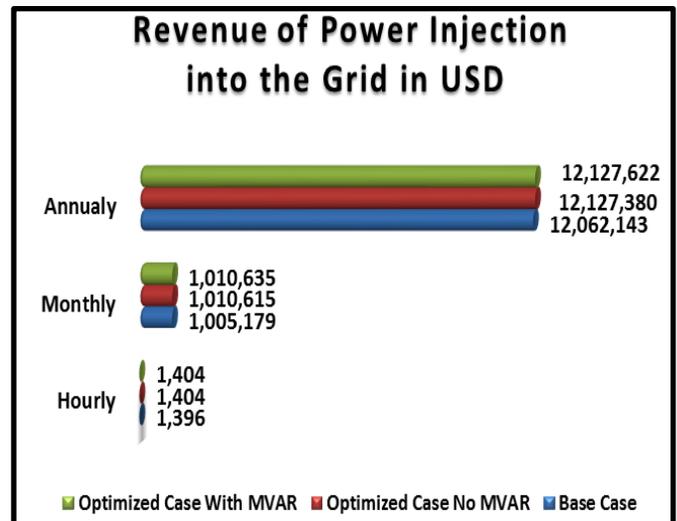


Figure 7. Revenue due to power injection in the grid

The figure illustrates the potential of the optimal scenarios in increasing the revenue using \$37.3 MWh tariff rate; \$65,200/year for the no MVAR optimal case and \$65,500/year for the with MVAR optimal case benchmarked to the base case.

VI. CONCLUSION AND FUTURE WORK

This paper presented the potential of minimizing the power system loss for a real-life hydrocarbon facility using the GA base approach considering no MVAR and with MVAR injection scenarios. Consequently, the increase in the injected power to the grid due to the loss optimization was also captured. The paper demonstrated that the reduction of power loss associated with MVAR injection is minimum. The economic advantages of the optimal scenarios modes versus the base mode were highlighted in this paper. The economic advantages of the with MVAR injection scenario compared to the no MVAR scenario did not support the shunt capacitor installations as the advantages are minimal. Improvement to the system buses voltage profile was shown to be a byproduct of the system power loss optimization. Future study may need to address the effectiveness of different selection, crossover and mutation methods in optimizing the system loss through GA evolutionary process.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Power System Planning Department/Saudi Aramco Oil Company, Brunel University and King Fahd University of Petroleum & Minerals for their support and encouragement throughout the study.

REFERENCES

- [1] The Electricity Co-Generation Regulatory Authority (ECRA), "Annual Peak Load for the Kingdom", 2014.
- [2] R. K. Kapadia and N. K. Patel, "Reactive Power Optimization Using Genetic Algorithm," IEEE, Engineering (NUICONE), Ahmedabad, 2013.
- [3] P. Yonghong and L. Yi, "An Improved Genetic Algorithm for Reactive Power Optimization," Control Conference (CCC), 2011 30th Chinese, pp.2105-2109.
- [4] M. R. Silva, Z. Vale, H. M. Khodr and C. Ramos, "Optimal Dispatch with Reactive Power Compensation by Genetic Algorithm," Transmission and Distribution Conference and Exposition, 2010 IEEE PES.
- [5] D. Guan, Z. Cai and Z. Kong, "Reactive Power and Voltage Control Using Micro-Genetic Algorithm," Mechatronics and Automation, Changchun, 2009, pp.5019-5024.
- [6] R. Suresh and N. Kumarappan, "Genetic Algorithm Based Reactive Power Optimization Under Deregulation," ICTES 2007, India, pp.150-155.
- [7] S. Durairaj and B. Fox, "Evolutionary Computation Based Reactive Power Optimization," Information and Communication Technology in Electrical Sciences (ICTES 2007), India, 2007, pp.120-125.
- [8] W. N. W. Abdullah, H. Saibon and K. L. Lo, "Genetic Algorithm for Optimal Reactive Power Dispatch," IEEE Catalog No: 98EX137, 1998, pp.160-164.
- [9] K. Iba, "Reactive Power Optimization by Genetic Algorithm," IEEE Trans. on Power Systems, Vol.9, No.2, 1994, pp. 685-692.
- [10] K. R. C. Mamandur and R. D. Chenoweth, "Optimal Control of Reactive Power Flow for Improvements in Voltage profiles and for Real Power Loss Minimization," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 7, 1981, pp. 3185-3193.
- [11] M. O. Mansour and T. M. Abdel-Rahman, "Non-Linear VAR Optimization Using Decomposition and Coordination," IEEE Trans. on Power Apparatus and Systems, PAS-103, No. 2, 1984, pp. 246-255.
- [12] Y. Hongwen and T. Junna, "Reactive Power Optimization Research of Power System Considered the Generation Transmission and Distribution," Industrial Technology, Chengdu, 2008.
- [13] M. A. Abido, "Intelligent Control Course Notes," King Fahad University of Petroleum & Minerals, 2007.
- [14] C. Belhadj and, M. A. Abido, "An optimized fast voltage stability indicator", IEEE Budapest Power Tech '99 Conference, Budapest, Hungary, 1999.
- [15] H. Saadat, "Power System Analysis", book, 2nd edition, 2004, pp.232-240.
- [16] M. T. Al-Hajri and M. Abido, "Assessment of Genetic Algorithm Selection, Crossover and Mutation Techniques in Reactive Power Optimization," IEEE, CEC 2009, May 8-21, 2009, pp.1005-1011.
- [17] F. Namdari, L. Hatamvand and N. Shojaei, "Simultaneous RPD and SVC Placement in Power Systems for Voltage Stability Improvement using SOA," IEEE UPEC 2014, Romania.

Chemistry-inspired, Context-Aware, and Autonomic Management System for Networked Objects

Mahmoud ElGammal and Mohamed Eltoweissy
 Department of Electrical and Computer Engineering
 Virginia Polytechnic Institute and State University
 Blacksburg, Virginia, USA
 {gammal,toweissy}@vt.edu

Abstract—We present a new method for network configuration and management in pervasive computing systems inspired by the Chemical Affinity theory. We coined our approach C_2A_2 : Chemistry-inspired, Context-Aware, and Autonomic management system for networked objects. The hypothesis behind C_2A_2 is that by paralleling the model of interaction that takes place between atoms during a chemical reaction, a form of collective intelligence emerges among system components enabling them to achieve a common global objective while relying primarily on preferences expressed at the individual level. In C_2A_2 , both physical and logical entities in the network are modeled as atoms with varying levels of affinity toward each other. Network reconfiguration is realized by breaking existing bonds between atoms and establishing new ones based on inter-atom affinities, which change continuously in response to context dynamics. Context is seen as having either a catalytic or inhibiting effect on reactions and is used to guide bond creation in favor of reactions that are most suitable to the situation at hand. Bonds are established by exchanging messages between atoms using a protocol that leverages the Affinity Propagation algorithm, which is used in C_2A_2 as a reaction execution engine. Finally, we use simulation to evaluate the performance of C_2A_2 in clustering and task assignment in wireless sensor networks.

Keywords—*affinity propagation; bio-inspired computing; chemical affinity; context awareness; Internet-of-Things; networked objects; pervasive computing.*

I. INTRODUCTION

In this paper we propose a new approach to network configuration and management in pervasive computing systems inspired by the Chemical Affinity theory, which was the key concept in the development of the idea of chemical equilibrium. The hypothesis behind our work is that by paralleling the model of interaction that takes place among atoms during a chemical reaction, we enable a form of collective intelligence to emerge among nodes, enabling them to achieve a common global objective while relying primarily on local decisions.

The objective of our work is to enable effective multi-mission Smart Cyber-Physical Spaces (CPSs) that are context-aware, adaptable, autonomic, and efficient. This would eventually allow for self-managing CPSs capable of hosting multiple applications operating in different contexts with competing demands, whereby for each context the system optimizes its different parameters to the situation at hand. We have coined our system C_2A_2 (short for Chemistry-inspired, Context-Aware, and Autonomic management system for networked objects). In C_2A_2 , context-awareness goes beyond affecting the operational state of the system to guiding its self-management, configuration, and optimization. Chemical

Affinity and Affinity Propagation [3] –inspired techniques for self-management make the ability to serve multiple concurrent goals efficiently an intrinsic property of the network – one that doesn't require special handling.

The motivation behind our work stems from the observation that a certain degree of congruence exists between the structure and interaction dynamics in a pervasive computing system and their analogues in several natural ecologies. Adding to that the possibility of modeling ecology dynamics using chemical reactions – which, as pointed out in [31], has been proven in [9] and [10] – this motivated us to explore the possibility of using the concept of chemical affinity to model different interaction patterns that take place in a pervasive computing system. The heterogeneous nature and rapidly changing structure of networks that underlie future pervasive computing environments require individual system components to possess a high degree of autonomy and adaptability. We believe that by incorporating the concepts of chemical affinity, affinity profiles, and affinity propagation as outlined in the remainder of this document, we can achieve a system that possesses such qualities. The notion of chemical affinity emphasizes the individuality of each system component, allowing it to freely express the services it can offer to the rest of the system, as well as what it needs to consume from other components in order for it to function properly. This emphasis on individuality is essential in a ubiquitous environment where the relationship between different actors in the system is often based on a supply and demand model. Moreover, the flexibility provided by allowing each entity in the system to dynamically vary its affinities toward other entities allows for fine-grained adaptation and provides a means for subsuming context dynamics into system operation. Finally, the utilization of affinity propagation permits the system to explore different solutions concurrently in the search space of any given problem, allowing for a sort of collective intelligence to emerge as individual entities take local decisions that serve high-level objectives.

The contributions of this work lie in the utilization of the following techniques in order to realize the aforementioned system characteristics:

- **Chemical affinity –inspired collective intelligence:** we introduce the notion of perceiving a CPS as a dynamic chemical solution and utilize the concept of chemical affinity to model the interactions between its components. Each node in C_2A_2 is governed by a mutable *affinity profile*, which determines its affinity to other entities in the system, such as events, tasks, users, or other nodes. Affinity profiles allow nodes to take individual decisions that ultimately serve a global system objective, and they serve to provide a

flexible interconnection framework, which plays a central role in realizing self-organization.

- **Affinity propagation:** we reutilize the Affinity Propagation clustering algorithm [3] as a reaction execution engine, which we then use for executing different network configuration and management tasks. Affinity Propagation is a clustering algorithm that relies on exchanging *messages* between data points to compute pair-wise similarities, which are then used to recursively combine similar points together. The algorithm need not be initialized with exemplar points, which is advantageous in dynamic environments where new resources might join or leave the system at any time.

The remainder of this paper is organized as follows. Section II provides some background on the chemical affinity theory and presents an overview of the related work in literature. Section III discusses the C_2A_2 system. Section IV presents a number of case studies and an experimental evaluation of the system performance. Finally, Section V concludes the paper and discusses future work.

II. BACKGROUND AND RELATED WORK

A common feature among most – if not all – natural computing models that draw their inspiration from the chemical reaction metaphor, is that the system state is represented as a fluid in which reagents of different types move freely and interact with each other according to predefined reaction rules. Developing concrete applications based on this concept requires mature models of computation that can be used to encode real-life problems using the chemical formalism and describe programs to solve them, as well as runtime systems that can actually execute these programs. The former field of study has seen significant research activity, ushered by the Γ language by Banatre et al [18] and continued through various other works such as the Chemical Abstract Machine by Berry et al [1], the Molecular Dynamics model by Bergstra et al [19], Membrane Computing (P Systems) by Păun [2], and more recently in the Biochemical Tuple Spaces model by Viroli et al [20]. Some of these works contributed incremental improvements over previous models while others offered entirely new approaches, but all have served to present the chemical metaphor as a mature and viable option for modeling computational processes, especially for applications where concurrency and self-organization are two necessary characteristics. However, despite the progress on this front, not as much attention has been given to the runtime systems on which chemical computing models can be executed [21].

The earliest runtime system for a chemical machine is perhaps the one described in [18], where an implementation of the Γ language on a massively parallel machine (aka the Γ -machine) is proposed. In order to evaluate a Γ program, the runtime system has to perform two tasks: (a) search for reagents that satisfy reaction conditions (in other words, determine which reactions to fire), and (b) applying the actions associated with fired reactions on the system. It can easily be shown that a roughly similar breakdown of tasks would also apply to any other runtime, not just the Γ -machine. The first task requires solving an NP-hard optimization problem, and with C_2A_2 , we attempt to put forward a practical, efficient, and scalable solution to this problem.

Existing runtime systems employ different approaches to address this problem. One approach can be described as the *search-and-match approach*, and it usually relies on some data structure that stores information about reagents and reaction rules where a search algorithm is then used to find reagents that satisfy the left-hand side conditions of any given reaction. The implementation proposed for the Γ -machine in [18] falls under this category. However, it can be considered more of a proof of concept as it assumes a number of processors equal to the number of reagents, which would be faced by strict scalability limits in reality. Additionally, it only considers one form of reactions (more specifically, reactions that take exactly two input values and produce two output values), which would impose further constraints on the practicality of this approach. More efficient methods belonging to this category have also been proposed, such as The Chemical Machine by Rajcsányi et al [22], which relies on the more sophisticated RETE pattern-matching algorithm [23].

Another approach that is mainly used for simulation but is also used in some runtime systems is the *computational chemistry approach*. Methods belonging to this category rely on algorithms that have long been used by theoretical chemists to solve many quantitative chemical problems using simulation with acceptable accuracy. Several algorithms have been developed under this category and which have been improving in efficiency over time, such as Gillespie's First Reaction Method [24], the Next Reaction Method by Gibson et al [25], Slepoy et al's constant-time Monte Carlo algorithm for simulating biochemical reaction networks [26], ALCHEMIST [27] and others. These methods offer a statistically correct depiction of the evolution of species concentration in a chemical solution over time, which is necessary in applications that rely on accurate simulation of the laws of chemical kinetics.

These two approaches have different points of strength and weakness. The *search-and-match* approach can be used to model the behavior of a self-organizing system in terms of microscopic interactions among its lowest-level components, which makes it a more versatile tool for modeling a wide range of applications. On the downside, it offers limited control on the macroscopic behavior of the whole system [28]. The *computational chemistry* approach on the other hand offers greater control over the macroscopic behavior of the system, which allows for better overall stability and predictability, but only if the target application lends itself easily to this approach, such as the case studies given in [27, 29, 30, and 28]. With C_2A_2 , we aim to combine some of the advantages of these two approaches.

III. THE C_2A_2 SYSTEM

Before we're able to utilize the chemical metaphor in our target domain, we must first map some key elements that would allow us to mirror the more complex operations that require interaction and cooperation among such elements. At the lowest level, the most basic building block of interest to us is the *atom*, which is paralleled by an individual entity in the modeled system (e.g. network node, event, user, etc.). Atoms from different elements possess distinct qualities, resulting in different affinities toward atoms from other substances. When an atom combines with other atoms, either from the same or

different substance, a *molecule* is formed, which represents a higher-order structure. The counterpart in the network would be a group of cooperating nodes that are structurally *less attached* to the rest of the network – for instance, a cluster within a clustered network, or a group of nodes assigned to a particular task. A molecule can be defined as a specific configuration of *bonds* among atoms of particular elements. In the network, such bonds are paralleled by network connections, where connections between nodes belonging to the same cluster (intra-cluster) mirror the intra-molecular forces holding a molecule together, and connections among different clusters (inter-cluster) would be similar to the weaker inter-molecular forces that exist between different molecules.

The objective of the network is to restructure itself by optimizing *bond* selection between nodes in order to better serve the tasks presented to it. The network is described as *stable* if no such bonds are in the process of being formed. By establishing all necessary bonds, the node groups (*molecules*) required to serve all present tasks are constructed, which parallels the state of *chemical equilibrium*. An important realization regarding the state of chemical stability, or equilibrium, is that despite the apparent steadiness of the system, the potential for dissolution and thereby the formation of new bonds between different reactants is always present. This possibility emanates from the fact that the state of stability is in reality the result of a balance between antagonistic forces that never cease to exist as long as the reactants themselves exist. Whenever a change is introduced in the system – as in the addition or removal of a reactant or a change in system conditions – existing affinity forces become unsettled and they compete again until equilibrium is gradually regained, possibly resulting in new products being formed. This latent potential, even under a stable or steady state, provides the system with a high degree of dynamism in the face of change.

A. Abstract System View

We now present an abstract view of the system we propose. The system is seen as a pool of networked resources with different roles, capabilities, and other attributes of different degrees of relevance according to hosted applications – such as function, geographic location, resource levels, and so forth. Throughout the lifetime of the system, it is presented with events that can be categorized into three classes: *application events*, *system events*, and *circumstantial events*. An example for application events is user queries, while an example for system events is change in resource levels, and an example for circumstantial events is any change in the conditions within which the system operates. The system is programmed to respond to each event by executing a set of tasks. In order to do that, a group of resources with appropriate capabilities must be selected and assigned to each task. This process involves optimizing resource selection, configuring selected resources, and reorganizing the pool of resources by establishing the required connections among its members.

B. C_2A_2 Abstraction Domains

One of the fundamental challenges that manifest themselves when embarking on the task of designing a pervasive computing system is the heterogeneity of its elements.

Ubiquity is a highly desirable feature in all but the most narrowly scoped of such systems, and this immediately entails the involvement of a host of highly diverse real-life entities. Such entities play different roles in the system in terms of whether they produce or consume information, the type of information they produce or consume, the tasks they partake during information processing, and the depth of their impact on how the system is configured and the selection of its short-term objectives. Accordingly, this implies that such entities would possess extremely heterogeneous and non-normalized characteristics, and consequently interaction models. It is then only imperative that an appropriate abstraction tool be utilized so as to normalize and homogenize such diverse elements of the system for it to be able to deal with them uniformly and efficiently. As illustrated in Fig 1, C_2A_2 can be perceived at three levels of abstraction, or as having parallel manifestations in three different domains, which are discussed below.

1) The Physical Domain

In the first – and lowest – level, lies the *physical domain*, which encompasses all the elements of the system in their real-life representation. These are heterogeneous physical and logical entities that are either among the system stakeholders or are necessary for its operation. The physical entities can be categorized into *active* and *passive* ones. Active entities are those that participate in imposing the (short-term) functional objectives of the system, in addition to elements that partake in the execution of such objectives. An example of the former is system users, whether they are human users or software agents, while an example of the latter is the various hardware components that carry on the execution of requests generated by users. These components can either be infrastructural (e.g. communication devices, generic processing devices, network health monitoring hardware, etc.) or application-specific, which are customized and properly equipped to generate, collect, and process the exact information needed by the possibly different applications running on the system. Passive entities in the physical domain are necessary for proper operation of the system, but they play a role that is rather assistive, as they mainly comprise resources that handle storing and managing network state and application data, deploying and distributing middleware updates, and overseeing the knowledgebase responsible for defining and formulating the system response to various events. The third and final element in the physical domain is the different types of events that constitute contextual information.

