PESARO 2015

The Fifth International Conference on Performance, Safety and Robustness in Complex Systems and Applications

ISBN: 978-1-61208-401-5

April 19 - 24, 2015

Barcelona, Spain

PESARO 2015 Editors

Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway

PESARO 2015

Foreword

The Fifth International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO 2015), held between April 19th-24th, 2015 in Barcelona, Spain, continued the inaugural event dedicated to fundamentals, techniques and experiments to specify, design, and deploy systems and applications under given constraints on performance, safety and robustness.

There is a relation between organizational, design and operational complexity of organization and systems and the degree of robustness and safety under given performance metrics. More complex systems and applications might not be necessarily more profitable, but are less robust. There are trade-offs involved in designing and deploying distributed systems. Some designing technologies have a positive influence on safety and robustness, even operational performance is not optimized. Under constantly changing system infrastructure and user behaviors and needs, there is a challenge in designing complex systems and applications with a required level of performance, safety and robustness.

We take here the opportunity to warmly thank all the members of the PESARO 2015 Technical Program Committee. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to PESARO 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 are grateful to the members of the PESARO 2015 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that PESARO 2015 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of performance, safety and robustness in complex systems and applications.

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

PESARO 2015 Advisory Committee:

Piotr Zwierzykowski, Poznan University of Technology, Poland
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Yulei Wu, Chinese Academy of Sciences, China
Harold Liu, IBM Research, China

PESARO 2015

Committee

PESARO Advisory Committee

Piotr Zwierzykowski, Poznan University of Technology, Poland
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Yulei Wu, Chinese Academy of Sciences, China
Harold Liu, IBM Research, China

PESARO 2015 Technical Program Committee

Amr Aissani, USTHB - Algiers, Algeria
Morteza Biglari-Abhari, University of Auckland, New Zealand
Salimur Choudhury, Queen's University - Kingston, Canada
Dieter Claeys, Ghent University, Belgium
Juan Antonio Cordero, École Polytechnique / INRIA, France
Nadir Farhi, IFSTTAR/COSYS/GRETTIA, France
John-Austen Francisco, Rutgers University - Piscataway, USA
Mina Giurguis, Texas State University - San Marcos, USA
Teresa Gomes, University of Coimbra, Portugal
Mesut Günes, FU-Berlin, USA
Michael Hübner, Ruhr-University of Bochum, Germany
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Harold Liu, IBM Research, China
Olaf Maennel, Tallinn University of Technology, Estonia
Kia Makki, Technological University of America (TUA), USA
Stefano Marrone, Seconda Università di Napoli, Italy
Birgit Milius, Institut fuer Eisenbahnwesen und Verkehrssicherung (IfEV) /Technische Universitaet – Braunschweig, Germany
Asoke Nandi, Brunel University, UK
Maciej Piechowiak, Kazimierz Wielki University - Bydgoszcz, Poland
Francesco Quaglia, DIAG - Sapienza Universita' di Roma, Italy
M. Zubair Rafique, KU Leuven, Belgium
Roger S. Rivett, Land Rover - Gaydon, UK
Dhananjay Singh, Electronics and Telecommunications Research Institute (ETRI), South Korea
Mukesh Singhal, University of California, Merced, USA
Young-Joo Suh, Pohang University of Science and Technology (POSTECH), South Korea
Yuejin Tan, National University of Defense and Technology - Changsha, China
Yang Wang, Georgia State University, USA
Yun Wang, Bradley University - Peoria, USA
Jun Wu, National University of Defense and Technology, China
Yanwei Wu, Western Oregon University, USA
Yulei Wu, Chinese Academy of Sciences, China

Copyright Information

Table of Contents

A Compositional Safety Specification Using a Contract-Based Design Methodology

Markus Oertel*, Peter Battram[†],
Omar Kacimi[‡] and Sebastian Gerwinn[§]
OFFIS – Institute for Information Technology,
Escherweg 2, 26121 Oldenburg, Germany
Email: *oertel@offis.de, †battram@offis.de,
‡kacimi@offis.de, §gerwinn@offis.de

Achim Rettberg
Carl von Ossietzky University Oldenburg
Escherweg 2, 26121 Oldenburg, Germany
Email: rettberg@iess.org

***Abstract*—Model-based design methodologies have become the standard approach to develop safety critical systems. Therefore, many approaches exist to model faults, failures and their propagation. Nevertheless, due to the frequent use of off-the-shelf components as well as the need to react efficiently on changes, the importance of modular and compositional techniques is gaining constantly. Here, we present an approach for compositional reasoning on safety specifications that supports multiple abstraction levels in the design process. Especially in the safety domain, it is obvious that a safety concept is just valid under certain conditions, e.g. that only a limited amount of components may fail at the same time. Therefore, we extend existing safety specification methods based on contracts, which explicitly distinguish between assumptions and guarantees, building a well-founded framework for compositional reasoning. Our formalization method can be used to develop a safety specification starting from the top level system component and refine it until the lower hardware and software layers while preserving the validity of early performed analyzes. On a practical level, we further describe how safety specifications can be formalized into a model checking problem and analyzed using existing tools.**

Keywords–Safety Critical Systems, Safety Contracts, Contract-based Design, Model-based Design, Fault Modeling, Model Checking, Formal Methods

I. Introduction

Safety relevant systems are characterized by a high amount of required verification and validation activities necessary for qualification or certification. Changes in the system often cause a tremendous re-verification effort. Fenn et al. [1] described the experience of industrial partners in which the costs for re-certification are related to the size of the system and not to the size of the change. Also, Espinoza et al. [2] discovered that the monolitic and process oriented structure of the safety cases required by nearly all domain-specific safety standards may require an entire re-certification of the system after changes. Therefore, recent work has focused on establishing modular specifications and processes aiming at preserving the safety properties even in the presence of changes [3][4][5]. Additionally, from a computational perspective, modular and decomposable specifications are highly desirable as the overall computational load can also be substantially reduced. That is, the analysis of the whole system in one state-space may not be feasible, while those of sub-parts are expected to be still in an acceptable performance corridor.

To obtain a compositional safety aspect, a specification language is needed that can describe the fault propagation and mitigation of a component, as well as a theory for composing components and comparing their specifications. Existing approaches to specify safety properties in a modular way (see Section II for an overview) either lack expressiveness, or compositionality as defined by Hungar [6] and Peng [7]: there exists a separation between the specification of a component and its actual behavior. If a component is replaced by another component meeting its original specification, the correct functional behavior of the composed system is maintained.