2) The Logical Domain

In the next level lies the *logical domain*, which constitutes the first level of abstraction. In this domain traditional object-oriented software modeling techniques are applied in order to represent all entities in the physical domain as well as their respective interrelationships. Each entity is represented by a proxy object that serves three purposes: (1) it provides a faithful depiction of the characteristics of its corresponding real-life object in the physical domain, represented using a normalized form that permits and simplifies manipulation via software, (2) it acts as an observer of its physical counterpart, and as an upstream channel that propagates any changes to its state to the higher layers of the system, and (3) it acts as a surrogate that provides access to the physical entity, allowing

control and configuration commands to flow downstream through the system to be finally effected in the physical domain.

3) The Chemical Domain

The third and final level is the *chemical domain*, which adds homogeneity to the normalized representation attained in the previous level. Semantic mapping is used to reduce the high dimensionality of the OOP representation, and produce a homogenous model where all entities are represented uniformly and their different characteristics are abstracted via different quantification techniques. Each entity is represented as an atom with an associated *affinity profile*. Events

constituting the context are represented as either catalysts or inhibitors, which according to each atom's affinity profile, either attract or repel an atom with a certain strength. This is used as a platform for executing *reaction rules* that emulate chemical reactions with the objective of reorganizing and reconfiguring the system, taking into consideration user-generated requests, the internal state of the system, as well as contextual information. In the rest of this document, we will be focusing more onto the components of this layer, which will be referred to as the Chemical Abstraction Layer (CAL), and discuss the various operations that take place therein.

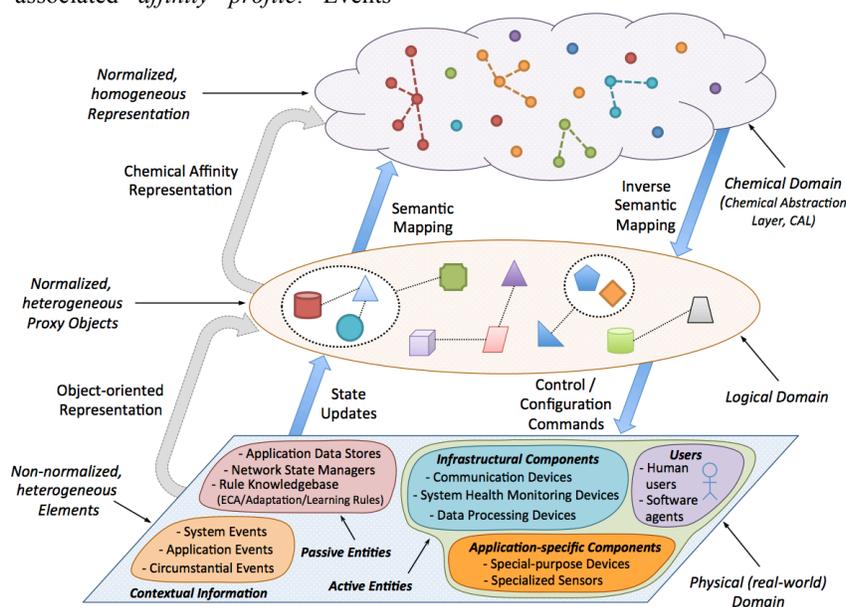


Figure 1. C₂A₂ at different levels of abstraction.

C. The Chemical Abstraction Layer (CAL)

In chemical-inspired systems (e.g. CHAM [1]), the state of a system is like a chemical solution in which floating reactants can interact with each other according to specific reaction rules. A certain stirring mechanism is responsible for causing motion within the solution, allowing for possible contacts between reactants. A solution can be heated to break complex molecules into smaller ones up to ions. Conversely, a solution can be cooled to rebuild heavy molecules from simpler reactants. Furthermore, to make it possible to model and solve problems of hierarchical structure, a molecule is allowed to contain a sub-solution enclosed in a membrane, which can be somewhat porous to allow communication between the encapsulated solution and its environment [1]. In this section we show how all of these concepts are realized in C₂A₂.

1) Affinity Profiles

One of the goals that we attempt to achieve in C₂A₂ is adaptability. In order to optimize the system under changing circumstances and varying tasks, the system configuration should be easily and dynamically adaptable to both external and internal stimuli. Several factors must be considered when assigning tasks, allocating resources, or carrying out any

network configuration operation. For instance, in a network of heterogeneous nodes, the decision to allocate a node to a certain task must take into consideration the compatibility between such task and the capabilities of the selected node. Other factors may also affect the decision, such as node location, its residual power, whether the node is allocated to another task with a higher priority, and so forth. We use the concept of affinity to express the strength of such compatibility.

In our proposed scheme, we would like each node to behave as independently as possible in order to minimize the communication overhead, but without compromising the functionality of the network as a whole. In other words, we would like to guide nodes to take local decisions that collectively serve global goals. Each real-life entity, be it physical or logical, is represented in CAL as a reactant (atom or molecule) with the capacity to interact with other reactants to establish new bonds and possibly disband existing ones. In C₂A₂ this capacity is expressed using an *affinity profile*. An affinity profile is akin to atomic valence, which determines the ability of an atom to form bonds with other atoms. Unlike valence, however, affinity profiles are mutable. In CAL, affinity between two reactants might change over time due to changes in context or in the internal state of the physical

domain counterpart of either reactant. This allows us to incorporate context dynamics in the system configuration loop and adapt each individual component accordingly.

An affinity profile can be modeled mathematically as a vector $P_i = \langle f_{ij} \rangle$, where f_{ij} is the affinity of reactant i toward reactant j . Affinities are expressed as real numbers where maximum affinity, minimum affinity, and neutrality are denoted by ∞ , $-\infty$, and 0 respectively. One advantage of this representation is its neutrality to the application-specific interpretation of affinity values, which provides a flexible and generic tool to carry out various system configuration tasks. CAL is absolutely agnostic to application semantics, and thus, the process of assigning initial affinities and updating them as different events occur during system operation is handled independently via a semantic mapping mechanism that resides between the logical layer and CAL. Such mechanism is cognizant of application semantics, contextual information, as well as the precise state of the entities being mapped, which is made possible by its access to their representation in the logical domain. Affinities are not necessarily symmetrical; i.e. f_{ij} and f_{ji} don't have to be equal, which allows for more independence for reactants to bond with the most suitable peer. As will be shown in the examples in Section IV, each entity independently controls its initial affinity toward other entities, and it is up to the reaction execution mechanism to resolve conflicts that arise between them. Initial affinities can be computed using a variety of ways that can be as plain or as elaborate as the system permits. For instance, neural networks can be used to produce the affinity values based on arbitrary inputs, including, but not restricted to, context parameters. This would provide a means for incorporating machine learning techniques, which can be used to retrain the neural networks in order to produce affinities that optimize system performance over time.

2) Context Representation

In CAL, context is seen as having either a catalytic or inhibiting effect on ongoing reactions. A catalyst is a substance that causes or accelerates a reaction without itself being consumed, while an inhibitor has the opposite effect. In addition to its congruity with the chemical frame of reference employed herein, we also find that this perception of context is in accordance with how context information is used in a context-aware system, where the presence of a certain context element (also referred to as *parameter*, *variable* or *dimension*) typically triggers an adaptational response where the system incorporates the newly sensed context information into the process of deciding how the system configuration should be changed to best serve the task at hand. The catalytic or inhibiting effect owing to the presence of a specific context parameter is exhibited at two levels. At a high level, it controls the overall attraction force by which a reaction rule pulls reactants. This has the effect of regulating the rate at which each reaction takes place by controlling the amount of reactants consumed by it. At a lower level, context is used to mutate the affinity profiles of individual reactants, leading to different bonding choices than what would have taken place in absence of contextual influence. Both effects are modeled as a multiplicative factor that either strengthens or weakens the

affinity between a rule and a reactant or between two reactants that are potentially to be consumed by the same reaction.

Now that we have discussed the role of affinity profiles, which is one of the key factors in determining the behavior of reactants in CAL and the influence of context on them, we now turn our attention to the final factor that constitutes the driving force behind reaction execution in CAL.

3) Affinity Propagation

Affinity propagation (AP) is a clustering algorithm that operates by exchanging messages between data points [3]. In essence, AP is a heuristic approach for finding an approximate solution to the maximization problem in graphs, which is known to be NP-hard [14]. AP relies on the max-product belief propagation algorithm [12, 13] to optimize an objective function that aims to maximize the sum of similarities between nodes and their exemplars. Despite being conceived as a clustering algorithm, AP is utilized in this work as the reaction execution engine in CAL, where it is responsible for exchanging affinities between reactants and updating them iteratively during the course of a reaction until final bonds are established. In order to explain how this is achieved, we first need to discuss how AP works in slightly more detail.

AP takes as input the initial measures of similarity between each pair of data points and simultaneously considers all data points as potential exemplars to each other. These values also include self-similarities, or *preferences* [3], which express the a priori willingness of individual points to become exemplars. Message exchange then takes place between data points until a high quality set of exemplars and corresponding clusters gradually emerge. In this paper, we use the binary variable model formulation of AP presented in [11] because of its ease of extension. In this model, cluster affiliations are obtained by computing an $N \times N$ array of binary variables h_{ij} using the factor graph shown in Fig 2, where $i, j \in \{1..N\}$ and N is the number of data points. $h_{ij} = 1$ if j is the exemplar of i , otherwise $h_{ij} = 0$. As pointed out in [11], in order for exemplar assignments to be valid, two constraints must hold in the solution obtained by the factor graph:

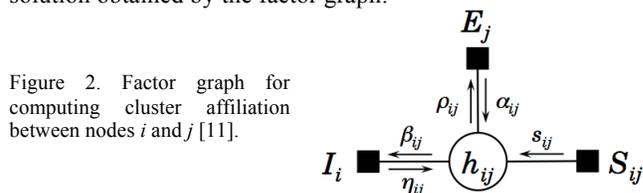


Figure 2. Factor graph for computing cluster affiliation between nodes i and j [11].

- **1-of-N Constraint:** Each node i must select exactly one exemplar (i.e. $\forall i \in [1..N]: \sum_{j=1}^N h_{ij} = 1$), which can be i itself if it is an exemplar node (in which case $h_{ii} = 1$).
- **Exemplar consistency constraint:** A node i may only choose j as its exemplar if j has chosen itself as an exemplar (i.e. $\forall i \neq j, h_{ij} = 1 \rightarrow h_{jj} = 1$).

It can easily be seen that any solution satisfying the two constraints above would be correct, although not necessarily optimal. There are three function nodes that contribute to the value of h_{ij} , which are shown in the factor graph in Fig 2: S_{ij} represents the similarity between nodes i and j , while I_i and E_j correspond to the two constraints above, respectively. Cluster affiliations are found by executing the max-sum (or log-

domain max-product) algorithm on the graph, which works by exchanging the five messages shown in Fig 2:

- s_{ij} is a scalar that expresses how similar node i is to node j . In this formulation of AP, this value also subsumes i 's preference.
- *Responsibility* messages (ρ_{ij}) are sent from a node i to a candidate exemplar j , and they represent the accumulated evidence for how well suited point j is to serve as an exemplar for point i .
- *Availability* messages (α_{ij}) are sent from j to i , reflecting the accumulated evidence of how appropriate it would be for point i to choose j as its exemplar, given the support j has from other candidate cluster members. Together with ρ_{ij} , these two messages are used to verify the fulfillment of constraint E_j .
- Similarly, β_{ij} and η_{ij} are used to test whether constraint I_i is violated for node i .

The derivation of these messages is given in detail in [11]. If either constraint is violated, its corresponding factor assumes a value of $-\infty$, otherwise it is set to θ . The objective function being maximized by the algorithm is given in (1). The function maximizes the sum of intra-cluster similarities but also ensures that invalid cluster assignments are discarded by incorporating the constraints, which would cause the function to yield $-\infty$ if either constraint is violated for any node.

$$\mathcal{F}(\{h_{ij}\}) = \sum_{i,j} s_{ij} h_{ij} + \sum_i I_i(h_{i,:}) + \sum_j E_j(h_{:,j}) \quad (1)$$

Evidently, (1) is only suitable for clustering. However, [11] shows how this formulation of AP can be extended to solve other problems, such as the Facility Location [15] and Maximum Bipartite Matching [16] problems. Shamaiah et al. [17] also apply the same technique to develop distributed routing mechanisms for networks. In C_2A_2 we utilize the extensibility of the binary AP model to produce cluster affiliations that obey the reaction rules defined in CAL. More precisely, we modify the factor graph and objective function above by manipulating node similarities and defining different constraints such that the resulting clusters would resemble the structure of molecules produced by the reaction rules. We now describe how this was achieved in detail.

4) Utilizing AP as a reaction execution engine in CAL

CAL is an implementation of a restricted P System [2] where reactions can only take one form in either direction. Reactions in the forward direction take the form $ca \rightarrow cu$, while reverse directions take the form $cu \rightarrow ca$. In both forms, c is an optional catalyst, and a and u are multisets such $|a| > I$ and $|u| = I$. The forward form is used to compose complex molecules from multisets of atoms and/or simpler molecules, while the reverse form is used to dissolve molecules produced previously by forward reactions to their simpler components, thereby returning them back to the environment. Both forms may optionally require the presence of a catalyst as a condition to be triggered. Reactions of different forms that produce an arbitrary number of outputs are not allowed.

Under this restriction, the problem of assigning reactants to forward reactions can be seen as a clustering problem in which reaction rules play the role of exemplars while reactants (atoms and molecules) play that of cluster members. However,

one property of P Systems necessitates a slight departure from traditional clustering, which is that reactions must be performed in a *maximally parallel* way. This means that reactants are assigned to rules until no further assignments are possible, which imposes the consequence that a single rule may assume exemplarity of multiple cluster instances, all of identical structure, where the members of each instance have a one-to-one correspondence to the elements in the left-hand side multiset in the forward reaction form, a .

This reformulated clustering problem can be solved by constructing a factor graph as the one shown in Fig 3. In this graph, we assume that the system contains N reactants and R reaction rules, and the goal is to match subsets of these N reactants with reaction rules such that each subset (or reactant group) is congruent with the left-hand side multiset of the reaction it is matched with (the multiset a in the forward reaction form), and no single reactant is involved in more than one reaction. One important practical consideration to note is that even though multiple instances of the same reactant type might exist simultaneously in the system, their inclination to partake in the same reaction may differ due to application semantics. Similarly, if the left-hand side multiset of a reaction consists of two reactant types, r_1 and r_2 , where multiple instances exist of the former but only one exists of the latter, the different r_1 instances are not to be considered interchangeable, but rather, the one with the highest affinity to the r_2 instance should be picked if possible. This is important because it might not be preferable for the physical entities that correspond to the reactant instances to communicate with each other for any reason, such as being physically separated by a large distance. Because of this, the input provided to the algorithm consists of similarities between reactants and reaction rules as well as reactants and each other, where the latter is based entirely on application semantics, while the former is based additionally on whether the reactant appears in the left-hand side multiset of the rule. If it does not, then the two are incompatible and their similarity is set to $-\infty$. In this scheme, self-similarities (or preferences) have no significance for reactants, whereas for reaction rules, they represent the context influence.

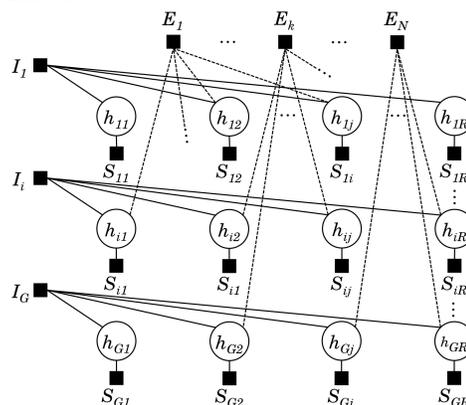
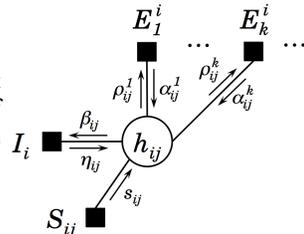


Figure 3. Factor graph of reformulated clustering problem for executing reactions in CAL.

As illustrated in Fig 3, the factor graph is 2-dimensional and is constructed such that reaction rules are organized horizontally and are indexed by the variable $j \in \{1..R\}$, while

symbols representing reactant groups are organized vertically and are indexed by the variable $i \in \{1..G\}$. Reactant groups represent the elements of the power set over all reactants, which has a size of 2^N . Obviously, this can be prohibitively expensive. In practice, however, reactant groups that don't appear in any reaction rules as well as those that don't satisfy a minimum similarity threshold among its members can be eliminated, reducing the size of the set by several orders of magnitude. Similar to the original AP formulation, assigning a value of 1 to h_{ij} indicates that the reactant group i is to be consumed by reaction rule j . Due to the fact that a single reactant may belong to multiple groups, if group i is assigned to a rule (i.e. $\exists j$ s.t. $h_{ij} = 1$), we must guarantee that all reactants in group i are not assigned to any other rule. This is achieved by adding the *single-assignment constraint* factors (labeled $E_{\{1..N\}}$ in the graph), one for each reactant, which collectively guarantee that if $h_{ij}=1$ then all other variables that appear in rows corresponding to groups that share any reactants with i are set to 0. Finally, the I_i factors represent the *1-of-N constraint*, which plays the same role as in the standard AP problem.

Figure 4. Factor graph fragment for computing the affiliation of reactant group i with reaction rule j .



A fragment of the factor graph for computing a single variable is shown in Fig 4, which is similar to that of AP (Fig 2) except that the exemplar consistency constraint E_j is now replaced by a varying number of single-assignment constraints E_k^i (the notation stands for the constraint in $E_{\{1..N\}}$ that corresponds to the k^{th} reactant in group i). The number of single-assignment constraints imposed on each h_{ij} variable is equal to the number of reactants in group i , and each constraint is connected to all other $h_{gr} | (g,r) \neq (i,j)$ where reactant group g includes the reactant associated with the constraint. As shown in (2) and (3), the *1-of-N* constraint is satisfied only if each reactant group is assigned to at most one reaction rule, while the single assignment constraint is satisfied if no more than one of the h_{ij} variables linked to the constraint is set to 1. The objective function in (4) is similar to that of AP, where the sought solution maximizes the sum of similarities between reactant groups and reaction rules while satisfying all constraints.

$$I_i(h_{i:}) = \begin{cases} 0 & : \text{if } \sum_{j=1}^R h_{ij} \leq 1 \\ -\infty & : \text{otherwise} \end{cases} \quad (2)$$

$$E_k(h_{:k}^i) = \begin{cases} 0 & : \text{if } \sum_{(i,j) \neq (i,k)} h_{ij}^k \leq 1 \\ -\infty & : \text{otherwise} \end{cases} \quad (3)$$

$$\mathcal{F}(\{h_{ij}\}) = \sum_{i=1}^G \sum_{j=1}^R s_{ij} h_{ij} + \sum_{i=1}^G I_i(h_{i:}) + \sum_{k=1}^N E_k(h_{:k}^i) \quad (4)$$

Messages passed between nodes in the modified graph have the same semantics as in the original AP formulation. However, due to the varying number of constraints imposed on each variable and their slightly different roles in our problem, the equations for computing the values of these messages must

be rederived. The binary formulation of AP simplifies this process, as it lets us derive the formula for each message by calculating its value for each setting of the hidden binary variable h_{ij} then taking the difference [11]. We now discuss the derivation of the five message types shown in Fig 4:

- s_{ij} : this message is independent of h_{ij} and it represents the initial affinity between reactant group i and reaction rule j . If reactants of group i match those of rule j , then the value of the message is set to the average similarity between each reactant in the group and the reaction rule, multiplied by the preference value of the rule. Otherwise, the pair is not a candidate for matching and the message is assigned a value of $-\infty$. In the former case, the average is taken as a normalization mechanism for countering the fact that the number of reactants (and subsequently constraints) may vary from one reactant group to another. Multiplying by the preference value is used as a means for adjusting the rate at which the reaction takes place as a result of contextual influence.
- β_{ij} and η_{ij} have the same role as in AP, which is to enforce the *1-of-N* constraint, and thus their formulas are unchanged, except that the average over the α messages is taken for the same reason explained earlier.
- α_{ij}^k : this message represents the accumulated evidence of how appropriate it would be for the k^{th} reactant in group i to partake in reaction rule j . To calculate its value, we fix the value of h_{ij} to 1 or 0, then we find an optimal assignment for the other variables associated with the same reaction. When $h_{ij} = 1$, this means that the k^{th} reactant (and all of group i) is assigned to rule j , and thus all other variables that correspond to reactant groups that contain the same reactant must be set to 0, yielding $\alpha_{ij}^k(1) = \text{avg}_{(g,r) \neq (i,j)} \{ \rho_{gr}^k(0) \}$. For $h_{ij} = 0$, the other variables are unconstrained and the maximum over both configurations should be taken, resulting in $\alpha_{ij}^k(0) = \text{avg}_{(g,r) \neq (i,j)} \{ \max_{h_{gr}} \rho_{gr}^k(h_{gr}) \}$. We obtain α_{ij}^k by taking the difference between $\alpha_{ij}^k(1)$ and $\alpha_{ij}^k(0)$ where we end up with the formula in (8), in which we have relied on the fact that $x - \max(x, y) = \min(0, x - y)$.
- ρ_{ij}^k : this message represents the accumulated evidence for how well suited reaction j is to consume the k^{th} reactant in group i , and it has a similar formula as that of AP but takes into consideration the availabilities of the other reactants involved in the group. A summary of the formulas for all messages is given in (5) – (9):

$$s_{ij} = \begin{cases} p_j \cdot \text{avg}_k \{ \text{sim}(k^i, j) \} & : \text{if } i \equiv \text{rule } j \\ -\infty & : \text{otherwise} \end{cases} \quad (5)$$

$$\beta_{ij} = s_{ij} + \text{avg}_{k^i} \{ \alpha_{ij}^k \} \quad (6)$$

$$\eta_{ij} = -\max_{r \neq j} \beta_{ir} \quad (7)$$

$$\alpha_{ij}^k = \text{avg}_{(g,r) \neq (i,j)} \{ \min(0, \rho_{gr}^k) \} \quad (8)$$

$$\rho_{ij}^k = s_{ij} + \eta_{ij} + \text{avg}_{r^i} \{ \alpha_{ij}^r \}_{r \neq k} \quad (9)$$

When the algorithm terminates, decoding the final solution is straightforward. Reaction rules with any variables set to 1 in their respective columns are considered to have been triggered. Those reactions are denoted by j s.t. $\sum_{i=1}^G h_{ij} \geq 1$. Similarly, a

reactant group i is considered to have been consumed by a rule if one of the variables in its row is set to 1, i.e. $\sum_{j=1}^N h_{ij} > 0$. Individual reactants within each such group are substituted by a molecule that represents the whole group in the system, while all other reactants are remain unaffected.

IV. EVALUATION

We now provide a number of examples where we put the concepts behind C_2A_2 to use. This serves to elucidate how such concepts can be applied in a concrete application and also to gauge their feasibility and efficiency in a practical setting.

A. Clustering in Wireless Sensor Networks

The first application we use to present C_2A_2 is clustering in wireless sensor networks (WSNs). In WSNs, clustering is often used to increase the longevity of the network and ensure balanced resource utilization. In its most basic form, WSN clustering involves just two parameters: node location and residual energy. In this model, each node is represented as an atom with an affinity profile that specifies its affinity toward other nodes in the network. The affinity profile can be sparse, where missing values are taken to mean $-\infty$, indicating that the two nodes can never be in the same cluster (due to their being too far from each other, for instance). Messages exchanged between nodes are calculated using the equations (10) – (13), which are a slight variation from the original AP formulas:

- Similarities between each pair of nodes are calculated according to (10), which in this model is designed to minimize the sum of squared error (i.e. the distance between each node and its exemplar).
- Availability of a node i to serve as an exemplar for another node j is initialized as shown in (11), which takes into consideration the exponential increase in communication cost as the distance between the two nodes increases. In subsequent iterations, availabilities are updated using (12) which takes into account the accumulated evidence that node j should be chosen as an exemplar in addition to the positive responsibilities received by j from other nodes.
- Responsibility of a node j to serve as an exemplar for a node i (from the latter's point of view) is calculated using (13), which relies on the similarity between the two nodes but also taking into consideration the suitability of i to be chosen as an exemplar by other nodes.

$$s(i, j) = c_1 \cdot \text{distance}(i, j)^2 \quad (10)$$

$$a_o(i, j) = c_2 \cdot \frac{\text{residual_power}(i)}{\text{distance}(i, j)^2} \quad (11)$$

$$a(i, j) = \min\{0, r(j, j) + \sum_{k: k \neq \{i, j\}} \max\{0, r(k, j)\}\} \quad (12)$$

$$r(i, j) = s(i, j) - \max_{k: k \neq j} \{a(i, k) + s(i, k)\} \quad (13)$$

In this experiment we used ns-2 [8] to compare our results to those obtained using two widely used WSN clustering protocols; generalized LEACH [4] and Average Minimum Reachability Power (AMRP) HEED [5]. The simulation parameters are identical to those used in [5], and the experiments were carried out using ns-2 [8]. The sensors are deployed randomly in a $100 \times 100 \text{ m}^2$ area with a base station located at (50,175). During each TDM frame, each node collects data and sends a data packet to its cluster head, and each cluster head fuses these packets into one message and

sends it to the base station. Clustering is triggered at the end of each round, where a round is 5 TDM frames.

The simulation results are shown in Fig 5, which shows the ratio of the total number of messages to the total number of nodes. The ratio seems to follow a logarithmic curve, where the overhead is inversely proportional to the number of nodes, approaching an asymptote of 1. The effect of this on network longevity is illustrated in Fig 6, where WSN clustering using C_2A_2 can be seen to outperform both LEACH and HEED clustering algorithms in terms of the number of clustering rounds that can be performed under identical conditions.

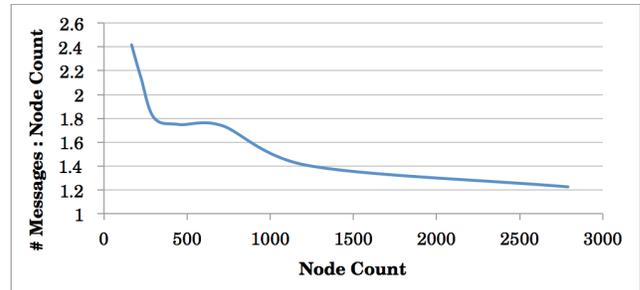


Figure 5. The ratio of the total number of messages exchanged during each clustering round is inversely proportional to the number of nodes with an asymptote of 1.

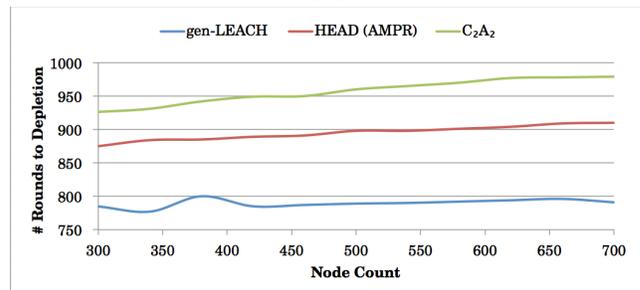


Figure 6. Number of clustering rounds until power depletion for different deployment sizes.

B. Constrained Task Assignment

The next case study we utilize to demonstrate C_2A_2 is the problem of constrained task assignment in a network of heterogeneous nodes, which has been formulated in several works in the literature, such as [6] and [7]. The problem can be stated as follows: given a set of tasks T and a set of nodes N , where each task $t \in T$ is associated with a demand vector $d(t)$, and each node $n \in N$ is associated with a supply vector $p(n)$, it is required to find an assignment $M(N \rightarrow T)$ such that:

- For each task $t \in T$, M must contain an assignment associating t with a set of nodes whose aggregate supply satisfies $d(t)$.
- The assignment M should be selected such that the objective function $f = \sum_{t \in T} w(t) - \sum_{m \in M} c(m_{n \rightarrow t})$ is maximized, where $w(t)$ is the reward gained by the system for executing task t , and $c(m_{n \rightarrow t})$ is the cost incurred by the system for assigning node n to task t .

This problem is known to be reducible to the sub-graph isomorphism problem, which is NP-Complete, and therefore finding an optimal solution (denoted henceforth as f^*) could be prohibitively expensive for large inputs. However, the problem

is particularly suited for C_2A_2 since it is aligned with the system's ability to reconfigure network resources into separate taskforces. The goal is to approximate f^* better than typical greedy solutions while keeping the cost of obtaining the solution within the same order of magnitude. We would like each taskforce to be represented in CAL as a molecule that has an atom (corresponding to a task) in its center with bonds linking it to a number of atoms representing the nodes assigned to the task. The formation of such bonds depends on the mutual affinity between a task and a node, which is based on matching the demand and supply vectors, and gauging the reward gained and the cost incurred by the system. It is important to note that a single node atom might be involved in two molecules, which could happen if the supply vector of the node can satisfy the demand of more than one task. The AP parameters we use to achieve this can be calculated as follows:

- To ensure that bonds are only created between task atoms and node atoms (and not between two tasks or two nodes), the similarity between two tasks or two nodes is set to $-\infty$, indicating absolute dissimilarity.
- To guide bond formation such that task atoms are always in the center of molecules and node atoms are always peripheral, the preference (or self-similarity) of a task atom is set to $+\infty$ while that of a node atom is set to $-\infty$ (as mentioned previously, the self-similarity parameter determines the willingness of a data point to become an exemplar).
- Since we want all task atoms to be exemplars and all node atoms to be followers, the responsibility of a task atom and the availability of a node atom are both set to $-\infty$. The justification of this is that no atom should be *responsible* – in AP terms – toward a task atom, since the latter is always an exemplar. Similarly, a node atom is never *available* to any other atom, since it should never be an exemplar.
- Similarity between a node and a task is based on the compatibility between their respective supply and demand vectors, as reflected in (14), which also considers the reward associated with the task.
- Availability messages sent from a task to a node are initialized according to (15), which takes into account the cost associated with the node in addition to its ability to satisfy the demand of the task.
- Availabilities and responsibilities are updated using the same equations shown in (12) and (13), respectively.

$$s(n, t) = c_1 \cdot w(t) \cdot \{|i: p(n)[i] \geq d(n)[i]\} \quad (14)$$

$$a_o(n, t) = c_2 \cdot \frac{|i: p(n)[i] \geq d(n)[i]|}{c(m_{n-t})} \quad (15)$$

The efficiency of our task assignment solution in approximating f^* is shown in Fig 7, where the obtained results are compared to the optimal solution as well as a greedy solution for different combinations of node and task counts. In an average of 100 runs, our approach consistently outperforms the greedy solution, and is a very close approximation of f^* . Moreover, the average number of AP iterations required was below 10 in our experiments, which means that the improvement over the greedy solution does not come at the expense of a much higher time complexity, as seen in Fig 8 which shows a logarithmic scale of the time complexity of the three methods. Our solution is slightly more costly than the

greedy solution, but lies within the same order of magnitude. Both algorithms, however, are significantly faster than the optimal one.

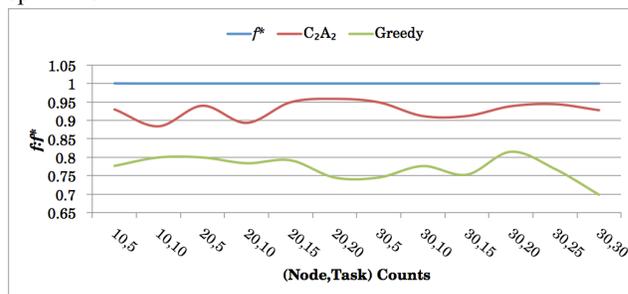


Figure 7. Normalized f^* vs. C_2A_2 and greedy approximations.

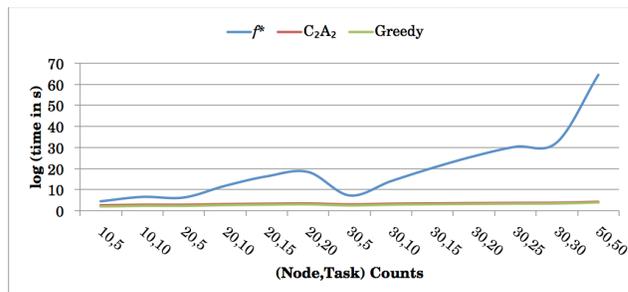


Figure 8. Time complexity (logarithmic scale).

V. CONCLUSION

In this paper, we presented C_2A_2 , a chemistry-inspired, context-aware, and autonomic management system networked objects. We proposed a framework for implementing a pervasive computing environment built around the chemical affinity theory. We presented an abstraction through which network components can be mapped to the chemical domain, allowing us to carry out several network operations by simulating the interaction model that takes place between atoms during a chemical reaction. We introduced the concept of dynamic context-aware affinity profiles, which govern the behavior of individual system components, ensuring adaptability in response to context changes and other interesting events. We also extended and repurposed the Affinity Propagation clustering algorithm as a reaction execution engine in our Chemical Abstraction Layer (CAL), allowing distributed exchange of affinities among individual nodes while steering them toward convergence on a common goal. We used simulation to verify the efficacy of our approach using the problems of clustering and constrained task assignment in WSNs. As a natural extension to this work, we are exploring the utilization of reinforcement learning techniques and exploratory self-adaptation, where the system associates past decisions with the monitored effect on performance, thereby allowing the system to self-optimize in anticipation of potential events expected to take place in the future.

REFERENCES

- Berry, G. and G. Boudol (1992). "The Chemical Abstract Machine." Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. POPL '90.

- [2] Păun, G. (2003). "Membrane Computing." *Fundamentals of Computation Theory*.
- [3] Brendan Frey and Delbert Dueck (2007). "Clustering by passing messages between data points." *Science*.
- [4] Heinzlman, W.R. and H. Balakrishnan (2000). "Energy-Efficient Communication Protocol for Wireless Microsensor Networks." *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*.
- [5] O. Younis, S. Fahmy, and P. Santi (2004). Robust communications for sensor networks in hostile environments. *Proceedings of the 12th IEEE International Workshop on Quality of Service (IWQoS)*.
- [6] Zhao, B., M. Wang, et al. (2008). "Topology-Aware Energy Efficient Task Assignment for Collaborative In-Network Processing in Distributed Sensor Systems." *Distributed Embedded Systems: Design, Middleware and Resources*.
- [7] Xu, S. (2010). "Energy-efficient task assignment of wireless sensor network with the application to agriculture." Iowa State University Digital Repository.
- [8] Network Simulator (ns-2): http://nsnam.isi.edu/nsnam/index.php/User_Information
- [9] Berryman, A. A. (1992). "The Origins and Evolution of Predator-Prey Theory." *Ecology* 73, 5 (October): 1530-1535.
- [10] Gillespie, D. T. (1977). "Exact Stochastic Simulation Of Coupled Chemical Reactions." *The Journal Of Physical Chemistry* 81, 25: 2340-2361.
- [11] Givoni, I. (2012). "Beyond Affinity Propagation: Message Passing Algorithms for Clustering." PhD Thesis, University of Toronto.
- [12] Pearl, J. (1988). "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference."
- [13] Kschischang, F.R. et al. (2001). "Factor Graphs and the Sum-Product Algorithm." *IEEE Transactions on Information Theory* 47, 2 (February): 498-519.
- [14] Dueck, D. (2009). "Affinity Propagation: Clustering Data by Passing Messages." PhD Thesis, University of Toronto.
- [15] Drezner, Z. (1995). "Facility location: a survey of applications and methods." Springer Verlag.
- [16] Cormen, T. H. et al. (2001). "Introduction to Algorithms, Second Edition." The MIT Press.
- [17] Shamaiah, M. et al. (2011). "Distributed routing in networks using affinity propagation." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011: 3036-3039.
- [18] J.-P. Banâtre et al. (1988). "A Parallel Machine for Multiset Transformation and its Programming Style." *Future Generation Computer Systems* 4(2): 133-144.
- [19] Bergstra, J. and I. Bethke (2002). "Molecular Dynamics." *The Journal of Logic and Algebraic Programming*, 51: 193-214.
- [20] Viroli, M. and M. Casadei (2009). "Biochemical Tuple Spaces for Self-organising Coordination." *Proceedings of the 11th International Conference on Coordination Languages and Models (COORDINATION 2009)*, 5521: 143-162.
- [21] Kreyssig, P. and P. Dittrich (2011). "On the Future of Chemistry-Inspired Computing." *Organic Computing – A Paradigm Shift for Complex Systems*. C. Müller-Schloer, H. Schmeck and T. Ungerer, Springer Basel, 1: 583-585.
- [22] Rajcsányi, V. and Z. Németh (2012). "The Chemical Machine: An Interpreter for the Higher Order Chemical Language." *Euro-Par 2011: Parallel Processing Workshops*.
- [23] Forgy, C. (1982). "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem." *Artificial intelligence*, 19.
- [24] Gillespie, D. T. (1977). "Exact Stochastic Simulation Of Coupled Chemical Reactions." *The Journal Of Physical Chemistry* 81, 25: 2340-2361.
- [25] Gibson, M. and J. Bruck (2000). "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels." *The Journal of Physical Chemistry*, 104: 1876-1889.
- [26] Slepoy, A., et al. (2008). "A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks." *The Journal of Chemical Physics* 128(20).
- [27] Pianini, D., et al. (2013). "Chemical-oriented Simulation of Computational Systems with Alchemist." *Journal of Simulation*, 7: 202-215.
- [28] Monti, M., et al. (2013). "Chemistry-Inspired Algorithm for Emergent Distributed Consensus in WSNs." *Distributed Computing in Sensor Systems, 2013 IEEE International Conference on*.
- [29] Meyer, T. (2010). "On Chemical and Self-healing Networking Protocols." *PhD Thesis, University of Basel*.
- [30] Monti, M., et al. (2013). "Stability and Sensitivity Analysis of Traffic-Shaping Algorithms Inspired by Chemical Engineering." *Selected Areas in Communications, IEEE Journal on*, 31: 1105-1114.
- [31] Viroli, M., M. Casadei, et al. (2011). "Spatial Coordination of Pervasive Services through Chemical-Inspired Tuple Spaces." *ACM Transactions on Autonomous and Adaptive Systems* 6(2).