The main contribution of this paper is to add support of abstraction techniques to safety specifications to enable a top down design process. Requirements on higher abstraction levels leave room for many implementation possibilities (therefore also many different fault propagations), e.g. requiring the absence of single point of failures, which are specialized on lower levels of abstraction. The correct break-down of requirements can be automatically analyzed. The benefit of our specification is that the analysis results on higher level are not compromised if the system is further refined. Another benefit of such an approach is the possibility to automatically compare the assumed fault propagation with the behavior implemented [8]. Our work is based on the *design by contract* principle, which has a well-founded semantic framework for composing components and argue on the correctness of their refinement. As a language, we extend the already existing safety contracts with compositional aspects.

The paper is organized as follows. We briefly review the contract-based approach in Section III together with the existing safety contracts. We describe in Section IV how we can abstract from concrete component failures on higher abstraction levels and we propose a mapping to a linear temporal logic (LTL) based model checking problem (see Section V). Finally, we apply the approach to an example use-case (Section VI) and conclude our work in Section VII.

II. Related work

We briefly present the already existing approaches, which provide a modular or hierarchical safety specification for embedded systems.

A. HipHops

Hip-Hops [9] is a failure logic modeling-based approach [10] that aims at automating the generation of safety analysis artifacts such as Fault Trees and Failure Mode and Effect Analysis tables (FMEA). To this end, design models, e.g., SIMULINK models [11], are annotated with a fault propagation specification, which states for a component possible input and internal faults as well as how they influence the output. The deviations from the nominal behavior considered by Hip-Hops follow HAZOP [12] guidelines and include, for example, HAZOP guidewords *omission*, *too late*, and *value*. The atomic faults can be annotated with probabilities of occurrence, which in turn can be used along the structure of the generated fault tree to calculate the overall probability of a failure at top level. The popularity of the approach is based on the seamless integration in existing safety processes, since automation support is added to well-known analyses.

Our approach is more requirement-oriented. We directly formalize safety requirements and address the correctness of their refinement. Although the stated fault propagation relationships are similar, there is no need to generate a set of fault trees for each failure and their manual interpretation. As we are focusing on safety requirements and proving the correctness of their break-down, we do not need to compare fault trees with requirements.

B. Fault Propagation and Transformation Calculus

The fault propagation and transformation calculus (FPTC) [13] is addressing the problem of a modular safety specification. The architectural design is expressed using Real-Time Networks (RTN) to model the communication channels. The calculus itself is based on direct propagation notations using failure modes in accordance to HAZOP guidewords. The approach claims to be modular or compositional by providing a fixed-point algorithm to calculate the set of all possible occurring input and output faults for every component in the system. In contrast to the solution presented in this paper, the FPTC approach does not provide any support for multiple abstraction levels, i.e., specifying a component, then designing the subcomponents, and prove the correctness of the refinement.

Additionally, within FPTC it is not possible to state temporal properties in the propagation of faults (e.g. that a fault is going to be detected after some time). Furthermore, in case of multiple input and output ports, the notation of *positive propagation* is more complicated than the *negative fault containment* properties stated within safety contracts. Also, the order of the evaluation of the FPTC requirements matters, which is not the case for the invariant safety patterns. A minor drawback of FPTC is also the necessity of an RTN model, while safety contracts can be attached to any port-based component model.

C. Cause Effect Graphs

Kaiser et al. [14] address the problem that the typical structure of a fault tree is related to the hierarchy of the influences of a failure, but not to the hierarchy of the system. Therefore, they extend fault trees to be directed acyclic graphs, which they call Cause Effect Graphs (CEGs). This representation has the benefit of uniquely representing repeated events (identical causes used in multiple sub-trees) in fault trees. The CEGs allow to create a direct mapping of parts of the fault tree to components in the architecture, making these elements reusable. A tool called UWG3 has been developed to evaluate the approach.

Kaiser et al. address compositionality and re-useability in their approach, but do not integrate abstraction. At the time of the analysis the model still needs to be complete and cannot be refined later without the need of a full re-evaluation. The same limitations with respect to the interpretation of fault trees as they have been stated for Hip-Hops (see Section II-A) also apply to this approach.

D. Previous work on safety contracts

Boede et al. [15] presented an approach for hierarchical safety specifications using safety contracts. They introduced patterns to describe the hierarchy of faults, the hierarchy of functions, and the relation between faults and functions, e.g.: `function <function-name> can be impacted by <failure-list>`. Although the approach is based on contracts, the stated patterns do not fully exploit the contract notation as assumptions are neglected, a major principle of contracts.

While stating the need for a hierarchical failure specification language, the semantics for the presented patterns are not precisely given. Hence, it is hard to estimate if the proposed analysis can be automated.

Oertel et. al [16] presented an approach for expressing safety properties in a contract-based fashion providing formally defined semantics and application guidelines. The presented approach, however, did not consider abstraction techniques to refine safety concepts suitable for a top-down design. Since the approach presented in this paper is extending the work of Oertel et. al [16], the main contract templates and their semantics are briefly presented in Section III-B.

III. Background

A. Overview Contract-based Approach

Contracts [17][18] separate a requirement into an *assumption*, which describes the expected properties of the environment, and a *guarantee*, which describes the desired behavior of the component under analysis, provided the assumption is met by the operational context, i.e., the environment. This separation allows for building a sound theory thereby permitting to reason in a formal way about the composition of systems. Contracts, belonging to the class of assume-guarantee reasoning techniques [19], are a widely adopted approach for compositional verification [20][7].

Contract semantics for reactive and embedded systems are defined over traces of systems [6][18]. Components, in the following denoted with $\mathtt{M}$, are characterized by ports (P), that are either defined as input or as output ports. A trace assigns a value V out of the value domain $\mathcal{V}$ to each of the ports at any given point in time $t \in \mathcal{T}$. Therefore, a trace is of the form $[\mathcal{T} \rightarrow [P \rightarrow \mathcal{V}]]$. The traces of a component $\mathtt{M}$ are denoted $[\![\mathtt{M}]\!]$. This set comprises all possible behaviors of a component (implementation), even in case of unacceptable inputs due to a general requirement of input openness, i.e., requiring systems to never confine or otherwise refuse input. The (semantically concurrent) composition of multiple sub-components $\mathtt{M}_1 \ldots \mathtt{M}_n$ to a component $\mathtt{M}$ is defined [18] as the set of traces accepted by all components:

$$[\![\mathtt{M}]\!] = \left[\!\!\left[\bigtimes_{i=1}^{n} \mathtt{M}_i \right]\!\!\right] = \bigcap_{i=1}^{n} [\![\mathtt{M}_i]\!]$$

In some modelling languages it is possible to name the port-ends of delegation- and assembly-connectors differently. For the same of simplicity we do not foresee such a behavior and assume to have identical port names if they are connected.