An Approach to Model Resources Rationalisation in Hybrid Clouds through Users Activity Characterisation

Giovanni Battista Barone, Davide Bottalico,
Rosanna Campagna and Giuliano Laccetti
University of Naples Federico II
Naples, Italy
Emails: {gbbbarone, davide.bottalico,
rosanna.campagna, giuliano.laccetti}@unina.it

Vania Boccia
Italian National
Institute of Nuclear Physics
Seat of Naples, Italy
Email: vania.boccia@na.infn.it

Luisa Carracciuolo
Italian National
Research Council
Naples, Italy
Email: luisa.carracciuolo@cnr.it

Abstract—In recent years, some strategies (e.g., server consolidation by means of virtualisation techniques) helped the managers of large Information Technology (IT) infrastructures to limit, when possible, the use of hardware resources in order to provide reliable services and to reduce the Total Cost of Ownership (TCO) of such infrastructures. Moreover, with the advent of Cloud computing, a resource usage rationalisation can be pursued also for the users applications, if this is compatible with the Quality of Service (QoS) which must be guaranteed. In this perspective, modern datacenters are “elastic”, i.e., able to shrink or enlarge the number of local physical or virtual resources from private/public Clouds. Moreover, many of large computing environments are integrated in distributed computing environment as the grid and cloud infrastructures. In this document, we report some advances in the realisation of a utility, we named *Adaptive Scheduling Controller (ASC)* which, interacting with the datacenter resource manager, allows an effective and efficient usage of resources, also by means of users jobs classification. Here, we focus both on some data mining algorithms which allows to classify the users activity and on the mathematical formalisation of the functional used by ASC to find the most suitable configuration for the datacenter’s resource manager. The presented case study concerns the SCoPE infrastructure, which has a twofold role: local computing resources provider for the University of Naples Federico II and remote resources provider for both the Italian Grid Infrastructure (IGI) and the European Grid Infrastructure (EGI) Federated Cloud.

Keywords—*Adaptive scheduling and resources management; Virtualisation and Cloud computing; large scale and distributed systems; data analysis.*

I. INTRODUCTION

The TCO of large scale computing systems, with an emphasis on systems built to support a wide range of users and different applications includes, among others, the initial hardware cost (for computing nodes, storage systems, racks, facilities, etc.), the personnel/system administrator costs (salaries for software and hardware maintenance requiring specialised know-how), business premises, and energy costs (which includes the additional power requirements for cooling and power delivery inefficiencies) [1].

The aim of a good system manager is the TCO minimisation, that can be performed also by means of the resources rationalisation (e.g., through the reduction of the number of

active resources on the basis of the effective jobs request, the use of virtual resources by means of cloud computing paradigms, and so on). All said, however, without neglecting users satisfaction. Users of a large general purpose system can have conflicting demands (i.e., short versus long jobs, sequential versus parallel applications, etc.). For this reason the achievement of all user satisfaction is a hard task for any system manager.

In our previous works (e.g., in [2]), we described some issues related to the design and the implementation of a control system (ASC) which, using an “adaptive” approach in job scheduling policy, allows a balanced, effective and efficient use of computational resources. In particular, ASC, on the basis of a *jobs classification*, is asked to dynamically identify, given historical data, the most “suitable” configuration of the resource manager/scheduling systems for the current jobs work flow.

With the advent of new recent distributed paradigms, e.g., Cloud computing, modern datacenters are being more “elastic” (according to the U.S. National Institute of Standards and Technology (NIST) characterisation of Cloud computing based infrastructures) [3]), i.e., able to shrink or enlarge on demand the number of active resources, by recruiting both local physical (and virtual) resources and remote resources from public Clouds. Moreover, many of large computing environments are integrated in distributed computing environment as the grid and cloud infrastructures. All these above described issues increase the dynamic nature of the current computing environments.

In this paper, we provide a description of the advancements in the mathematical formalisation and implementation of ASC for modern datacenters and we report the work made toward a complete jobs classification by means of data mining techniques.

In Section II, we provide some details on the related works in the field of adaptive scheduling, focusing on all the key differences between the approaches existing in literature and ours; in Section III, we report a short description of the ASC system, with a focus on its mathematical formalisation; in Section IV, we describe the case study of the SCoPE computing infrastructure at University of Naples Federico II reporting some results related to the use of *Data Mining* techniques for job classification in a “production context”.

The SCoPE infrastructure has a twofold role: local computing resources provider for the University of Naples Federico II and remote resources provider for both the IGI [4] and the EGI Federated Cloud [5].

II. RELATED WORK

An adaptive solution to the scheduling problem, in the sense used by Casavant and Kuhl [6], is able to change dynamically algorithms and parameters that define the scheduling policy, according to the previous and current behaviour of the system in response to previous decisions taken by the scheduling system itself.

The most common property to search for a scheduler (or resource management subsystem) is dynamicity. In a dynamic scenario, the scheduler takes into account the current state of affairs as it perceives them in the system. This is done during the normal operation of the system under a dynamic and unpredictable load. In an adaptive system, the scheduling policy itself reflects changes in its environment (the running system).

A preliminary approach in jobs scheduling, such as those described by Serazzi and Calzarossa [7], exists in modelling adaptive control systems able to maximise a given performance criterion, such as system throughput. However, in the last years the heterogeneity of applications using general purpose computational grows together with the complexity of resource requirements. Maximising system throughput shouldn't longer be the only requirement for a scheduling scheme. The quality of service perceived by the user offers an instructive example where the solution of the problem is achieved in a "acceptable" time [8]. Recent approaches in jobs scheduling take into account both *efficiency* and *fairness* for homogeneous workloads [9], but the open challenge is to achieve the same goal for *not* homogeneous workloads.

If we have in mind what Feitelson says in [10]:

In reality, the metrics attempt to formalise the real goal of a scheduler:

- Satisfy the users,
- Maximise the profit.

more sophisticated and finer-grained resource coordination mechanisms are required.

If we consider the problem of user communities heterogeneity and dynamicity using a large scale computational resource, an "adaptive" approach to the job scheduling seems to be a promising way for getting the right trade-off among different, and often conflicting, classes of applications demands sharing the same set of resources. Moreover, the overall set of resources, generally included in a modern "elastic" datacenter, is dynamic and heterogeneous itself (because of the chance to include different resources from Clouds).

Here, we propose a pragmatic approach to solve this kind of problem. The approach is led by the fact that we consider the overall set of users organised in communities each of them having almost homogeneous requirements. The above conditions induce a classification of the users (or equivalently of the jobs) into different classes: the terms "users" and "jobs" are intended to represent the same entities.

The idea to organise jobs into groups to improve the solution of scheduling problem on complex computing systems, from High-Performance Computing (HPC) system to

the distributed infrastructures, is not new [11][12][13]. The principal aim of all those works is to present and to discuss the task distribution problem onto HPC and distributed systems to improve performance and load balancing of the applications.

Our approach is based, instead, on the idea of organising jobs in groups or *clusters* "naturally" deriving from the nature of our audience. The aims are mainly 1) to achieve the efficiency of the entire system rather than the performance of a single application and 2) to reduce the computational complexity for the optimisation problem solution required to find the most "suitable" configuration of the system.

To get an automatic classification of jobs we use Data Mining procedures, i.e., "data clustering". The term "data clustering" refers to the process of the Data Mining techniques aimed to divide into natural groups the instances represented by data [14].

There are different ways to represent results of clustering. The groups to be identified may be exclusive (so that any instance belongs to only one group) or may be overlapping (so that an instance may fall into several groups). Besides, they may be probabilistic, whereby an instance belongs to each group with a certain probability. They may be hierarchical, such that there would be a crude division of instances into groups at the top level, and each of these groups is refined further-perhaps all the way down to individual instances. In this work, we consider algorithms building clusters in numerical domains, partitioning instances into disjoint clusters.

According to the approach described by Estivill-Castro [15], in order to define the "data clustering" problem we have to:

- express assumptions on the model describing the nature of the data (e.g., a probabilistic model),
- formulate the mathematical model for the problem (e.g., an optimisation problem)
- choose the solving algorithm for the problem (e.g., the *k-means* algorithm [14])

So, as starting point and "proof of concept" for the validation of our approach, we considered algorithms based on the classic clustering algorithm *k-means*. Such algorithm can usefully be used to iteratively compute an approximation to the solution of the minimisation problem of a functional, based on an Euclidean norm, depending on data with multivariate normal distribution [15].

In the next section, we provide a mathematical formalisation for ASC based on the hypothesis that a heterogeneous work flow can be partitioned into homogeneous classes of jobs. The way we used to do this partitioning is described in Section IV.

III. ASC: ARCHITECTURE AND OPERATING MODEL

A. Architecture description

We call "adaptive" a system able to "reconfigure" itself on the basis of changing in user typology. Such a mechanism, analysing system behaviour by some *key-statistics* (e.g. depending on queue waiting time, jobs throughput, resource usage, and so on), dynamically defines a new set of scheduler *key-parameters* values. The scheduler's new configuration has to meet both user satisfaction and efficiency/productivity in computational resource usage.

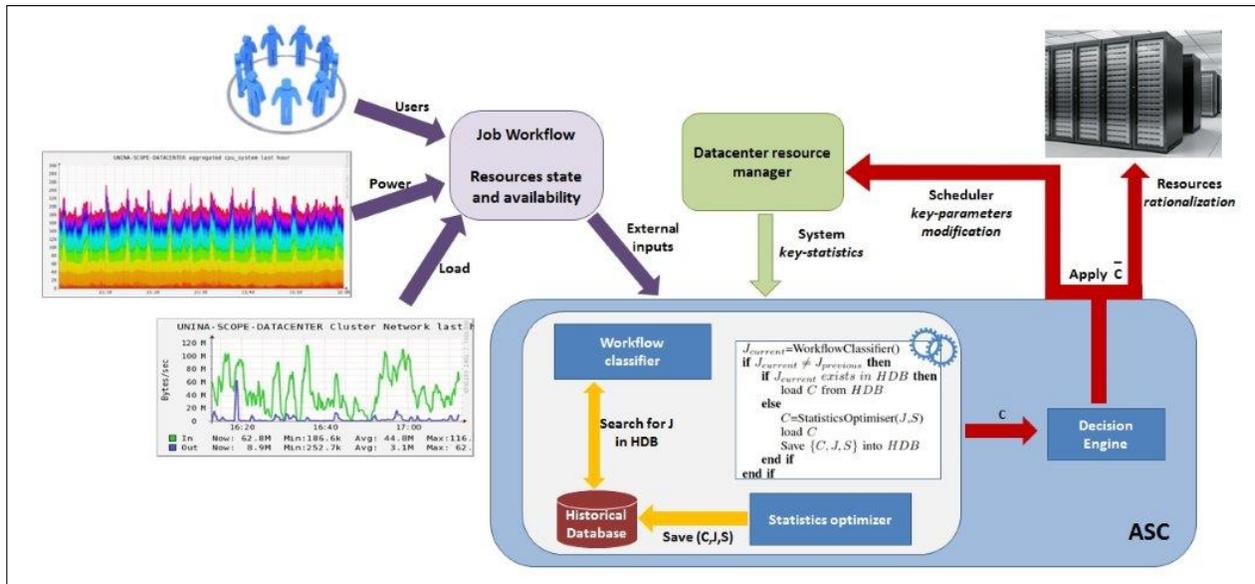


Figure 1. An “adaptive” approach in job scheduling policy by an external controller

From now on, C is the set of the resource manager configurations (each of them is identified by a set of value for *key-parameters*), J is the vector representing the work flow (partitioned into m classes of homogeneous jobs), and S contains the values of the considered metrics.

Looking at Figure 1, the ASC engine is organised into four core software modules, each one with a specific role in the operative model:

- the *Work flow Classifier* is asked to perform a job classification into m classes of homogeneous jobs starting from the set of n heterogeneous jobs;
- the *Historical Database* has the role to save for each meaningful classified work flow J , the set C of values for *key-parameters*, obtained from the *Statistics Optimiser* that optimise the considered *key-statistics* S ;
- the *Statistics Optimiser* provides the new set of values for C ;
- the *Decision engine*, implements a new asset (the \bar{C} configuration) for the datacenter by choosing physical or virtual resources from local or public Clouds also on the basis of other criteria (e.g., resource power consumption and geographical location).

The algorithm in Figure 2 describes the interaction among ASC core modules; particularly, $J_{current}$ refers to the new work flow, while $J_{previous}$ is the last classified work flow; *HDB* refers to the Historical Database.

B. Statistics Optimiser modelling

In our first ASC formalisation (see [2]), we obtained:

$$C = \sum_{j=1}^m \alpha_j(\mathbf{J}) \cdot F_j(\mathbf{S}) \quad (1)$$

```

1: ...
2: loop
3:   Jcurrent=WorkflowClassifier()
4:   if Jcurrent ≠ Jprevious then
5:     if Jcurrent exists in HDB then
6:       load C from HDB
7:     else
8:       C=StatisticsOptimiser(J,S)
9:       load C
10:      Save {C, J, S} into HDB
11:    end if
12:  end if
13:  C̄=DecisionEngine(C)
14:  apply(C̄)
15:  Sleep (some time)
16: end loop
17: ...
    
```

Figure 2. ASC engine algorithm

The symbol (\cdot) denotes a suitable operator, each function F_j computes *optimal* parameters values for job type j while α_j expresses the *weight* to be considered for job type j .

For each job class $j = 1, \dots, m$, we consider the following classic *key-statistics* (the first is representative of the whole computing system efficiency, while the second and third ones express user satisfaction):

$$\text{System effectiveness ratio } E^{(j)} = \frac{\sum_{i=1}^n p_i t_i}{PT};$$

$$\text{System Make span } M_k^{(j)} = \max_{i=1, \dots, n} (t_i + q_i);$$

$$\text{Queue waiting time average } Q^{(j)} = \frac{\sum_{i=1}^n q_i}{n}$$

where P is the total number of available processors, n is the total number of jobs for the work flow, Q is the total queue time for all the jobs of the work flow, T is the wall clock run time for all jobs completion, p_i , t_i and q_i are respectively the

number of processors requested, the execution time and the queue time for the job i .

Since:

$$M_k^{(j)} = \|\underline{t} + \underline{q}\|_\infty \geq \left| \|\underline{t}\|_\infty - \|\underline{q}\|_\infty \right|$$

and because of the equivalence of norms, we have:

$$M_k^{(j)} \geq \left| \|\underline{t}\|_1 - \|\underline{q}\|_1 \right| = \left| \sum_{i=1}^n t_i - \sum_{i=1}^n q_i \right|$$

Under the *realistic* assumption:

$$\sum_{i=1}^n q_i < \sum_{i=1}^n t_i \quad (2)$$

we have:

$$\sum_{i=1}^n t_i \leq M_k^{(j)} + nQ^{(j)}$$

so, for each job class $j = 1, \dots, m$, we want to solve the following:

Problem: To compute the set of the scheduler key-parameters $C_{Opt}^{(j)}$ such that

$$C_{Opt}^{(j)} = F_j \left(S_{Opt}^{(j)} \right) \quad (3)$$

where $S_{Opt}^{(j)} = \left(E_{Opt}^{(j)}, M_{k_{Opt}}^{(j)}, Q_{Opt}^{(j)} \right)$ and $E_{Opt}^{(j)}, M_{k_{Opt}}^{(j)}, Q_{Opt}^{(j)}$ are the solutions of the constrained "optimisation" problem:

$$\begin{cases} \max\{E^{(j)}\} & s.t. \\ T = \sum_{i=1}^n t_i \leq (M_k^{(j)} + nQ^{(j)}) \\ \sum_{i=1}^n q_i < \sum_{i=1}^n t_i \end{cases} \quad (4)$$

◇

We highlight that the problem (4) includes also the two limit cases:

best case: $q_i = 0 \forall i$, i.e., $Q^{(j)} = 0$ (there are no jobs in system queues)

$$\begin{aligned} M_k^{(j)} &= \max_{i=1, \dots, n} (t_i + q_i) = \max_{i=1, \dots, n} t_i \\ E^{(j)} &= \frac{\sum_{i=1}^n t_i p_i}{P \max_{i=1, \dots, n} t_i} \leq \frac{\sum_{i=1}^n t_{max} p_i}{P t_{max}} \approx 1 \end{aligned} \quad (5)$$

In the ideal case, there aren't new jobs in queue and all the processors in the system are allocated for jobs:

$$P = \sum_{i=1}^n p_i \quad (6)$$

so the system reaches the maximum efficiency ($E = 1$).

worst case: $q_i \rightarrow \infty \forall i$. This case is not possible because of the (2). Thus, being at most: $q_i \rightarrow t_i \forall i$ we have:

$$\begin{aligned} M_k^{(j)} &= \max_{i=1, \dots, n} (t_i + q_i) \rightarrow \max_{i=1, \dots, n} (2t_i) \\ E^{(j)} &\rightarrow \frac{\sum_{i=1}^n t_i p_i}{P \max_{i=1, \dots, n} (2t_i)} \leq \frac{\sum_{i=1}^n t_{max} p_i}{P 2t_{max}} \approx \frac{1}{2} \end{aligned} \quad (7)$$

so even if the (6) is verified, system efficiency E results under the 50%.

Both in (5) and in (7), t_{max} is the maximum wall clock time allowed on the queues.

The scheduler *key-parameters* value affects the queue time for different job types and therefore the values of the metrics considered above. On the basis of some information provided from the system:

- the fixed maximum execution time t_{max} for each job i submitted to a certain queue of the scheduling system;
- the total amount of requested processors P ;
- the user request of processors for each job i

we can estimate the metrics E , Q and M_k only if a probability distribution function for the queue time values q_i is known for all the jobs in the work flow.

Presently, we are working to define a stochastic model to forecast queue times also thanks to the already classified work flows.

IV. A CASE STUDY FOR THE ASC WORK FLOW CLASSIFIER

In this section, we report some experiences related to the use of data mining techniques to automate the work flow partitioning into homogeneous job classes as described in Section III. We use computational resources at the University of Naples Federico II, acquired in the context of PON *Sistema Cooperativo Per Elaborazioni scientifiche multidisciplinari* (S.Co.P.E.) Italian National project [16].

SCoPE resources are made available to national and international relevant distributed infrastructures (IGI and EGI) and, thus, used not only by the local users.

Due to the user communities heterogeneity, computational resources are used both for "traditional" GRID jobs and for HPC applications. From a heuristic analysis, we observe that SCoPE jobs are applications mostly sequential or with a low degree of parallelism (DOP) with a short duration. Just a subset, however large enough of SCoPE jobs, has a medium-high DOP and a more long duration.

The computational resources (about 2000 cores) are accessed by submitting jobs to the Resource Management System (based on Maui-Torque systems).

To automate the definition of the jobs classification \mathbf{J} , we consider a set of clustering algorithms implemented by the *Waikato Environment for Knowledge Analysis (WEKA)* package [17]. WEKA is a well-known suite of machine learning software which supports several typical data mining processes, particularly data preprocessing, clustering, classification, regression, visualisation, and feature selection.

ARFF format is used by WEKA to represent data sets that consist of independent, unordered instances. Each instance is characterised by its values on a fixed, predefined set of features or attributes.

Since we are interested on a jobs classification based on the duration and on the DOP of each job, we chose to represent each job with the two following attributes:

- `ntasks` the number of concurrent tasks of the job
- `tte` the job Total Time of Execution

The clustering processes are performed, by the three following algorithms and related WEKA command lines, on data related to the last 2 years (over 3 millions jobs):

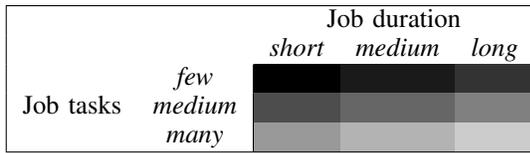


Figure 3. Jobs “ideal classification”

SKMeans: provides clustering with the k-means algorithm [14].

```
weka.clusterers.SimpleKMeans -N 9
-A "weka.core.EuclideanDistance"
-R first-last" -I 500 -S 10
```

XMean: provides k-means extended by an “Improve-Structure part” and automatically determines the number of clusters [18].

```
weka.clusterers.XMeans -I 1 -M 1000 -J 1000
-L 2 -H 9 -B 1.0 -C 0.5
-D "weka.core.EuclideanDistance"
-R first-last" -S 10
```

Festivity: provides the “farthest first traversal algorithm”, which works as a fast simple approximate “clusterer” modeled after simple k-means [19].

```
weka.clusterers.FarthestFirst -N 9 -S 1
```

For all the algorithms, we chose that the maximum possible number of clusters is $N_{cluster,max} = 9$ (see the red highlighted parameters in the command lines above) on the basis of the “ideal classification” represented in Figure 3.

The aim of the tests described above was to identify the most effective clustering algorithm in terms of:

- 1) computational cost
- 2) compliance to the results of our heuristic jobs classification

All the algorithms build at least four classes, where the most numerous one collects sequential jobs (or with low DOP jobs) consistently with our heuristic classification.

Anyway, the SKMeans based algorithm seems to be the most useful choice because, from our point of view, it responds better to the point 2 above: it builds more different classes, each of them with a significant number of elements (see Table I). The SKMeans based algorithm is, in terms of computational cost, more expensive than other ones but its cost can be considered acceptable compared to the frequency of scheduler reconfiguration (taking into account the mean duration of the jobs, we can assume a period of few days).

Figures 4 and 5 show, respectively, the real work flow characterisation and the results of the clustering process performed by WEKA on data related to the work flow execute on SCoPE resources during the month of March 2014. The clustering processes are performed by the SKMeans based algorithm which we choose as the “optimal” one for our purpose. During the considered month, different type of work flows and jobs are present on SCoPE infrastructure. A similar behaviour is

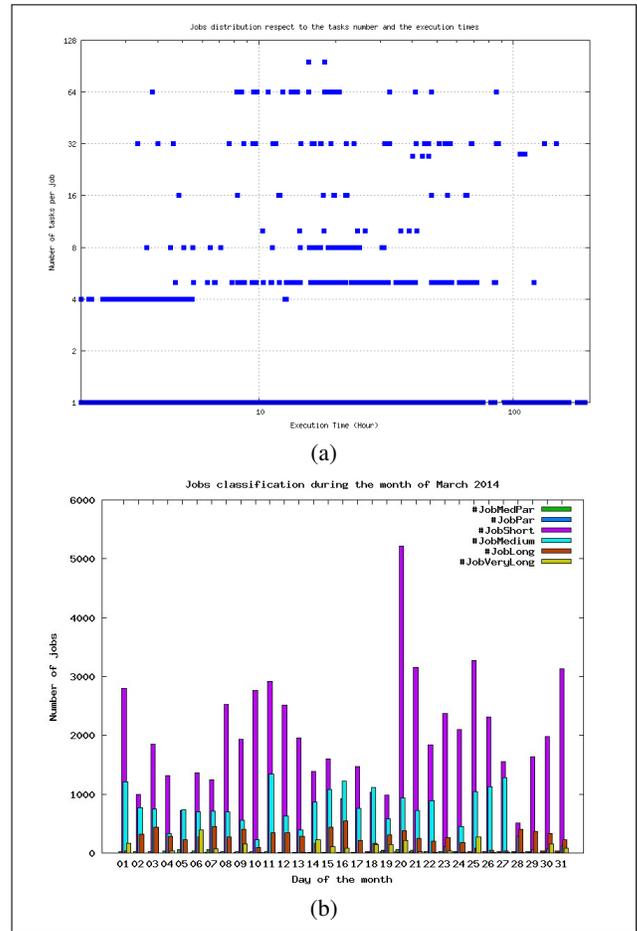


Figure 4. March 2014: real job work flow representation by two different views: the jobs distribution both respect to the number of tasks and to the execution time (a); the trend of the jobs number as a function of the month days (b).

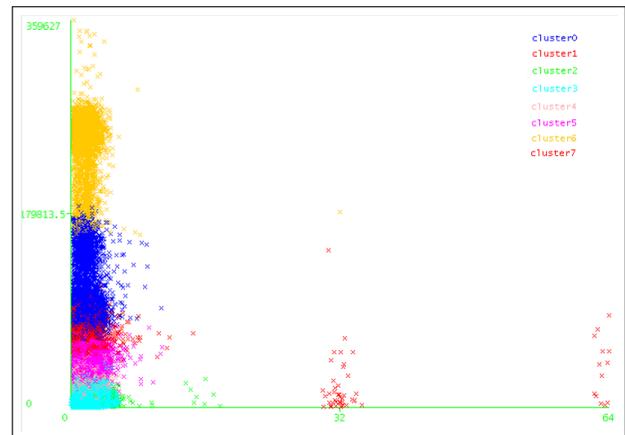


Figure 5. March 2014: clustering processes results on SCoPE jobs

TABLE I. CLUSTERING PROCESSES RESULTS ON ALL SCOPE JOBS

SKMeans (Execution times: 786.33 secs)		
Clusters	# of Elements	% of Elements
0	5804	0
1	11117	0
2	265923	7
3	52620	1
4	337429	9
5	417363	11
6	83707	2
7	2472225	68

XMeans (Execution times: 87.16 secs)		
Clusters	# of Elements	% of Elements
0	11333	0
1	142700	4
2	10937	0
3	3481218	95

FarthestFirst (Execution times: 12.75 secs)		
Clusters	# of Elements	% of Elements
0	3639639	100
1	3	0
2	30	0
3	106	0
4	192	0
5	97	0
6	12	0
7	4916	0
8	1193	0

present on other months of the year, confirming the need for adaptive approach to the scheduling problem.

V. CONCLUSION AND FUTURE WORK

In this document, we described the progresses made to devise ASC, which aims to gain a balanced, efficient and effective use of computing resources by heterogeneous users communities. Here, we gave details about ASC mathematical formalisation focusing on the chance to have a computable estimation for the involved *key-statistics*; moreover the use of data mining techniques allowed us to build a job clustering into homogeneous classes, starting from real heterogeneous jobs work flows.

Presently, we are working on:

- the dynamic work flow classification starting from job clustering results here obtained;
- the mathematical model with the aim to define, for different sets of *key-parameters* and already classified work flows, the probability distribution function for queue times.
- the realisation of a historical database of scheduler configurations, known work flows and related *key-statistics* values;
- the deployment of some features to enable ASC to take into account other parameters, e.g., computing nodes availability/dependability and metrics related to environmentally conscious computing services [20].

ACKNOWLEDGEMENT

This work is part of the activities of a multidisciplinary group (GTT), responsible for the SCoPE infrastructure management. It has been realised thanks to the use of the SCoPE computing infrastructure at the University of Naples, also in the framework of PON "Rete di Calcolo per Superb e le altre applicazioni" (ReCaS) project.

REFERENCES

- [1] A. Nazir and S.-A. Srensen, "Cost-benefit analysis of high performance computing infrastructures," in SOCA. IEEE, 2010, pp. 1–8.
- [2] G. Barone, V. Boccia, D. Bottalico, L. Carracciuolo, A. Doria, and G. Laccetti, "Modelling the behaviour of an adaptive scheduling controller," in Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on, July 2012, pp. 438–442.
- [3] Peter Mell and Tim Grance. The NIST Definition of Cloud Computing. [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> [retrieved: February, 2015]
- [4] Italian Grid Infrastructure. [Online]. Available: <http://www.italiangrid.it/> [retrieved: February, 2015]
- [5] Federated cloud, Building a grid of clouds. [Online]. Available: <https://www.egi.eu/infrastructure/cloud/> [retrieved: February, 2015]
- [6] T. Casavant and J. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," IEEE Transactions on Software Engineering, vol. 14, no. 2, 1988, pp. 141–154.
- [7] G. Serazzi and M. Calzarossa, "Adaptive Optimization of a System's Load," IEEE Trans. Software Eng., vol. 10, no. 6, 1984, pp. 837–845.
- [8] I. Foster. What's faster a supercomputer or EC2? [Online]. Available: <http://ianfoster.typepad.com/blog/2009/08/whats-faster-a-supercomputer-or-ec2.html> [retrieved: February, 2015]
- [9] H. Sun, Y. Cao, and W.-J. Hsu, "Fair and Efficient Online Adaptive Scheduling for Multiple Sets of Parallel Applications," in ICPADS. IEEE, 2011, pp. 64–71.
- [10] D. Feitelson, L. Rudolph, U. Schwiegelshohn, K. Sevcik, and P. Wong, "Theory and practice in parallel job scheduling," in Job Scheduling Strategies for Parallel Processing, ser. Lecture Notes in Computer Science, D. Feitelson and L. Rudolph, Eds. Springer Berlin Heidelberg, 1997, vol. 1291, pp. 1–34.
- [11] K. Gkoutioudi and H. D. Karatza, "Task cluster scheduling in a grid system," Simulation Modelling Practice and Theory, vol. 18, no. 9, 2010, pp. 1242–1252.
- [12] J. Aguilar and E. Gelenbe, "Task Assignment and Transaction Clustering Heuristics for Distributed Systems," Information Sciences, vol. 97, 1997, pp. 1–2.
- [13] Z. C. Papazachos and H. D. Karatza, "The impact of task service time variability on gang scheduling performance in a two-cluster system," Simulation Modelling Practice and Theory, vol. 17, 2009, pp. 1276–1289.
- [14] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.
- [15] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," SIGKDD Explor. Newsl., vol. 4, no. 1, Jun. 2002, pp. 65–75.
- [16] Merola L. on behalf of S.Co.P.E. Project, "The S.Co.P.E. Project," in Proceedings of the Final Workshop of the Grid Projects of the Italian National Operational Programme 2000-2006 Call 1575, Edited by Consorzio COMETA, 2008, pp. 18–35.
- [17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explor. Newsl., vol. 11, no. 1, Nov. 2009, pp. 10–18.
- [18] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in Proceedings of the Seventeenth International Conference on Machine Learning, ser. ICML '00. Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.
- [19] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," Mathematics of Operations Research, vol. 10, 1985, pp. 180–184.
- [20] G. Barone, R. Bifulco, V. Boccia, D. Bottalico, and L. Carracciuolo, "Toward a flexible, environmentally conscious, on demand high performance computing service," in Data Compression, Communications and Processing (CCP), 2011 First International Conference on, June 2011, pp. 136–138.

Visualization Tool for Development of Communication Algorithms and a Case Study Using the K Computer

Syunji Yazaki

The University of Electro-Communications
Tokyo, Japan

Email: yazaki.syunji@uec.ac.jp

Ryohei Suzuki

Cresco Ltd.
Tokyo, Japan

Email: r-suzuki@cresco.co.jp

Fumiyoshi Shoji

RIKEN Advanced Institute for Computational Science
Hyogo, Japan

Email: shoji@riken.jp

Kenichi Miura

Fujitsu Limited
Kanagawa, Japan

Email: k.miura@jp.fujitsu.com

Hiroaki Ishihata

Tokyo University of Technology
Tokyo, Japan

Email: ishihata@stf.teu.ac.jp

Abstract—In this paper, we introduce our visualization tool, the Communication Log Viewer (CLV), that assists the development of collective communication algorithms. We also present visualization results as a case study. CLV visualizes information regarding node events and network statistics in linked multiple views. CLV also has a function for analyzing the results obtained from network simulators and actual machines in the same framework, which is useful when developers repeatedly test their algorithms on a simulator and an actual system. For a case study, we visually evaluated two all-to-all algorithms on the full system of the K computer that has 82,944 nodes. As a result, we confirmed that an optimized all-to-all algorithm implemented for the K computer performed better than an all-to-all implemented in Open MPI. We also confirmed that the barrier operation used in the K computer's Message Passing Interface (MPI) functions keep link utilization high. However, there is also a trade-off between the number of barriers and link utilization.

Keywords—Visualization; Mesh/Torus network; All-to-all

I. INTRODUCTION

Parallel application programmers frequently utilize collective communications implemented in Message Passing Interface (MPI) libraries to design applications. Collective communications usually produce a large number of communications, especially on the peta-scale parallel systems that consist of tens of thousands of nodes. In the applications running on such large systems, communication takes longer than computation.

Optimizing communication algorithms is an important means of maximizing the performance of parallel applications [1]. Many parallel systems listed in the Top500 [2], such as the Cray XK7 [3], Blue Gene/Q [4], and K computer [5], employ mesh/torus topology. Mesh/torus topology generally provides better scalability with respect to hardware cost. However, the bisection bandwidth is relatively narrow compared to that of other topologies, such as Fatree [6] and Dragonfly [7].

Visualization tools that abstract and visualize communication behavior are necessary tools for optimizing communication algorithms [8]. Developers repeatedly test communication algorithms under development on network simulators and actual systems to find potential areas for optimization. The test results are usually obtained as huge logfiles and extensive numerical data. Looking at the logfiles and numerical data alone, it is difficult to determine potential areas for optimization.

We briefly presented our visualization tool, the Communi-

cation Log Viewer (CLV), that supports the design of collective communication algorithms in [9]. Our tool has a function that visualizes both the results obtained from a network simulator and an actual system in the same framework. Our tool also visualizes both events that occur in the node and statistics regarding traffic in the network simultaneously with linked multiple views. This enables the user to distinguish quickly which events in the nodes correspond to which congested network links.

In this paper, we describe the details of CLV that can visualize both the node events and network statistics. We also show a visual evaluation of all-to-all on a full system of the K computer that has 82,944 nodes. In the rest of this paper, Section II explains the workflow for developing communication algorithms and Section III presents related work. Section IV then describes the features of CLV. Section V provides a case study, and Section VI concludes the paper.

II. WORKFLOW AND REQUIREMENTS FOR VISUALIZATION

A workflow to develop collective communication algorithms involves the following steps.

- 1) Designing an algorithm that takes into account the network architecture of the target system
- 2) Testing the algorithm on a network simulator and generating simulation logfiles
- 3) Analyzing and evaluating the behavior and efficiency of the algorithm based on the information in the logfiles
- 4) Implementing the algorithm on the target system, if the algorithm has achieved the expected performance in the simulation
- 5) Evaluating the algorithm based on logfiles and numerical data obtained from the performance counters of the target system

Considering this workflow, the following functions are needed in visualization tools:

- R1 Visualizing the simultaneous network statistics and node events with concise association
- R2 Mapping the information to the actual network structure of the target system
- R3 Showing the information in multiple linked views
- R4 Displaying concise information by filtering
- R5 Supporting the outputs obtained both from simulators and actual systems in the same framework

Visualizing both the network statistics and node events in the same tool allows the developer to easily find bottleneck links and the events that cause them. Additionally, the developer also can intuitively understand which part of a network is heavily used by observing the topologically mapped information. The tool should provide multiple views with multiple levels of abstraction because needed information is sometimes lost at different levels of abstraction. In addition, information that is not the current focus of the investigation must be filtered out by the developer. As discussed above, developers evaluate their algorithms on a simulator and target system. Considering this, the tools should analyze both outputs obtained from simulators and actual systems within a single framework.

III. RELATED WORK

Bhatele et al. visualized communications that occurred in an Adaptive Mesh Refinement (AMR) application [10]. Existing visualization tools, such as Jumpshot [11], ParaProf [12], and Vampir [13], visualize a profiler’s outputs. Other well-known tools such as OpenSpeedShop [14] and TAU [15] provide an integrated environment for performance analysis. These tools summarize outputs obtained by the profiles in graphs, tables, and figures. They are useful for analyzing events that occur in the nodes. However, these tools basically do not support network statistics because profilers cannot be designed to acquire the network statistics.

Minkenber and Rodriguez proposed a simulation environment to support the development of high performance computing systems [16]. An MPI task simulator works with a network simulator in this simulation environment to emulate parallel applications running with specific topology and hardware construction. This environment uses Paraver [17] to visualize the communication.

SimCon [18] was developed to find appropriate overlay networks for running parallel applications. This simulator displays networks based on physical distances and links between pairs of communicating nodes. Users can understand the communication situation from the simulation results. Gamblin et al. [19] also developed a tool to evaluate parallel applications to optimize node mapping. This tool presents a topological view of the network and places the information on this view. By observing this view, users can intuitively check the communication situation. However, these tools also require external tools to obtain network statistics.