A contract is a tuple $C = (A, G)$ describing a set of traces using the assumption A and the guarantee G:

$$[\![C]\!] = [\![A]\!]^{-1} \cup [\![G]\!]$$

with $(.)^{-1}$ denoting the complement of a set. Although not formally required by the definition of contracts, assumptions typically only specify constraints on the input, whereas guarantees reflect the input/output relation and therefore contain restrictions of allowed outputs while being input-open. We assume all contracts to be stated in the *canonical form* [21], which allows some simplifications in the following definitions of the operators and relations. A contract is said to be in canonical form if $[\![G]\!]$ at least contains $[\![A]\!]^{-1}$.

In order to reason about the correctness of the composition of a system, a set of basic relations and operators are defined on contracts and implementations.

Definition 1 (Satisfaction)**.** *A component M satisfies [18][22][21] a contract $C = (A, G)$, denoted $M \models (A, G)$, iff all its traces are permitted by the contract, i.e., $[\![M]\!] \subseteq [\![C]\!]$, and its inputs and outputs coincide to those underlying the contract. Consequently,*

$$M \models (A, G) \Leftrightarrow [\![M]\!] \cap [\![A]\!] \subseteq [\![G]\!]$$

Refinement is a relation between two contracts, stating that the refined contract is a valid concretisation of the other, i.e., the refining contract is a valid replacement in all possible operational contexts satisfying all (and maybe more) requirements satisfied by the refined contract.

Definition 2 (Refinement)**.** *According to [17] and [22] a contract C_1 refines C_2 iff it has the same signature, i.e., same input and output ports, yet imposes relaxed assumptions and more precise guarantees:*

$$C_1 \preceq C_2 \Leftrightarrow A_1 \supseteq A_2 \wedge G_1 \subseteq G_2$$

Parallel composition ($\otimes$) is used to combine multiple contracts into a new contract that represents the behavior of (concurrently) composed components each of which satisfies their individual contracts.

Definition 3 (Parallel Composition)**.** *For two contracts C_1 and C_2, the parallel composition $C_1 \otimes C_2$ is defined as:*

$$((A_1 \cap A_2) \cup \neg(G_1 \cap G_2), (G_1 \cap G_2))$$

B. Overview of Safety Contracts

The assumption and the guarantee of a contract can be described using various languages depending on the analysis target. In case of safety requirements we use a formal, pattern-based language, called *safety patterns* [16], which can be translated to many other suitable target languages, such as LTL or timed-automata [18]. By using LTL in [16], four safety patterns are introduced to express different safety properties like fault propagation, safety mechanisms and transitions to a safe-state. These patterns are depicted in TABLE I.

TABLE I. Overview of the Safety Patterns

Pat. 1	Structure	**none of** $\{\{m_1, m_2\}, \{m_3, m_4\}\}$ **occurs**
	Intuition	This pattern speaks about malfunction or mode (m) combinations that are not allowed.
	LTL expression	$(\mathbf{G}\neg m_1 \vee \mathbf{G}\neg m_2) \wedge (\mathbf{G}\neg m_3 \vee \mathbf{G}\neg m_4)$.
Pat. 2	Structure	expr-set **does not occur**.
	Meaning	Derived from pattern 1, where just one set of malfunctions or modes is used. Semantics are identical. Introduced for convenience.
Pat. 3	Structure	m_1 **only followed by** m_2
	Intuition	It is proposed to restrict the occurrence of m_1. The pattern is typically used to define consecutive modes.
	LTL expression	$\mathbf{G}(m_1 \rightarrow (m_1 \mathbf{W} m_2))$.
Pat. 4	Structure	m_1 **only after** m_2
	Intuition	It is proposed to restrict the occurrences of malfunctions or modes. This pattern is typically used to express that a safety mechanism shall only be activated after the fault occurrence.
	LTL expression	$\neg m_1 \wedge G((\mathbf{X} m_1) \rightarrow (m_1 \vee m_2))$

Pattern 1 and its derived Pattern 2 state that only system runs (traces) are accepted in which the combination of malfunctions (as stated in the expression sets) is absent. Pattern 3 and Pattern 4 are used to order malfunctions or modes of the system.

Oertel et. al [16] presented several contract templates that proved to be useful for automotive safety concepts. We are focusing here on a subset of them, since the others can be handled identically. The basic template is depicted in contract C_1 and describes the robustness of the output of a component against internal faults. More precisely, the contract explicitly mentions all combinations M_f of malfunctions on the inputs (Inp_{mf}) or internal malfunctions (Int_{mf}) of an component that could cause the output to fail (`out_fail`). Therefore, $M_f \subseteq \mathcal{P}(Inp_f \cup Int_f)$ and the corresponding safety contract consisting of assumptions A and guarantees G can be written as:

C_1	**A:**	**none of** $\{M_f\}$ **occurs.**
	G:	$\{$`out_fail`$\}$ **does not occur.**

In order to fulfill the requirements of the ISO 26262, safety contracts are able to express the degradation of a system, i.e., switching to a safe-state. This safe-state is expressed in functional terms (e.g., functional variables need to be in a defined range), and is therefore just considered as an identifier

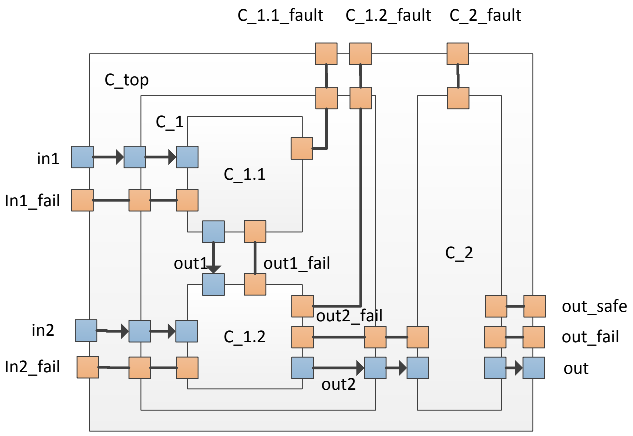

Figure 1. All internal faults were "externalized" using fault activation ports. Blue ports are considered as functional ports, while red ports describe the failure modes of the functional ports.

in safety contracts. Contract template C_2 is using this mechanisms to state that a system is either operating normally or that it is in a safe-state, if the assumption is met. The assumption is violated if one of the stated combinations of malfunctions occurs.