Landge et al. [20] developed a visualization tool to analyze packet traffic on the torus network. This tool focuses especially on recent Blue Gene systems. It provides two linked views to show an overview of the packet traffic. One view is a 2D projection, mainly used to show brief trends and patterns of the traffic. The other view is a 3D topological view that maps traffic patterns to the physical structure of a target network. This tool allows application developers to understand link utilization and find bottleneck links intuitively. However, application developers still need to combine it with another tool to inspect the cause of the bottleneck.

IV. CLV

A. System summary

CLV is designed to implement two requirements, R1 and R5, into a single tool. Existing visualization tools introduced in Section III have some features that implement R2, R3, and

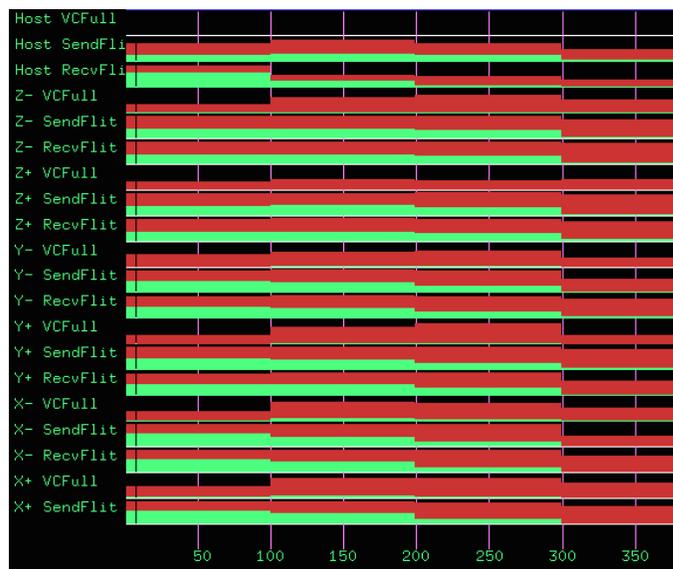


Figure 1. Example of the time series view. Maximum and average values are represented by red and green bars, respectively.

R4. R1 and R5 can also be realized by combining multiple tools. However, there is no integrated tool that implements R1 and R5. CLV also has basic functions that are implemented in existing visualization tools. In addition, CLV can read and analyze logfiles obtained both from simulators and actual systems in the same framework. This feature is useful when developers need to test their algorithms on a simulator and an actual system repeatedly. As of now, Booksim [21] and Message Flow Simulator (MFS) [22] can be used as the simulators, and the K computer using the Tofu Performance Analysis (Tofu PA) [23], is targeted for the actual system. The developer can use existing simulators or performance measurement tools for traffic data acquisition. CLV provides an utility to convert the data format to feed the data into CLV.

CLV mainly provides two views: time series and topological. The time series view summarizes network statistics in the time series. This view is also used to select a particular time on which to focus. All views in CLV are linked from the time series view. When a user selects a time in the time series view, all other views show the data at that time. The topological view maps the events that occurred in the nodes and network statistics to the physical structure of the network in 3D space. As of now, CLV can visualize the messages sent as the events. This view shows link utilization as well as the duration of the communication delay. Users can filter the information by selecting nodes or links in this view. The users also can understand where the messages sent events occurred in the network from this view.

CLV was implemented in C++ with OpenGL library. Thus, the CLV can be compiled on any OS that has OpenGL implementation. We also used the OpenGL Utility Toolkit (GLUT) [24] to construct the user interface.

B. Time series view

Figure 1 shows an example of the time series view. This screenshot shows the link utilization for sending (SendFlit) and receiving (RecvFlit) for all links in each dimension. It also shows the communication delay (VCFull). The horizontal

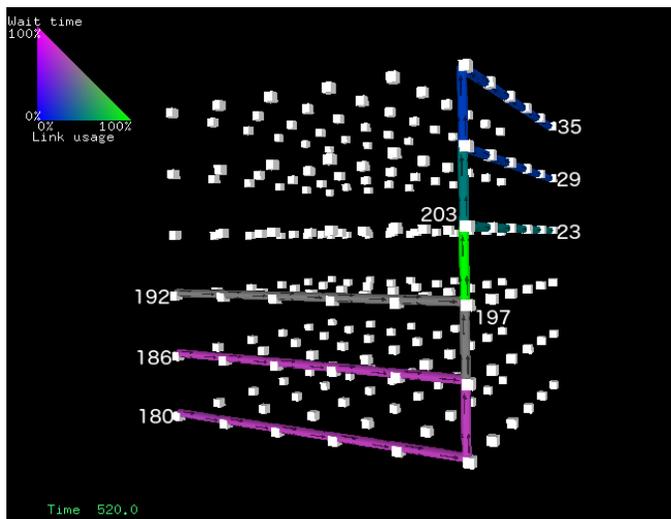


Figure 2. Examples of the topological view showing link utilization and communication delay

line represents the elapsed times. The time unit is an interval time of data specified in the logfiles. The link utilization and communication delay are defined as the fraction of bandwidth and number of wait cycles in each time unit, respectively. Dimensions “X,” “Y,” and “Z” correspond to the first, second, and third dimensions of the network. “Host” refers to a link connected from a node to a router (switch) in the network. The “+” and “-” symbols indicate the directions of the links.

The graphs in the screenshot indicate the maximum and average values by the red and green bars, respectively. If the average value (green) is close to the maximum value (red), most messages on all the links in those directions and dimensions have transferred efficiently. In contrast, if these values are far apart, a small number of links are heavily used, while most of the remaining links are not utilized.

C. Topological view

In the topological view, information is mapped to the physical structure of a network topology in 3D space. Figure 2 shows an example of the topological view. The triangle in the top left corner of the figure is a legend that specifies link utilization and communication delay by color.

The situation in Figure 2 is such that the three source nodes at the lower front left nodes (180, 186, and 192) are communicating with three other destination nodes at the upper back right (23, 29, and 35). We added the node numbers to the screenshot for this explanation.

Small arrows are printed on the links. These are showing a direction of the messages sent. From the direction of the arrows, the users can understand that the events of the messages sent occurred in node 180, 186, and 192.

The colors represent link utilization and communication delay. The green link between nodes 197 and 203 indicates that this link is fully used (100%). Furthermore, we can see that the links around this green link are colored dark green and gray. The user can determine that these links have lower link utilization from the legend. There are purple and gray links in the lower half of the view. This indicates that the packets on the purple links wait for a long time, while packets on the gray

links only wait for a short time. From this view, the user can guess that the packets on the purple links are blocked while the packets on the gray links are transmitted through the green links. In the same way, users can guess the cause of low link utilization from this view.

V. CASE STUDY

A. Comparing all-to-all algorithms on a simulator

We first compared two all-to-all algorithms using CLV on simulation results. We simulated all-to-all communications on a $10 \times 10 \times 10$ 3D mesh network using Booksim. In this simulation, each node sends one message to 999 other nodes. One message consists of 200 flits, and one flit is sent per cycle. Dimension order routing is used as the routing algorithm. Node numbers (ranks) are assigned in the order of x , y , and z dimensions.

We chose to compare two all-to-all algorithms using a simple spread algorithm and an algorithm optimized for torus. We refer to the simple spread and torus optimized algorithms as A2A and A2AT, respectively, in this paper. A2A is used in many MPI libraries such as MPICH [25], MVAPICH [26], and Open MPI [27]. A destination node number in A2A is calculated by $(src + i) \bmod N$ ($i = 1, 2, \dots, N - 1$). Here, src and N represent a source node number and the total number of nodes in the network, respectively. A2AT is an optimum topology-aware algorithm for mesh/torus networks that we previously proposed [28]. We have shown that A2AT performs better than a modified version of A2A. Here, we investigate the communication efficiency with respect to performance using the CLV visualization result.

Figure 3 presents the visualization results of both algorithms in the time series view. The figures show the first part of each simulation result. Link utilization in the y and z dimensions in Figure 3(a) decreases several times, as indicated by the rectangles. This indicates that A2A does not utilize links in the y and z dimensions at this time. In contrast, there are no large black spaces in Figure 3(b). This indicates that A2AT utilizes all links at any time. We confirmed from these results that A2AT provides better performance by utilizing the links in y and z dimensions to avoid using the bottleneck links.

B. Comparing all-to-all algorithms on the K computer

1) Preparation: The K computer implements a tuned all-to-all algorithm in its MPI library [29]. We refer to this MPI library as the Tofu MPI in this paper because the K computer employs a 6-dimensional mesh/torus network called the Tofu interconnect. In this algorithm, the order in which the messages are sent is modified while considering the routing algorithm of the K computer. The barrier operation is also used to make the messages sent in this algorithm uniform. We evaluated this algorithm on a full system of the K computer at the RIKEN Advanced Institute of Computer Science in Japan [30].

The fourth, fifth, and sixth dimensions are labeled a , b , and c , respectively, in the Tofu interconnect. The K computer has $82,944 (= 24 \times 18 \times 16 \times 2 \times 3 \times 2)$ computation nodes, excluding I/O nodes. The physical construction of the K computer was $24 \times 18 \times 17 \times 2 \times 3 \times 2$. However, nodes located on $z = 0$ are reserved for I/O nodes. Therefore, the full size of the system that could be used was $24 \times 18 \times 16 \times 2 \times 3 \times 2$ because no I/O nodes joined into the MPI communicator. Note that the y , a , and c dimensions are mesh networks while the others are

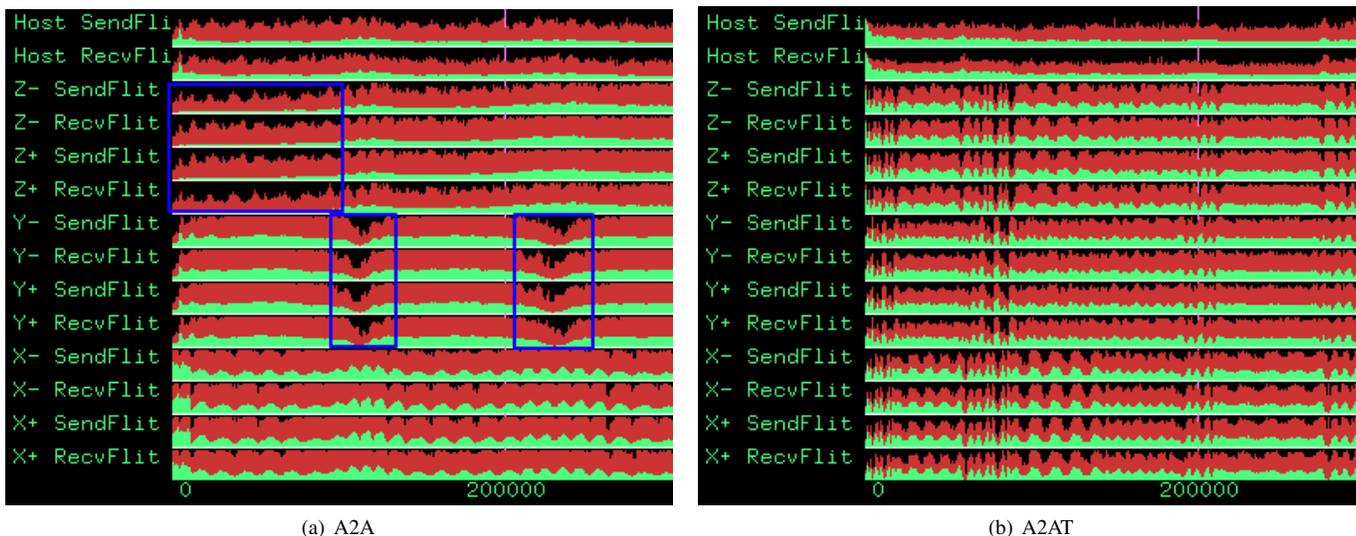


Figure 3. Comparison of simulation results for A2A and A2AT on a $10 \times 10 \times 10$ 3D mesh network in the time series view.

TABLE I. SUMMARY OF NETWORK STATISTICS OBTAINED WITH THE TOFU PA DURING ALL-TO-ALL COMMUNICATION WITH 32,768 BYTES of DATA

	Run time	Samples	Period	Raw logfile size
Tofu MPI	2.77 s	= 277	$\times 10$ ms	7.9 GB
Open MPI	22.94 s	= 2,294	$\times 10$ ms	59 GB

torus networks.

We ran the tuned all-to-all algorithms implemented in Tofu MPI and an existing all-to-all algorithm implemented in Open MPI with 32,768 bytes of data. We also obtained logfiles using the Tofu PA. TABLE I summarizes the logfiles that we obtained. The total amount of the messages to be sent was 205.0 TB.

Here, we estimate the ideal communication time of an all-to-all communication on the K computer. The lower bound of all-to-all communication time (L) for a mesh/torus network can be generalized as

$$L = (\alpha \lfloor \frac{k_b}{2} \rfloor \lceil \frac{k_b}{2} \rceil (\prod_{i=0}^{n-1} k_i) m) / B. \quad (k_i \neq k_b) \quad (1)$$

Here, k_b represents the sizes of the dimensions that have bottleneck links. The links in the longest dimension are the bottlenecks. Variables n and k_i indicate the number of dimensions and size of the i -th dimension, respectively. Note that the dimension corresponding to k_b will be skipped. Parameters m and B represent message size and link bandwidth, respectively. Parameter α is set to 1/2 or 1 depending on whether the bottleneck dimension has wrap-around links (torus) or not (mesh).

We need to determine a dimension size k_b that includes bottleneck links to calculate an ideal communication time from (1). Links in the longest dimension are basically bottleneck links. Yet, the x dimension is longer than the y dimension in the K computer network. However, the bandwidth of the x dimension is twice that of the y dimension because the x dimension is a torus network. Therefore, the bottleneck links of the K computer are in the y dimension. The effective

bandwidth of a single link in the K computer was 4.76 GB/s. Thus, the optimum communication time for an all-to-all communication in the K computer can be calculated to be 2.57 s from (1). From the table, we can find that the Tofu MPI spent 2.77 s on an all-to-all communication. This means that the tuned all-to-all algorithm achieved 1.08 times the optimum communication time.

2) *Visualization of all-to-all algorithms:* We then visualized an all-to-all communication using the logfiles obtained by the Tofu PA. The logfiles included various network statistics. However, the logfiles do not include information corresponding to the node events such as amount of traffic from the nodes to the nearest routers. Thus, we only visualized the network statistics at this time. We visualized the fraction of bandwidth and communication delay based on the number of sent packets and the number of wait cycles needed to inject packets.

Figure 4 shows screenshots of the visualization results in the time series view. The link utilization and message delay are represented by “Sbyte” and “VC” in Figure 4. Both average and maximum link utilization (Sbyte) of the Tofu MPI that is shown in Figure 4(a), were higher than those of Open MPI that is shown in Figure 4(b). By simply looking at these visualization results, we can intuitively understand that the Tofu MPI utilized more links than Open MPI. Link utilizations in the a , b , and c dimensions are relatively low compared to other dimensions in both figures. Dimensions a , b , and c construct a small $2 \times 3 \times 2$ sub-network, hence links in this small sub-network are not heavily used.

We also found that link utilization in the x , y , and z dimensions drops twice in Figure 4(a). The all-to-all algorithm implemented in the Tofu MPI uses the barrier operations at these two points to send messages uniformly. Right before the barrier points, it can be seen that the average link utilization indicated in green in the x dimension is declining. It then recovers after the barrier points. However, there are also black gaps around the barrier points. This indicates that the communications are being blocked during these cycles. This could be the barrier penalty. From this observation, we concluded that the barrier operations work well to keep the average link

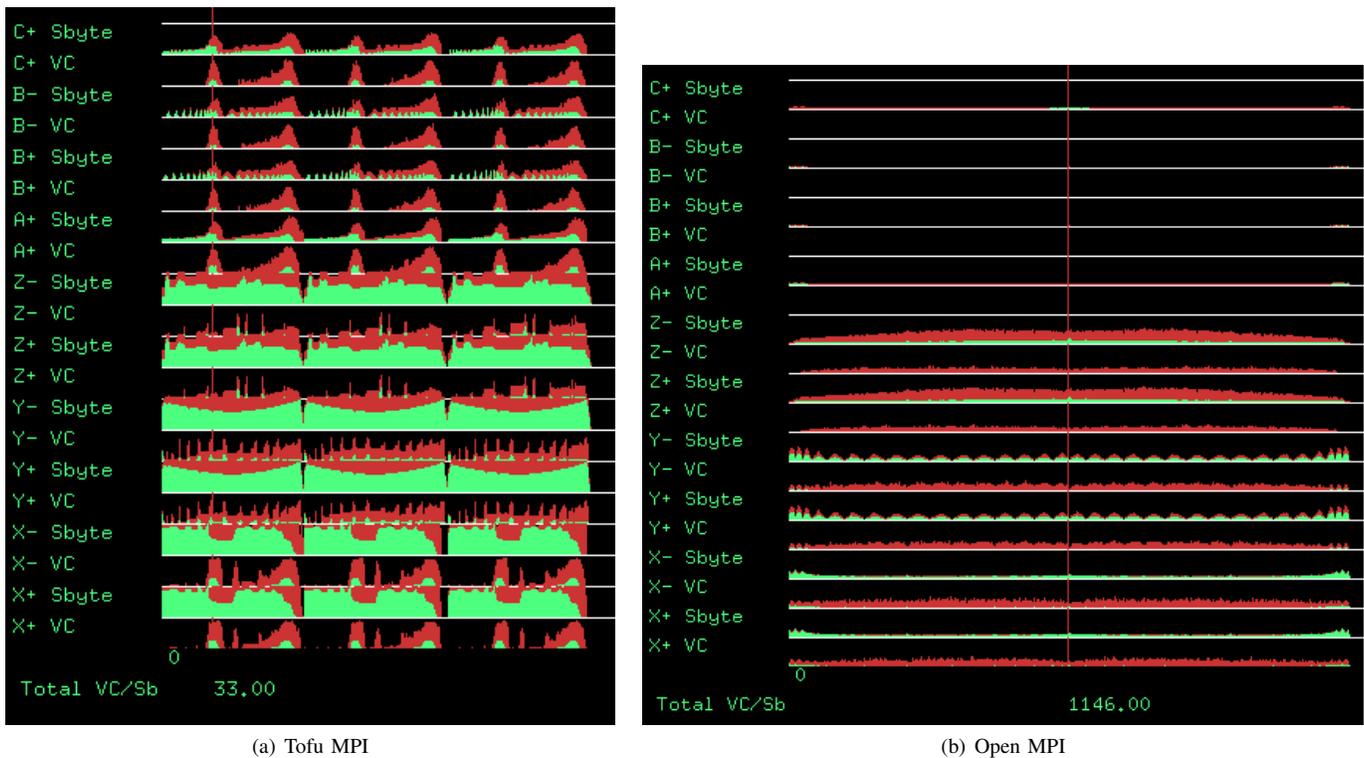


Figure 4. Comparison of all-to-all algorithms on full system of the K computer (82,944 nodes) in the time series view.