C_2 **A**: **none of** $\{M_f\}$ **occurs.**
G: {out_fail **and** !out_safe} **does not occur.**

In Oertel et al. [16], the top level contract talks about all non-desired atomic fault combinations, even if the atomic faults correspond to components of a more detailed abstraction level. Figure 1 illustrates that all faults in the most detailed (lowest abstraction level) components need to be made available to the containing components by use of fault ports. I.e., that the designer needs to know all possible internal faults while specifing the top-level component. This approach is not compositional, since changes on internal components directly cause changes on the containing component. Furthermore, a top-down design is impossible, since it is assumed that all components and their faults are known in advance.

IV. Compositional Faults & Fault Abstraction

Every component is only safe under certain conditions. For example, a dual-channel architecture is only safe as long as at least one channel is working correctly. In a contract-based approach these assumptions are explicitly stated for each requirement. Therefore, the contract assigned to the top level component guarantees the correctness or "safeness" of an output while assuming only a limited set of combinations of particular internal faults to occur in the system.

The assumptions that restrict the correctness of an output signal can in many cases be abstracted to restrict only the number of faults within the system. Using this technique we are still able to express the absence of single-points of failure, as required by the ISO 26262, but do not need to state all particular faults individually. Thereby this generalization enables a top-down design approach.

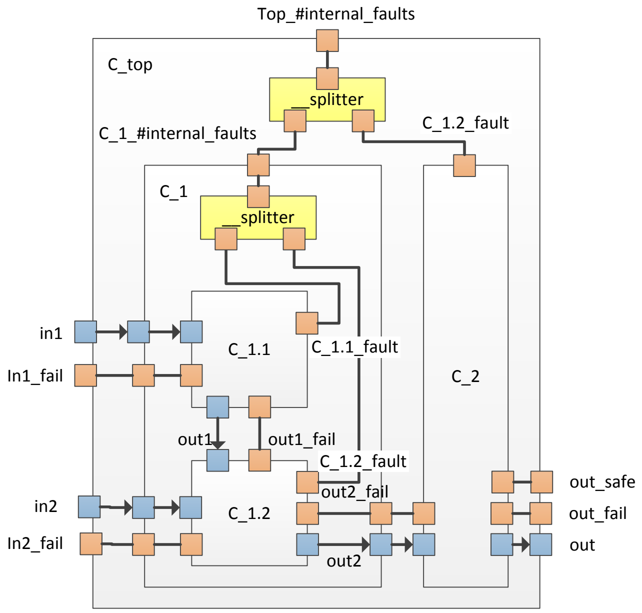

Figure 2. Count ports are introduced and fault-splitter components, that ensure that not more faults are passed to internal components as specified

Therefore, since traces are based on values on ports, we introduce *fault count* ports representing how many internal faults are present in a component (see Figure 2). Again, blue ports represent actual functional values that are used in the implementation of that component and red ports represent failure modes assigned to these ports. Notice, that more than one failure mode can be assigned to a functional port, as it is the case the out port. Using fault count ports, only at the most detailed abstraction level, i.e., the implementation, the particular faults that might occur in a component need to be specified. The fault count ports are of type Integer and belong to the safety aspect of a system only. From a functional point of view, these ports are virtual and do not occur in any generated code. Additionally, we also introduce virtual *splitter components* to create a connection between the fault count port on the higher component and the fault ports (or also fault count ports) on the sublevel components. They are called virtual, since they are not necessary for the specification, they are just necessary to still be compliant to the contract based design methodology and can therefore be generated automatically. The splitters distribute the maximum number of faults of the surrounding component to its subcomponents. Consequently, they allow multiple possibilities of how to distribute the number of allowed faults. On the one extreme, they could be just assigned to one single component and all the other subcomponents are assumed to be fault free or, on the other extreme, they could be evenly distributed aross all subcomponents. Under all of these possibilities the component needs to be safe. In terms of trace semantics, these splitters only allow traces that do respect the number of faults. For n subcomponents, the contract associated to the splitter component can also be characterized by the following contract:

C_3 **A**: true
G: $\texttt{out}_1 + \ldots + \texttt{out}_n < \texttt{\#internal\_faults}$

The guarantee creates a relation between the input port of the splitter (#internal_faults) and the output ports (out_n). If the assumption of a component states that x internal faults do not occur, this contract ensures that only less than x faults are activated on the subcomponents.

As the components are still characterized by input and output ports contracts are still applicable to the safety aspect using traces (see Section III) on fault-count ports and all definitions and analyses from contracts (refinement, etc.) remain valid. Fault count ports are only used for the internal faults of an component, fault that occur at the input ports are not affected since these faults can be easily passed to an additional subcomponents, which are introduced in a refinement step, without violating compositionality.

Intermediate and top level component's safety requirements can be specified using these counting fault ports. The most common usage for the top level assumption is to state that only one or two faults occur in the system. For example, a specification for the component C_1 in Figure 2 could be stated as follows:

C_4

A: **none of** {{in1_fail, in2_fail}, {C_1_#internal_faults=1, in1_fail}, {C_1_#internal_faults=1, in2_fail}, {C_1_#internal_faults=2} } **occurs.**

G: {out2_fail AND !out2_safe} **does not occur.**

This specification defines that the component shall be robust against one arbitrary fault, which may occur at one of the inputs or internally. Another useful scenario is, that the inputs are assumed to be correct and only internal faults are considered. Such a contract can be stated as follows:

C_5

A: **none of** {{C_1_#internal_faults=2}, {in1_fail}, {in2_fail} } **occurs.**

G: {out2_fail AND !out2_safe} **does not occur.**

The specification of an atomic component (like C_1.2 in Figure 2), for which the internal faults are known, uses the known internal faults, since no abstraction by using a fault count is necessary. The internal faults are represented as individual fault ports (see C_1.2_fault on component C_1.2 in Figure 2) and are directly connected to the splitter components. This ensures, that the specification of the atomic component can still be done accordingly to the guidelines presented by Oertel et al. [16], however one can additionally restrict the total number of different individual faults occurring simultaneously (in the same trace) on a more abstract level.

V. Semantics and Verification of Fault Abstraction

To be able to automatically check our specification and to provide precise semantics, we provide a mapping to boolean LTL. [23]. This decision is based on the availability of fast LTL model checkers and the benefit of relying on the same formalism used for the safety patterns itself (see Section III-B). Furthermore, assumption including an internal faults count bigger than 3 is not expected. Therefore, we cannot use the fault-count ports as integers, but need to create a mapping to boolean logic. Still, from a users perspective the workflow does not change, the specification should be written using the integer fault ports, the mapping is performed completely automatically.