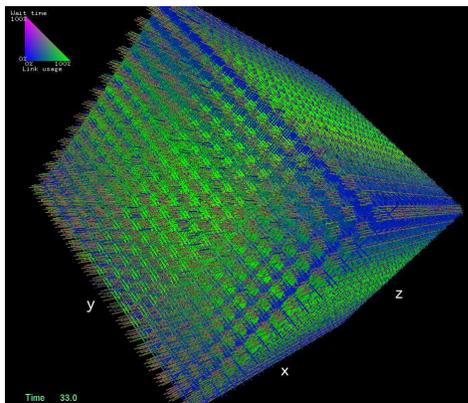


Figure 5. All-to-all communication using the Tofu MPI on full system of the K computer (82,944 nodes) in the topological view at 0.33 s

utilization high. However, there is a trade-off between the number of barriers and average link utilization.

Figure 5 shows a screenshot of the topological view of the Tofu MPI at 0.33 s. The red vertical line in Figure 4(a) indicates this time. We can easily focus on this time by selecting it in the time series view. Links near the middle of the y and z dimensions are green, as can be seen in Figure 5. This indicates that these links were fully utilized. However, links near the edge of the network that were not used as much are colored blue. This is because the y and z dimensions were a mesh network, which has no wrap-around links. Thus, links near the corner of the network were not used much more than the links in the central area of the network.

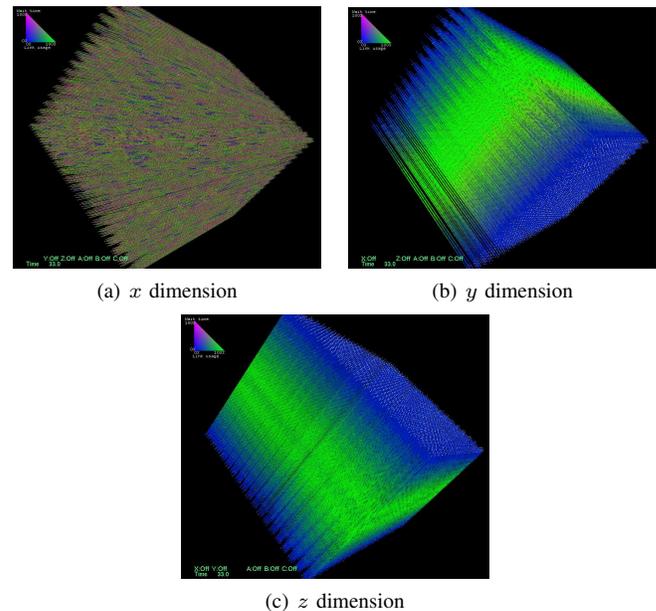


Figure 6. Views of Figure 5 filtered by dimension.

We next filtered this topological view by each dimension for further analysis. The visualization results are shown in Figure 6. In Figure 6(a), we can see many brown links in the x dimension. In contrast, there are no brown links, in the y and z dimensions in Figures 6(b) and (c). This indicates that many packets were blocked only in the x dimension. The K computer

employs routing that sends messages to the x dimension first. Thus, in all-to-all communication, many packets are injected into links in x dimension within a short time slot. This leads to congestion in the x dimension. However, the delay is not critical, as many links are brown. We suppose that messages are sent at a constant pace by utilizing the barrier operations explained above.

VI. CONCLUSIONS

We introduced CLV, a visualization tool to assist the development of collective communication algorithms. CLV provides time series and topological views that visualized information about node events and network statistics.

We also presented a case study to demonstrate the effectiveness of CLV. We compared the performances of simple (A2A) and topology-aware (A2AT) designs of all-to-all algorithms both in a simulation and on an actual system. We ran two all-to-all communication algorithms implemented in the Tofu MPI and Open MPI to compare their performances. The algorithm implemented in the Tofu MPI was optimized for the K computer. We found that the Tofu MPI achieved 1.08 times the optimum all-to-all communication time on the full system of the K computer that includes 82,944 nodes. We confirmed that the barrier operation used in the Tofu MPI effectively keeps link utilization high. We also pointed out that there is a trade-off between the number of barriers and link utilization for further optimization.

As of now, CLV can only visualize the messages sent as the node events. For the future work, other node events that affect network utilization should be visualized to help optimization of communication algorithms.

VII. ACKNOWLEDGMENTS

We would like to thank Mr. Yuji Oinaga and Mr. Naoki Shinjo of Fujitsu Ltd. for supporting this work. A part of this work was supported by JSPS KAKENHI Grant Number 2533016.

REFERENCES

- [1] Y. Gong, B. He, and J. Zhong, "Network performance aware MPI collective communication operations in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, no. 1, p. 1, 2013.
- [2] Top500 lists. [Online]. Available from: <http://www.top500.org/>, Feb., 2015.
- [3] B. Bland, "Titan - early experience with the titan system at oak ridge national laboratory," in *2012 SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC)*, pp. 2189–2211, Nov. 2012.
- [4] S. Kumar, A. Mamidala, P. Heidelberger, D. Chen, and D. Faraj, "Optimization of mpi collective operations on the ibm Blue Gene/Q supercomputer," *Intl. Journal of High Performance Computing Applications*, vol. 28, no. 4, pp. 450–464, 2014.
- [5] Y. Ajima, T. Inoue, S. Hiramoto, T. Shimizu, and Y. Takagi, "The tofu interconnect," *IEEE Micro*, vol. 32, no. 1, pp. 21–31, 2012.
- [6] C. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *Computers, IEEE Transactions on*, vol. C-34, no. 10, pp. 892–901, Oct. 1985.
- [7] J. Kim, W. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *35th Intl. Symposium on Computer Architecture*, 2008. ISCA '08., pp. 77–88, Jun. 2008.
- [8] K. E. Isaacs, A. Gimnez, I. Jusufi, T. Gamblin, A. Bhatel, M. Schulz et al., "State of the art of performance visualization," in *Proc. of Eurographics Conf. on Visualization 2014*, Jun. 2014, ILNL-CONF-652873.
- [9] R. Suzuki and H. Ishihata, "Visualization tool for network topology aware communication algorithm development," in *Supercomputing 2012 Research Poster*, Nov. 2012.
- [10] A. Bhatel, T. Gamblin, K. E. Isaacs, B. T. N. Gunney, M. Schulz, P.-T. Bremer et al., "Novel views of performance data to analyze large-scale adaptive applications," in *Supercomputing 2012*, pp. 31:1–31:11. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012.
- [11] A. Chan, W. Gropp, and E. Lusk, "An efficient format for nearly constant-time access to arbitrary time intervals in large trace files," *Scientific Programming*, vol. 16, pp. 155–165, Apr. 2008.
- [12] R. Bell, A. Malony, and S. Shende, "ParaProf: A portable, extensible, and scalable tool for parallel performance profile analysis," in *Proc. of Intl. Euro-Par Conf.*, pp. 17–26, Aug. 2003.
- [13] A. Knupfer, H. Brunst, J. Doleschal, M. Jurenz, M. Lieber, H. Mickler et al., *The Vampir Performance Analysis Tool-Set*, M. Resch, R. Keller, V. Himmler, B. Krammer, and A. Schulz, Eds. Springer Berlin Heidelberg, 2008.
- [14] M. Schulz, J. Galarowicz, D. Maghrak, W. Hachfeld, D. Montoya, and S. ord, "Open|speedshop: An open source infrastructure for parallel performance analysis," *Scientific Programming*, vol. 16, no. 2-3, pp. 105–121, 2008.
- [15] S. Shende and A. D. Malony, "The TAU parallel performance system," in *Intl. Journal of High Performance Computing Applications*, vol. 20, no. 2, pp. 305–312, 2006.
- [16] C. Minkenber and G. Rodriguez, "Trace-driven co-simulation of high-performance computing systems using omnet++," in *Proc. of the 2nd Intl. Conf. on Simulation Tools and Techniques*, pp. 65:1–65:8, 2009.
- [17] G. Jost, H. Jin, J. Labarta, J. Gimenez, and J. Caubet, "Performance analysis of multilevel parallel applications on shared memory architectures," in *Proc. of Intl. Parallel and Distributed Processing Symposium*, pp. 22–26, Apr. 2003.
- [18] M. Esch, J. Botev, H. Schloss, A. Hohfeld, I. Scholtes, and B. Zech, "SimCon - a simulation and visualization environment for overlay networks and large-scale applications," in *Proc. of the 1st Intl. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems*, pp. 1–9, 2008.
- [19] T. Gamblin, M. Schulz, T. Bremer, J. Levine, and V. Pascucci, "Intuitive performance visualization techniques for topological analysis on capability machines," in *IPSSJ SIG Technical Reports*, vol. 2011, no. 51, pp. 1–8, Jul. 2011.
- [20] A. Landge, J. Levine, A. Bhatel, K. E. Isaacs, T. Gamblin, M. Schulz et al., "Visualizing network traffic to understand the performance of massively parallel simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2467–2476, Dec. 2012.
- [21] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. M. Kaufmann Publishers Inc.: San Francisco, 2003.
- [22] S. Yazaki and H. Ishihata, "Message flow simulator for evaluating communication algorithms," in *Proc. of The Ninth IASTED Intl. Conf. on Parallel and Distributed Computing and Networks 2010*, pp. 291–298, Feb. 2010.
- [23] K. Ida, Y. Ohno, and S. Inoue, "Performance profiling and debugging on the k computer," *FUJITSU*, vol. 63, no. 3, pp. 287–331, May 2012 (in Japanese).
- [24] GLUT. [Online]. Available from: <https://www.opengl.org/resources/libraries/glut/>, Feb., 2015.
- [25] MPICH. [Online]. Available from: <http://www.mpich.org/>, Feb., 2015.
- [26] MVAPICH. [Online]. Available from: <http://mvapich.cse.ohio-state.edu/>, Feb., 2015.
- [27] Open MPI. [Online]. Available from: <http://www.open-mpi.org/>, Feb., 2015.
- [28] S. Yazaki, H. Takaue, Y. Ajima, T. Shimizu, and H. Ishihata, "An efficient all-to-all communication algorithm for mesh/torus networks," in *Proc. of The 10th IEEE Intl. Symposium on Parallel and Distributed Processing with Application*, pp. 277–284, 2012.
- [29] Fujitsu, "Information processing system and method of controlling the same across-reference to related application," US Patent no. 14/087043, 2012.
- [30] RIKEN Advanced Institute for Computational Science. [Online]. Available from: <http://www.aics.riken.jp/en/>, Feb., 2015.

Creating, Protecting and Sharing of Semantically-Enabled User-Orientated Electronic Laboratory Notebook in a Service Environment

Tahir Farooq

School of Computing, University of Leeds
Leeds, United Kingdom
tahir_farooq@hotmail.com

Zulkifly Mohd Zaki

Faculty of Science and Technology, Universiti Sains Islam
Malaysia, Nilai, Malaysia
zulkifly@usim.edu.my

Abstract— We discuss in-addition to our previous paper, the creating, protecting and sharing of semantically-enabled, user-orientated Electronic Laboratory Notebook (ELN) in a service environment. The advantages are that ELN service manages the end-to-end provenance life cycle on behalf of the community modellers regardless of local ELN system which involves individual ELN support, ELN version control and hardware/software issues. The performance of the solution presented showed a reasonable degree of end-user acceptance of the proposed approach.

Keywords-*electronic laboratory notebook; collaborative software; sharing; fine-grained access; online; service environment.*

I. INTRODUCTION

In our previous research [1], the Dynamic Role-Based Access Control (DRBAC) mechanism was discussed to protect and share the ELNs in a distributed, co-laboratory research environment. The ELN-DRBAC mechanism was used to allow community modellers to transfer locally created ELNs into a central repository where they can share their personal ELNs with other modellers in the community either as a whole ELN object or its elements (provenance trails) at fine-grained level. An evaluation was undertaken with the members of atmospheric chemistry community working on the EUROCHAMP-2 project [2]. The scientific goal of this community is to better understand the chemical processes (reactions) taking place in the lower atmosphere through the use of atmospheric simulation chambers. These processes can have significant impacts on both air quality and climate change.

In this paper, we extend our previous work to address the issues when creating ELNs in a local environment. The Electronic Laboratory Notebook - Creating, Protecting and Sharing (ELN-CPS) service provides an environment for the modellers to create, protect and share ELNs online without the use of local ELN system. This improves the consistency of conducting modelling experiments across the community. The local ELN system [3] was specifically designed to capture and retrieve high quality metadata concerning the modelling process together with modeller's reasoning. However, the analysis of the local user-orientated ELN system showed the following:

- i) It was difficult to maintain the different versions of ELN modelling metadata; and
- ii) The complexity of settings and configurations of the local ELN system requires technical assistance

before modeller can start using it [3]. This may hinder the modeller from using the ELN.

The ELN-CPS service environment encapsulates the functionality of the local ELN system and ELN protecting and sharing mechanism. It consists of two main parts:

- i) ELN protecting and sharing control which is based on dynamic role based access control; and
- ii) ELN simulation service and associated sub services (i.e., ELN retrieval and IPNav services).

In this paper, the following contributions are made:

- i) The existing user-orientated ELN system could be used across multiple platforms in a service environment. The ELN-CPS service is made online for creating, protecting and sharing of modelling metadata. It also addresses the need to set up the ELN system.
- ii) The management of different versions of modelling metadata to ensure modellers can retrieve the relevant and correct metadata across different versions of simulations.
- iii) The central archive mechanism of ELNs, particularly for group simulations are defined to avoid any disaster situation at local level.

This paper is structured as follows: Section 2 highlights the background of this research while Section 3 details the requirements for the ELN-CPS service. The ELN-CPS service architecture is discussed in Section 4 which is then used to elicit feedback from members of the atmospheric community in Section 5. Section 6 reviews the related work. In the last section, we conclude and present the future work.

II. BACKGROUND

The high level view of Electronic Laboratory Notebook Protecting and Sharing (ELN-PS) system is shown in Figure 1. The main function of the ELN-PS system is to enable the modellers to securely transfer and share their personal ELNs within or across the research laboratories. In other words, it is an interface to transfer local ELNs into the central ELN repository for sharing purposes.

The key component in ELN-PS system is the Electronic Laboratory Notebook Dynamic Role-Based Access Control (ELN-DRBAC) mechanism that manages the access control to transfer, protect and share ELNs. The use of ELN-DRBAC gives clear understanding to the people: what their responsibilities are and to whom they are giving access. The

typical NIST RBAC as discussed in research [1] is enhanced to DRBAC because a person in the community (with the right permissions) may need to share a whole ELN or a selection of it (provenance trails) at a fine-grained level.

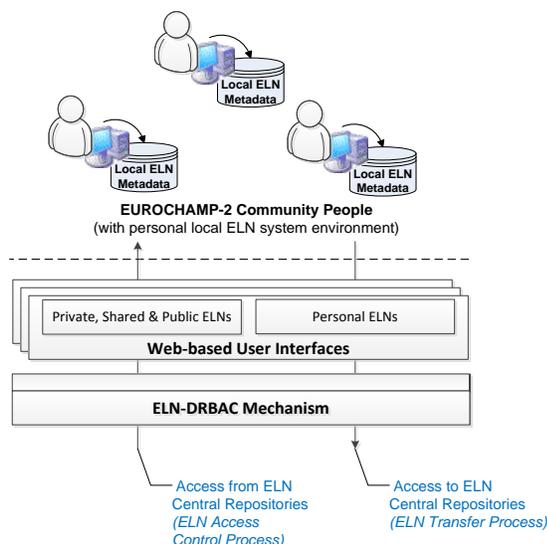


Figure 1. High level view of ELN-PS system.

In ELN-PS system mechanism, two dynamic allocations were introduced: a) persons to roles; and b) roles to permissions. The objective was to open the strict binding of three main components of the traditional RBAC system: users, roles and permissions to allow them to act dynamically within the RBAC framework. For example if a person with “modeller private” role wants to allow another person to download his/her personal ELNs, this mechanism gives the option to allocate “Download ELNs” permission to any person, independent of role allocation. This limited allocation of permission means that only those ELNs can be downloaded which are authorized by the ELN owner. Further a role may have different descriptions for different research groups. The “system manager” can activate and deactivate multiple permissions to any role according to the run time requirements of the domain people. In ELN-PS system, just like role sessions, the allocation of permissions to the roles is managed on the basis of permission sessions. So when a person uses ELN-PS system to protect and share ELNs, along with dynamic person-role sessions, the role-permission sessions are also created dynamically. This provides high level of dynamic inheritance in the ELN-PS system. Further detail about the ELN-PS system is referred to our previous research [1].

III. REQUIREMENTS FOR ELN-CPS SERVICE

In Section 2, the ELN-PS system is discussed to protect and share the generated ELNs locally in a co-laboratory research environment. The ELNs are valuable resources for e-Scientists as it helps with: the repeatability of experiments, tracking simulation runs, managing the data

generated, verifying experiment results and acts as a source of simulation insight [4].

The ELN-CPS service environment is shown in Figure 2 which encapsulates the functionality of the local ELN system. There are two main parts: a) ELN protecting and sharing control (ELN-DRBAC mechanism); and b) ELN service and associated sub services (i.e., ELN retrieval and IPNav services).

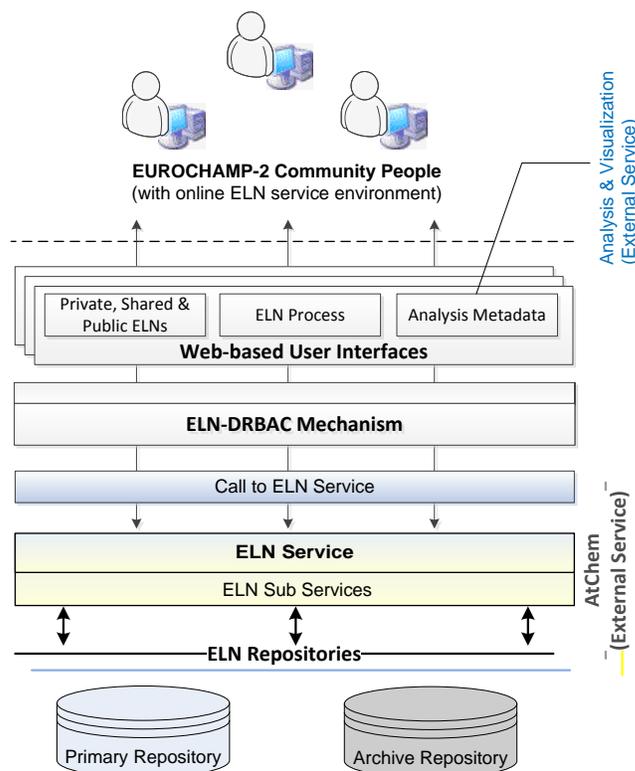


Figure 2. ELN-CPS service environment.