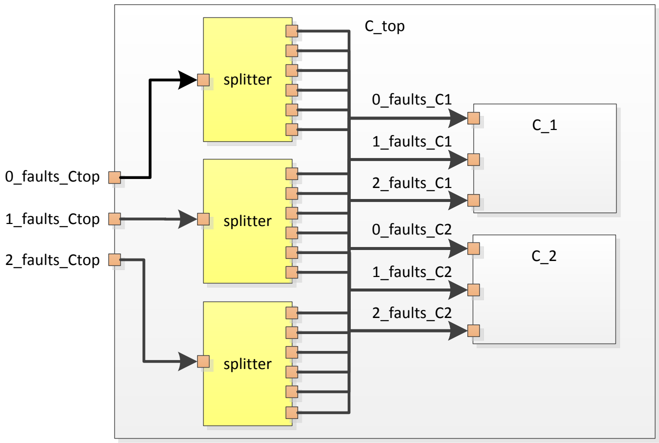

Figure 3. Explicitly represent the counting with boolean logic for LTL implementation. In this example a fault port with the value 2 is represented.

Since all numbers are explicitly stated, and we do not consider unbounded variables, the integer ports can be expressed in a combinatorial fashion. Hence, a fault count port is split up in individual boolean ports (*explicit fault count ports*) representing each a valid value of this port (see Figure 3). If a contract assumes that not more than n faults occur, all numbers of faults smaller than n are valid values. It needs to be stated, that only one of these ports can be active at a time. This is an assumption that needs to be added to all contracts that are attached to the component, e.g., for C_{top}:

C_6

A: **none of** { {0_faults_Ctop, 1_faults_Ctop}, {1_faults_Ctop, 2_faults_Ctop}, {0_faults_Ctop, 2_faults_Ctop} } **occurs.**

G: ...

Furthermore, we need a separate splitter component for each explicit-fault-number port. These splitters allow all logical combinations of ports that sum up to the defined number of faults. For n internal faults and m subcomponents the splitter contract restricts the set of all possible occurring malfunctions $M = \mathcal{P}(\{X_0\_faults\_C0, \ldots, X_n\_faults\_Cm\})$ to M_f:

C_7

A: true

G: **none of** $\{M_f$ } **occurs.**

with $M_f = \{\{x_i\_faults\_C_y\} \subseteq M : \sum_{i=1}^{(n+1)\cdot m} x_i \geq n\}$.

For the purpose of a single refinement analysis we can simplify the translation, since at analysis time splitters are not needed and can be replaced by all combinations of faults of the subcomponents directly. For a given set of internal faults I, the fault-count port `#internal_faults=n` in the assumption resolves to all sets of faults of length n of the powerset $\mathcal{P}(I)$. The refinement check is then implemented as a satisfaction check on the LTL properties of the contracts (see Section III-A). Rozier and Vardi [24], as well as Li et al. [25] suggested to use a generic model, allowing all possible behavior, to check the property against to reduce the problem to a classical model checking problem.

VI. Example

We model a fail-safe temperature sensor as an example (see Figure 4) to apply the top-down design process for safety contracts. The main safety requirement for the temperature sensor is to still deliver a correct or at least *safe* result under the assumption that there is at most one fault present within the system:

C_8
```
A: none of {{#sensor_faults=2}} occurs.
G: {temp_out_fail AND !temp_out_safe}
   does not occur.
```

The safe-state of the temperature sensor is to output the maximum temperature. This is a valid safe state if the temperature sensor is used in a cooling system, where overheating should be prevented.

A realization of this requirement can be a design with two simple but redundant temperature sensors in addition to a logic, which provides a voting mechanism. The sensors themselves do not provide any safety mechanisms and hence can fail immediately as a result of a single internal failure. Therefore, the safety contract for the temperature sensors 1 is (temperature Sensor 2 is specified in an identical manner):

C_9
```
A: none of {{temp1_fail}} occurs.
G: {temp1_out_fail} does not occur.
```

The logic component shall react to faults of the temperature. As the internal structure has not yet been decided, the requirement states, that even if one of the inputs fails or an internal fault occurs the result should at least be *safe*. Such a requirement can be expressed using a safety contract:

C_{10}
```
A: none of {
   {temp1_out_fail, temp2_out_fail},
   {#logic_faults=1, temp1_out_fail},
   {#logic_faults=1, temp2_out_fail},
   {#logic_faults=2}
   } occurs.
G: {temp_out_fail AND !temp_out_safe}
   does not occur.
```

In the refinement step of the logic component, two independent analog/digital converters are used to digitize the temperature signal. Both converters do not provide safety mechanisms and fail immediately:

C_{11}
```
A: none of {{ad1_fail}} occurs.
G: {ad1_out_fail} does not occur.
```

Again, the second converter is specified similarly. The signals are processed by an overwrite component, which compares the results and sets the safe-state if the values differ. The override component is not expected to fail in the lifetime of the device. It is a common assumption in voting architectures to assign a very small functionality to a component that is formally verified and produced in a more robust way than the rest of the system or replaced in a regular manner during service intervals. The safety contract is therefore specified only in terms of input faults:

C_{12}
```
A: none of {
   {ad1_out_fail, ad2_out_fail}
   } occurs.
G: {temp_out_fail AND !temp_out_safe}
   does not occur.
```

The corresponding splitter component, the connection of the splitter to the faults of the subcomponents as well as the contract for the splitter component can be generated automatically and do not need to be specified separately.

Refinement can now be checked on both levels of abstraction.

$$(C_7 \otimes C_8 \otimes C_9) \preceq C_6$$

as well as

$$(C_{10} \otimes C_{11} \otimes C_{12}) \preceq C_9$$

VII. Conclusion

Since the verification and validation of safety-critical systems consume up to 40% of the development costs, it is even highly undesirable if changes in a system require a re-verification of the whole system. Modular safety cases are expected to reduce these costs by enabling a determination of the area of the system affected by the incorporated changes. In this paper, we presented an approach to enable black-box safety specifications using safety contracts. In contrast to other approaches contracts provide a means of abstraction thereby allowing to develop a system in a top-down manner, i.e., to refine the specification by introducing the possible architecture of the sub-components. Being able to analyze the correctness of a refinement in an LTL-based model checking tool, we are now able to determine if a change in safety requirements needs further adaptations of the system to be compliant to that change. Although this approach is expressive in its specification, it is still intuitively to use.

The currently used LTL-backend does not allow to analyze big system. For some examples with complex specifications, we were not able to check more than 10 contracts in one refinement analysis. Alternatively, an automaton-based representation together with a bounded model checking technique could be a promising candidate for improving the efficiency. Furthermore, it seems to be possible to annotate the faults with probabilities of their occurrence to extend the scope of possible requirements to upper bounds in probabilities that a system may fail, rather than the number of faults.

Acknowledgment

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n°269335 (MBAT) and the German Federal Ministry of Education and Research (BMBF) as well as $01IS12005M$ (SPES_XT).

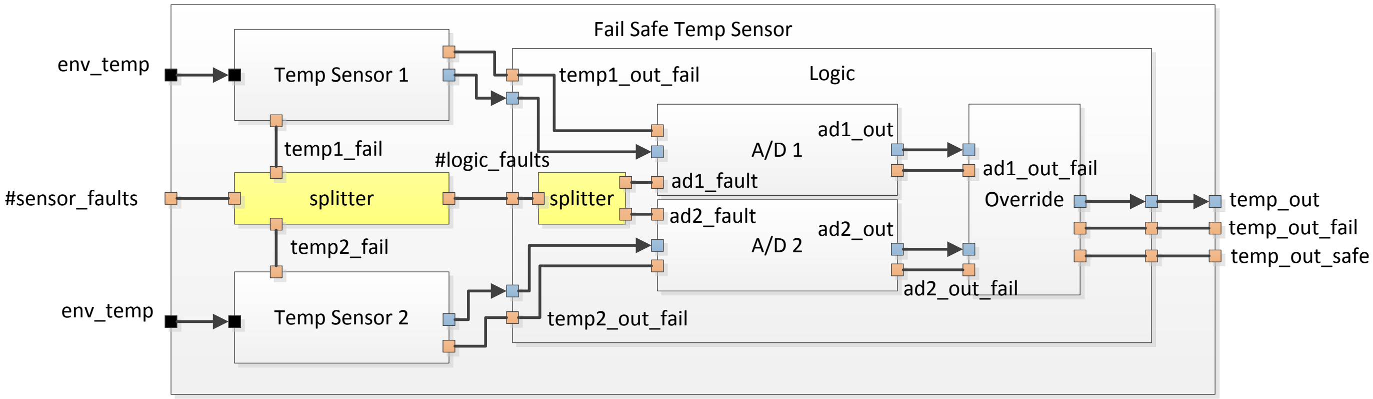

Figure 4. Architecture of a temperature sensor required to be robust against single-points of failure

REFERENCES

[1] J. L. Fenn, R. D. Hawkins, P. Williams, T. P. Kelly, M. G. Banner, and Y. Oakshott, “The who, where, how, why and when of modular and incremental certification,” in System Safety, 2007 2nd Institution of Engineering and Technology International Conference on. IET, 2007, pp. 135–140.

[2] H. Espinoza, A. Ruiz, M. Sabetzadeh, and P. Panaroni, “Challenges for an open and evolutionary approach to safety assurance and certification of safety-critical systems,” in Software Certification (WoSoCER), 2011 First International Workshop on. IEEE, 2011, pp. 1–6.

[3] M. Oertel and A. Rettberg, “Reducing re-verification effort by requirement-based change management,” in Embedded Systems: Design, Analysis and Verification. Springer Berlin Heidelberg, 2013.

[4] J. Fenn, R. Hawkins, P. Williams, and T. Kelly, “Safety case composition using contracts-refinements based on feedback from an industrial case study,” in The Safety of Systems. Springer, 2007, pp. 133–146.

[5] J. Vara, S. Nair, E. Verhulst, J. Studzizba, P. Pepek, J. Lambourg, and M. Sabetzadeh, “Towards a model-based evolutionary chain of evidence for compliance with safety standards,” in Computer Safety, Reliability, and Security, ser. Lecture Notes in Computer Science, F. Ortmeier and P. Daniel, Eds. Springer Berlin Heidelberg, 2012, vol. 7613.

[6] H. Hungar, “Compositionality with strong assumptions,” in Nordic Workshop on Programming Theory. Mälardalen Real–Time Research Center, November 2011, pp. 11–13.

[7] H. Peng and S. Tahar, “A survey on compositional verification,” Dep. of Electrical & Computer Engineering, Concordia University, Canada, Tech. Rep., 1998.

[8] M. Oertel, O. Kacimi, and E. Böde, “Proving compliance of implementation models to safety specifications,” in Computer Safety, Reliability, and Security, ser. Lecture Notes in Computer Science, A. Bondavalli, A. Ceccarelli, and F. Ortmeier, Eds. Springer International Publishing, 2014, vol. 8696, pp. 97–107.

[9] Y. Papadopoulos and J. McDermid, “Hierarchically performed hazard origin and propagation studies,” in Computer Safety, Reliability and Security, ser. Lecture Notes in Computer Science, M. Felici and K. Kanoun, Eds. Springer Berlin Heidelberg, 1999, vol. 1698, pp. 139–152.

[10] O. Lisagor, J. McDermid, and D. Pumfrey, “Towards a practicable process for automated safety analysis,” in 24th International system safety conference, 2006, pp. 596–607.

[11] Y. Papadopoulos and M. Maruhn, “Model-based synthesis of fault trees from matlab-simulink models,” in Dependable Systems and Networks, 2001. DSN 2001. International Conference on, 2001, pp. 77–82.

[12] J. Gould, M. Glossop, and A. Ioannides, “Review if hazard identification techniques,” Health and Safety Laboratory, Tech. Rep., 2000.

[13] M. Wallace, “Modular architectural representation and analysis of fault propagation and transformation,” in Proc. FESCA 2005, ENTCS 141(3), Elsevier, 2005, pp. 53–71.

[14] B. Kaiser, P. Liggesmeyer, and O. Mäckel, “A new component concept for fault trees,” in Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software - Volume 33, ser. SCS ’03. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, pp. 37–46.

[15] E. Böde, S. Gebhardt, and T. Peikenkamp, “Contract based assesment of safety critical systems,” in Proceeding of the 7th European Systems Engineering Conference (EuSEC 2010), 2010.

[16] M. Oertel, A. Mahdi, E. Böde, and A. Rettberg, “Contract-based safety: Specification and application guidelines,” in Proceedings of the 1st International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC 2014), 2014.

[17] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-vincentelli, W. Damm, T. Henzinger, and K. Larsen, “Contracts for systems design,” Research Centre Rennes - Bretagne Atlantique, Tech. Rep., 2012.

[18] A. Baumgart, E. Böde, M. Büker, W. Damm, G. Ehmen, T. Gezgin, S. Henkler, H. Hungar, B. Josko, M. Oertel, T. Peikenkamp, P. Reinkemeier, I. Stierand, and R. Weber, “Architecture modeling,” OFFIS, Tech. Rep., 03 2011.