At this point, it is worth to note that the initial user requirements were captured in [1][2]. In this service, two types of service interfaces are required as given below.

Type 1 - Owner’s active ELN: Owner of the ELN needs access via three functions: a) initial setup of a new ELN (or retrieve from the repository); b) the provenance capture for the next simulation step (this includes access to the inline provenance node navigator (IPNav) service to update current provenance trails); and c) transfer of ELN to the repository (after completing the simulation). Owner of the ELN can invite other members (with the necessary permissions) in the community to read and comments on the current set of the simulation steps. The comments will be recorded in the repository. For example, the research manager can view and comments on the state of modeller’s personal ELN.

Type 2 - Third Party User: Retrieval of the ELN using the functions provided in the ELN-PS system. These are accessed through two services: Retrieve service (search

criteria and output of the list of matched ELNs); and ELN archives (ELN management, etc.).

These requirements arise from the Qualitative user-orientated evaluation was used to assess the meeting of the requirements and value of the ELN-CPS service. This is discussed in Section 5.

A. Scenario Cases

The scenario cases are used to develop the ELN-CPS service, which allows a modeller to create ELN simulations using Web-based interfaces, protected and shared by the ELN-PS system. The scenarios are derived from the working practices of the EUROCHAMP-2 community members, though remain generalisable to other communities with similar requirements.

Part-1: Creating ELN in the service environment

Helen is a modeller working in her laboratory. She is informed by the system manager that ELN system is now available in the service environment and she can access it through the Web-based interfaces. This service facility is provided to secure all ELNs centrally to avoid any disaster situation and to overcome the ongoing technical issues (i.e., ELN version control and the complexity of setting up and configurations) at the user’s local ELN system.

Helen logs into the ELN-CPS service by activating “modeller private” role. She initiates a new version of simulation called “toluene chamber” simulation and shares the whole ELN with her “research manager”. Helen understands that there is no need of transferring local ELN into the central repository because the ELN is already created online in the central repository and it is protected.

Part-2: Sharing of ELN in the service environment

Helen (modeller) and Alvin (research manager) exchange comments on a new simulation with different trails and finalize the results for evaluation. The golden trail is then shared with the publication editor for evaluation.

IV. ELN-CPS ARCHITECTURE

The ELN-CPS service architecture is designed with the vision to operate user-orientated ELN system in a distributed, co-laboratory research environment. The constraints of the existing ELN system architecture and framework [3] are also considered during the design phase. The ELN-CPS service architecture is designed based on the distributed computing model [5], as shown in Figure 3.

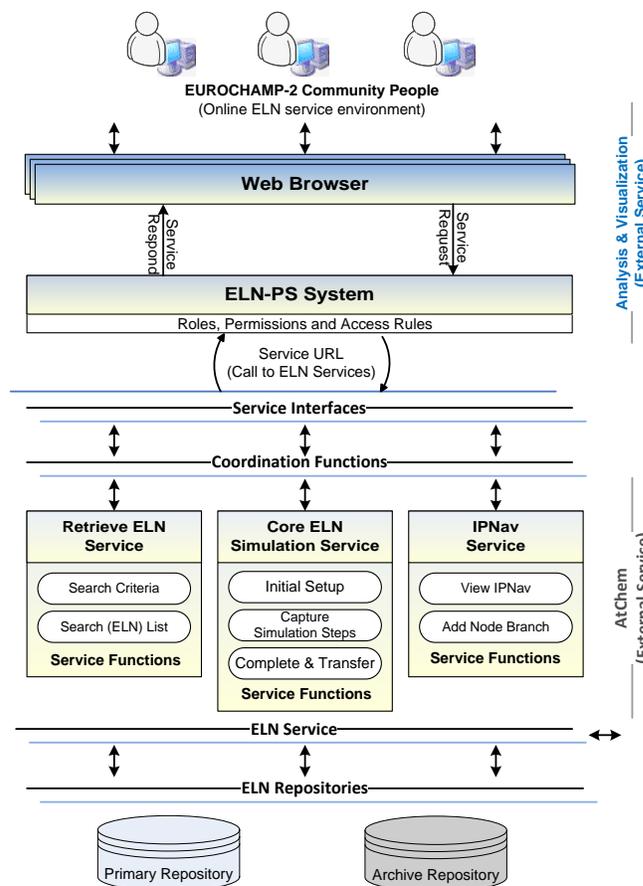


Figure 3. ELN-CPS service architecture.

The architecture grants the maximum configuration flexibility, because each function is mapped to a distinct physical (ELN development service) layer, which can be independently replicated. It presents the interrelation between different components of the ELN service, all accessible through the Web interfaces. In order to ensure loose coupling at the front-end of ELN-PS system, the proven Model-View-Controller (MVC) architecture [6] is embraced that efficiently handles session management for multiple Hypertext Transfer Protocol (HTTP) requests. The user interfaces as Java Server Faces (JSF) [7] does not contain any processing logic. These pages represent view part of MVC. Links in the user interface that requires processing logic to be executed submits the request that is mapped to an action. The action then selects and invokes the required processing logic. The processing logic is encapsulated in plain Java objects deployed as standard service.

The ELN-CPS service is developed using the JSF framework. All functions of the existing local ELN system are reused for this service. The service responses (i.e., send and receive) are managed according to the JSF framework. The ELN-CPS service keeps the detail of the service URL and invokes the service when the request for “Develop ELN” is generated from the modeller side. The sub services

are executed inside the main ELN service interface. JSF is a MVC framework that is capable of managing the front and middle tiers. To meet this goal, the design of the ELN-CPS service is based on the core JSF patterns as well as the industry standard development guidelines of scalability, flexibility and platform infrastructure. To serve the web pages, the Apache Tomcat web server is used which is one of the fastest and easiest to configure Java application servers [8]. The ELN-CPS service is implemented via the web. The modeller can access the service by using any web browser (e.g., google chrome, Mozilla Firefox, etc.) that they preferred. The core purpose of the scenario cases given in section 3 were to:

- i) Create the ELN simulations in a Web-based environment. Existing user-orientated ELN system only allows the creation of ELNs in a local environment.
- ii) Retrieve the ELN simulations using ELN service. It gives an advantage to the modellers to load the personal and shared ELNs into the online simulation interface for further simulations.
- iii) Protecting the ELN-CPS service through ELN-PS system.

Figure 4 shows a service interface in the ELN-CPS service environment to create a new simulation for a modeller who has assigned “modeller private” role. The modeller can invoke the service using “Develop ELN” function, which generates a request call and returns the service interface to create new simulation in the Web browser.

Figure 4. Creating new simulation.

This interface contains the basic parameters like simulation name, simulation type, EUROCHAMP chambers, etc. After setting up the initial parameters, the submit function is used to proceed further in mechanism development. To analyze the model output, the modeller has the option to download input/output model files in .zip format and can use their own tools as an external service.

V. EVALUATION WITH END USERS

The qualitative evaluation is presented in this section. This evaluation was the extension of our previous evaluation work [1]. The goal was to determine the potential value, likely advantages and disadvantages of using the ELN-CPS service. The evaluation plan included the demonstration of the two scenario cases (mentioned in Section 3) developed for ELN-CPS service and the collection of end user’s feedback using a specific evaluation questionnaires, designed for this purpose. Prior to this evaluation, the informal evaluations feedback were also captured during different presentations and discussions with the community members. However, due to geographical distributed locations of the community members and time constraint, we prefer to start formal evaluation process with two key evaluators having substantial experience of developing atmospheric chemistry models. Both of the evaluators regularly perform *in silico* experiments that make use of the Master Chemical Mechanism (MCM), so the evaluators could easily relate to and understand the ELN functionality they were presented with. The use of a small number of potential users, with close links to the software development team, to evaluate scientific software has also been applied successfully in the large e-Science projects, such as myGrid [9][10] and myExperiment [11][12]. By acting to meet the requirements of a small number of local, well known scientists (who acts as pioneers); whilst thinking about the requirements of the wider user community, a widely adopted software application can be developed.

A standard marking method using a likert scale was used to assess the answer of each question in the evaluation. Values were ranked from 1-5; 1 = poor - 5 = excellent. After the demonstration of each scenario case, the users were requested to answer the questionnaires and comments in the appropriate boxes. The answers obtained from the questions evaluated by two members of the community are given in

TABLE 1. ANSWERS FROM USER SURVEY.

Questions To Users	User	
	1	2
i) Do you acknowledge the concept of ELN-CPS service?	4	4
ii) Do you think the ELN-CPS service can overcome the maintenance issues, created in the local ELN system for the individual modeller?	3	4
iii) For security and protection of ELNs, do you think it is good to create, store and maintain ELNs online?	2	3
iv) Will you recommend the implementation of the ELN-CPS service in your laboratory?	3	3

At the end of the evaluation, the recommendations and comments from the participants were recorded. Samples of participant's recommendations and comments are provided below:

Recommendations and comments:

User 1:

- i) "Community needs to trust the party who controls ELN online with their scientific legacy – how do you gain their trust?"
- ii) An option to "back up" the private online ELN on their own private systems would promote the use of this service?"
- iii) "Need third party to look after the online service (need guarantees), e.g., British Atmospheric Data Centre (BADC)".

User 2:

- i) "Definitely the service can overcome the issue of maintenance. In local ELN, if you upgrade the functionality then you need to reinstall the ELN in every modeller's machine that use the local ELN. If using online, upgrading of the functionality would be easier".
- ii) "If I can trust the online system, then I will use it. But you need to think about how to persuade people to trust the service".

The evaluation results were very encouraging and both participants acknowledge the value of this service. These results could be considered as an initial feedback before going into the larger community for further evaluation. The major concern in the evaluation was regarding the trust relationship, i.e., how community people will be agreed to create ELNs outside their personal system? For this, the participants have given the following suggestions:

- i) To enhance the trust, there is a need to have an option for the end-users to store online ELN in their personal systems too;
- ii) The service should allow the modeller to choose either to perform their simulation runs locally or via online;
- iii) Need third party to look after the online service with their scientific legacy, e.g., British Atmospheric Data Centre (BADC) [19]. Guarantees should be signed by third party to ensure the security of the online ELN repositories.

VI. RELATED WORK

A service is an implementation of a well-defined logic functionality (in this research, it refers to the ELN service), and such services can then be consumed by clients (i.e., end-user application or another service) in different applications or processes [13]. Service Oriented Architecture (SOA) [20] is an architectural style for building software applications that use services available in a network such as the Web. SOA offers flexible approaches to distributed systems engineering including reuse of the existing applications.

Within the EUROCHAMP-2 community, the previously reported ELN [2] was based on a standalone local ELN to capture and retrieve the quality metadata performed by the modellers in a laboratory. However, there was an issue identified which was the need to reduce the complexity of setting up the ELN before it can be used [3]. Initially, virtualization [14] which provided a prefabricated environment for the ELN was implemented to address this issue; however, there is still a need for the ELN to maintain different versions of ELN metadata based on the role of the modellers. Two prominent projects related to this research are discussed. Although there are sheer number of work done related to this research, but we are only interested in the features aspects presented in those projects.

Linked Environments for Atmospheric Discovery (LEAD) is a large scale project to address meteorology research challenges to analyze and predict the atmosphere. Scientists are provided with necessary tools such as automated search, selection and transfer of required data products between computing resources [15] to build forecast models using model generated data and manage necessary resources for executing the model [16]. This includes: a) replacing the manual data management tasks with automated data discovery; b) allow transfer of large scale data products between resources; and c) searching and access of the data via GUI interface and underlying ontology [15]. LEAD project is deployed via web portal. This work seems similar to our research where features such as data discovery and transfer between computing resources are present. However, the ELN-CPS provides access and share simulations metadata at a fine grained level according to the modeller's role.

Open Source Project for a Network Data Access Protocol (OPeNDAP) is a system that facilitates scientific data archival and exchange between researchers [17]. The main feature of OPeNDAP is that it allows access to the data from a multiple applications. OPeNDAP uses the client/server model that utilizes the browsers to submit/receive requests/respond to the servers. It also allows researchers to browse and request data to be translated into a specific format. Three data object types provided by OPeNDAP [18], i.e., Data Descriptor Structure (DDS), which describes the structure of the data set, Data Attribute Structure (DAS) semantic metadata which gives the attribute values of the fields described in the DDS and the actual data in a binary structure. The different between OPeNDAP and ELN-CPS is that in the latter, it allows access and sharing of scientific data between modellers through the inline provenance node navigator (IPNav) and modeller's role. This includes access and sharing of the whole ELN or a selection of it at a fine-grained level in the trail.

VII. CONCLUSION AND FUTURE WORK

The research presented in this paper was the continuation of our previous research work. The aim was to introduce the ELN creating process in the service environment,

protected and shared through the ELN-PS system. The ELN-CPS service was developed and implemented to allow community modellers to create ELNs online and stored in the central ELN repository. This improved the consistency in the use of the ELN across the community and the management of the updated versions. The qualitative evaluation illustrated, how ELN-CPS service could be understood and accepted by a research community.

Future work will be to evaluate the ELN-CPS service in the wider ELN community. In order to get more conclusive results on the value of the ELN-CPS service, a larger set of ELN modellers is needed for evaluation. However, before going into the wider community for further evaluation, the issue of establishing a trust relationship between end-users and service providers needs to be addressed with robust plans for the safe storage of ELN data. This include considering the feedback from the evaluation which are to: 1) provide an option for the modelers to store their online ELNs into their local personal systems; 2) allow the modelers to perform their simulation runs locally or via online; and 3) allow the scientific legacy such as British Atmospheric Data Centre (BADC) [19] to look after the online service to ensure the security of the online ELN repositories. However, these recommendations need further investigation before it can be implemented.

ACKNOWLEDGEMENT

The research work presented in this paper was carried out at the School of Computing, Leeds University, supervised by Professor Peter M. Dew.

REFERENCES

- [1] Tahir Farooq, Richard Kavanagh, Peter Dew, and Zulkifly Mohd Zaki, "Protecting and Sharing of Semantically-Enabled, User-Orientated Electronic Laboratory Notebook Focusing on a case Study in the e-Science Domain," The Sixth International Conference on Future Computational Technologies and Applications, FUTURE COMPUTING 2014, May 2014, pp. 50-58.
- [2] Zulkifly Mohd Zaki, Peter M Dew, Lydia LMS Lau, Andrew R. Rickard, Jenny C. Young, Tahir Farooq, Michael J Pilling, and Chris J. Martin, "Architecture design of a user-orientated electronic laboratory notebook: A case study within an atmospheric chemistry community," *Future Generation Computer Systems*, vol. 29, Oct. 2013, pp. 2182-2196.
- [3] Z. Mohd Zaki, P. M. Dew, M. H. Haji, L.M.S. Lau, A. Rickard, and J. Young, "User-orientated electronic laboratory notebook for retrieval and extraction of provenance information for EUROCHAMP-2," *Proc. IEEE Seventh International Conference on eScience (ESCIENCE 11)*, Dec. 2011, pp. 371-378.
- [4] M. Greenwood, C. Goble, R. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn, "Provenance of e-Science Experiments - experience from Bioinformatics," *The UK OST e-Science second All Hands Meeting (AHM'03)*, pp. 223-226, [In UK e-Science All Hands Meeting, Nottingham, UK, Sep. 02-04, 2003].
- [5] Systems and software engineering - Architecture description, International Standard ISO/IEC/ IEEE 42010, IEEE, 2011.
- [6] Marty Hall, *Integrating Servlets and JSP: The Model View Controller (MVC) Architecture*, Prentice Hall and Sun Microsystems Press, 2010.
- [7] R. Hightower, *JSF life cycle, conversion, validation, and phase listeners Skill Level: Introductory*, IBM Corporation, 2008.
- [8] Xue Liu, Heo, J, and Lui Sha, "Modeling 3-tiered Web applications," *13th IEEE International Symposium*, Sept. 2005, pp. 307-310, DOI:10.1109/MASCOTS.2005.40.
- [9] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," *Nucleic Acids Res*, vol. 34 (Web Server issue), pp. 729-732, 2006.
- [10] T. Oinn et al., "Taverna: lessons in creating a workflow environment for the life sciences," *Concurrency and Computation: Practice & Experience - Workflow in Grid Systems*, vol. 18, Aug. 2006, pp. 1067-1100.
- [11] D. De Roure, and C. Goble, "myExperiment: A Web 2.0 Virtual Research Environment," *International Workshop on Virtual Research Environments and Collaborative Work Environments*, Edinburgh, UK, ePrint ID:263961, May 2007.
- [12] Carole Anne Goble, and David Charles De Roure, "myExperiment: social networking for workflow-using e-scientists," In: *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, June 2007, pp. 1-2, [ACM: Monterey, California, USA].
- [13] Carlos Granell, Laura Diaz, and Michael Gould, "Service-oriented applications for environmental models: Reusable geospatial services", *Environmental Modelling & Software*, vol. 25 Issue 2, Feb. 2010, pp. 182-198.
- [14] VMWare, "Virtualisation," <http://www.vmware.com/virtualization>, (retrieved: 15 February, 2015).
- [15] Beth Plale, Rahul Ramachandran, and S. Tanner, "Data Management Support for Adaptive Analysis and Prediction of the Atmosphere in LEAD," *22nd Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS)*, 2006.
- [16] Plale, B., K. Brewster, C. Mattocks, A. Bhangale, E. C. Withana, C. Herath, F. Terkhorn, and K. Chandrasekar, "Weather Forecast Data from The D2I-Vortex2 project," *Bloomington, Indiana: Data to Insight Center*, June 2012.
- [17] P. Cornillon, J. Gallagher, and T. Sgouros, "OPeNDAP: Accessing data in a distributed, heterogeneous environment," *Data Science Journal*, vol. 2, Oct. 2003, pp. 164-174.
- [18] Galip Aydin, *Service Oriented Architecture for Geographic Information Systems Supporting Real Time Data Grids*, PhD. Thesis, Indiana University, 2007.
- [19] British Atmospheric Data Centre, *Natural Environment Research Council*, <http://badc.nerc.ac.uk/home>, (retrieved: 15 February, 2015).
- [20] T. Erl, "Service-oriented Architecture: Concepts, Technology, and Design", Upper Saddle River: Prentice Hall. ISBN 0-13-185858-0, 2005.