[19] T. Henzinger, S. Qadeer, and S. Rajamani, “You assume, we guarantee: Methodology and case studies,” in Computer Aided Verification, ser. Lecture Notes in Computer Science, A. Hu and M. Vardi, Eds. Springer Berlin Heidelberg, 1998, vol. 1427, pp. 440–451.

[20] W.-P. de Roever, “The need for compositional proof systems: A survey,” in Compositionality: the significant difference. Springer, 1998, pp. 1–22.

[21] A. Benveniste, B. Caillaud, A. Ferrari, L. Mangeruca, R. Passerone, and C. Sofronis, “Multiple viewpoint contract-based specification and design,” in Formal Methods for Components and Objects, ser. Lecture Notes in Computer Science, F. de Boer, M. Bonsangue, S. Graf, and W.-P. de Roever, Eds. Springer Berlin Heidelberg, 2008, vol. 5382, pp. 200–225.

[22] B. Delahaye, B. Caillaud, and A. Legay, “Probabilistic contracts: a compositional reasoning methodology for the design of systems with stochastic and/or non-deterministic aspects,” Formal Methods in System Design, vol. 38, no. 1, 2011, pp. 1–32.

[23] A. Pnueli, “The temporal logic of programs,” in Foundations of Computer Science, 1977., 18th Annual Symposium on. IEEE, 1977, pp. 46–57.

[24] K. Y. Rozier and M. Y. Vardi, “Ltl satisfiability checking,” in Model Checking Software, Proceedings of the 14th International SPIN Workshop. Springer, 2007, pp. 149–167.

[25] J. Li, G. Pu, L. Zhang, M. Y. Vardi, and J. He, “Fast ltl satisfiability checking by sat solvers,” arXiv preprint arXiv:1401.5677, 2014.

Using Community Structure Information to Improve Complex Networks Robustness

Cinara G. Ghedini* and Carlos H. C. Ribeiro*
*Aeronautics Institute of Technology
Computer Science Division
São José dos Campos – SP – Brazil
Email: cinara,carlos@ita.br

***Abstract*—This paper discusses the relation between two emergent features of most complex networks: community structure and high sensitivity to attacks. More specifically, we consider how the former can support mechanisms to mitigate the latter. The main point stressed here is that information about the community structure can be useful to detect and mitigate vulnerable topological configurations w.r.t network connectivity. We demonstrate this through an attack and failure protocol that considers the importance of central nodes regarding their roles connecting nodes, either inside or outside communities. We also propose local mechanisms for evaluating topological configurations based on community information. The strategy for minimizing the impact of central node failures to network connectivity relies on the creation of redundant paths between communities. The networks evaluated exhibited a significant improvement in their robustness regarding connectivity maintenance, being almost unaffected by failures of central nodes. The experimental benchmark encompasses both real complex network datasets and networks generated by well-established construction methods.**

Keywords–attacks and failures tolerance; community structure; adaptive mechanisms

I. Introduction

Complex networks exhibit unique characteristics that are often discussed in the literature, as the small-world effect, the clustering coefficient and the degree distribution [1]–[4]. Beside them, community structure is one of the main properties of real complex networks undergoing studies. Its features can support the understanding of the network formation and evolution processes, the collective and individual behavior, the information dissemination, among others, thus, being useful for a broad range of applications, from disease spreading to marketing.

Despite some nuances deriving from topological characteristics of each network, it is known that some property values combinations, which occur in several real networks, can produce robust networks regarding failures. On the other hand, when the most central nodes fail, the topology of complex networks is fairly affected, compromising the network operation [5]–[11]. Despite this eminent feature being well stated, mechanisms to evaluate and mitigate such states are mostly neglected. Analytical approaches for global robustness estimation are presented in [12][13]. Wang *et al.* [14] propose a global approach for supporting the design of networks through mechanism for detecting and protecting those links which are crucial for the network robustness.

Based on this, our approach explores community structure information for detection and mitigation of vulnerable topological configurations. For that, we evaluate the impact that links connecting elements in the same community (inside links), different communities (outside links) or both, have on the network topology when they fail. The results demonstrate that nodes connecting different communities indeed play a central role regarding the communication among network elements. Taking advantage of this information, mechanisms to identify possible harmful topological configuration and to promote adjustments on the network are proposed. Such mechanisms are based on previous work presented in [15], which considers a node as the main agent for detecting vulnerability and local efficiency as the measure for representing the state of vulnerability. Here, we propose using the community local efficiency for both goals. Differently, Yang *et al.*, in a very recent paper [16], use community information for supporting a global link rearrangement procedure, as a possible way to improve the network robustness, without taking into account the detection of harmful configurations or methods to precisely revert them.

For benchmarking the experiments, we rely on real network datasets and networks generated by two constructive models: Barabasi and Albert's [2] and Klemm-Euguluz [17] models, plus a protocol to promote perturbations on the network and classical topological measures, such as the global and local efficiencies, and the size of the giant component. The findings are that concepts related to community structure can be used to improve the surveillance of communication and service networks in case of failures and attacks. In addition, they can be applied for maintaining/reinforcing the channels of interaction among agents on business, social and professional networks, or for supporting decision making in topology control protocol, w.r.t which connections should be preserved.

The rest of this paper is structured as follows. Section II presents the benchmark and the experiment protocol adopted to evaluate the network sensitivity to attacks and failures. Section III discusses the results of the new protocol to evaluate the role of community links in the network communication. Section IV presents the adaptive mechanisms proposed and their evaluated performances. Finally, Section V summarizes the conclusions and contributions, and point out some issues for future research.

II. Benchmark and protocol for network assessment

A combination of models and metrics provides the benchmark for assessing the exposure of complex networks to failures and attacks, and for supporting a targeted analysis. This section outlines the framework applied for carrying this analysis. The main components are the network models, the centrality measures, and the simulation protocol.

ISBN: 978-1-61208-401-5

A. Complex Network Models

The interaction among agents in a complex system tends to create efficient networks at global and local levels, often under a scale-free degree distribution. Based on this, many researchers have proposed different models to create networks with particular topological properties as convenient simulations of real networks. For our study we consider two of the most widely used models: the Barabasi and Albert's (BA) model [2] and the Klemm-Euguluz (KE) model [17], referred from this point on as BA networks and KE networks, respectively.

Table I presents the models main topological properties values: the number of network nodes (n) and edges (|E|), the average degree (⟨k⟩), and the global (E_{glob}) and local (E_{loc}) efficiencies - see Section II-D for technical details on how the efficiencies are computed.

TABLE I. TOPOLOGICAL PROPERTIES OF BA AND KE NETWORKS

Network	n	$\lvert E \rvert$	$\langle k \rangle$	E_{glob}	E_{loc}
BA	1000	5979	11.95	0.37	0.047
KE	1000	5973	11.94	0.30	0.60

B. Real Network Datasets

Real datasets were considered for the experimental analysis. Some of them are classical benchmarks for studies in community-related approaches, the others are classical datasets in the complex networks literature, in general. The datasets and their main topological properties are shown in Table II.

TABLE II. DESCRIPTION AND PROPERTIES OF REAL NETWORK DATASETS

Dataset	n	$\lvert E \rvert$	$\langle k \rangle$	E_{glob}	E_{loc}
The US heaviest traffic airports [18]	500	2980	11.92	0.37	0.62
The protein interaction of yeast [1]	417	511	2.45	0.19	0.05
American College football	115	613	10.6	0.45	0.40
Dolphins	62	159	5.13	0.37	0.26

C. Failures and Attacks Protocols

This work is based on the assumption that community structure can be worthwhile to support the evaluation and mitigation of vulnerable topological states. Thus, the network target of analysis has its nodes classified according to the community they belong. The approach to find and update communities is presented in [19].

For simulating failures, nodes are considered autonomous agents that can leave the network at random with a uniform probability distribution. On the other hand, to reproduce a possible scenario of attacks, central nodes must be removed from the network. There are several criteria to rank nodes according to their positions in the network, in general, the Betweenness Centrality (BC) has been considered as a convenient measure of centrality w.r.t. communication. BC establishes higher scores for nodes that are contained in most of the shortest paths between every pair of nodes in the network. In fact, considering communication networks, nodes with this feature are likely to be crucial to maintaining the network functionality.

For a given node i and a pair of nodes j, l, the importance of i as a mediator of the communication between g and l can be established as the ratio between the number of shortest paths linking nodes j, l which passes through node i ($g_{jl}(i)$), and the total number of shortest paths connecting nodes j and l (g_{jl}). Then, the BC of a node i is simply the sum of this value over all pairs of nodes, not including i [20]:

$$BC(i) = \sum_{j<l} (g_{jl}(i)/g_{jl}). \tag{1}$$

For assessing the relevance of community structure information to detect and mitigate vulnerable topological network configurations, a protocol for attacks and failures concerning the role of central nodes in the community structures were applied. It encompasses: 1) ranking nodes according to BC or random criteria; 2) removing links of the most central node from the network considering its role in the node community: inside, outside or both; and 3) computing the target properties values. At each iteration, the node ranking is updated until a previously defined fraction (f) of nodes become disconnected from the network.

The adaptive mechanism must compensate the central node failures with addition of new links. Thus, for its performance evaluation the most central nodes are completely removed from the network. For validation purposes, three heuristics were defined considering the constraints of creating new links according to their roles: *inside* or *outside* the community, or *both*. For model-based networks, the results were averaged over five realizations.

D. Evaluation Mechanisms

A network is modeled as a graph $G = (N, E)$ defined by a set of nodes (or vertices) $N = 1, 2, ..., n$ and a set of links (or edges) $E \subseteq NXN$. A connection between vertices may be absent when there is no direct relationship or communication between them, or it may assume a value in $[0, 1]$ representing the strength (weight) of the connection. Only undirected and unweighed networks are considered here.

The impact assessment is supported by classical topological metrics related to the most important topological features found in real networks, as follows.

1) Global Efficiency: Latora ***et al.*** [21][22] introduced a measure of efficiency which computes how efficiently nodes exchange information either in a local or global scope, independently of whether the network is weighted or unweighted, connected or disconnected. For a given pair of nodes (i, j), its contribution to the global efficiency is inversely proportional to the shortest distance between them (d_{ij}), therefore $e_{ij} = \frac{1}{d_{ij}}$.

The global efficiency $\mathcal{E}_{glob}(G)$ of a graph G can then be defined as:

$$\frac{\sum\limits_{i \neq j \in G} e_{ij}}{n(n-1)} = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}, \tag{2}$$

and therefore, $\mathcal{E}_{glob}(G) \geq 0$. From this point on, we normalize this measure, considering the ideal situation G_{ideal} where all the possible $n(n-1)/2$ edges are in the graph, this is the case when $\mathcal{E}_{glob}$ assumes its maximum value. Thus, the normalized efficiency is:

$$E_{glob}(G) = \frac{\mathcal{E}_{glob}(G)}{\mathcal{E}_{glob}(G_{ideal})}. \tag{3}$$

2) Local Efficiency: The local efficiency is defined as the ratio between the number of edges that actually exist among i's neighborhood (not including i itself) and the total number of possible links. If the nearest neighborhood of i is part of a clique, there are $k_i(k_i - 1)/2$ edges among the corresponding nodes, where k_i is the degree (number of links) of node i. Formally,

$$E_{loc}(G) = \frac{1}{n} \sum_{i \in G} E_{loc}(G_i), \tag{4}$$

where

$$E_{loc}(G_i) = \frac{1}{k_i(k_i - 1)} \sum_{l \neq m \in G_i} \frac{1}{d_{lm}}. \tag{5}$$

and G_i is the subgraph induced by the nodes directly connected to i.

3) Giant Component: In most real-world complex networks, it has been observed that there is a large connected component, often called giant component, together with a number of small components containing no more than a few percent of the nodes [23]. A connected component of a graph is a set of nodes such that a path exists between any pair of nodes in this set. Its analysis may provide valuable insights for quantitative analysis, for instance, on how information dissemination and percolation in Epidemiology-related systems are affected by the disconnection or loss of nodes [23]–[29].

Notice that the size of the largest connected component is often equated with the graph-theoretical concept of the "giant component", although technically the two are the same only in the limit of large graph sizes [4]. For the sake of simplicity, we adopt herein the denomination "giant component" whenever we refer to the largest component. As a matter of fact, the connectivity of a network G can be estimated by the relative size $S(G)$ of the giant component, given by the fraction of nodes in the network taking part in the largest connected component:

$$S(G) = \frac{n_{Giant}}{n}, \tag{6}$$

where n_{Giant} is the number of nodes in the giant component and n is the number of nodes in the network.

III. Community-based network robustness

For assessing the role that central elements play in the community structure concerning robustness to failures and attacks, the protocol for link removals (see Subsection II-C) was applied. The results are depicted using blue, red, and green lines representing the removal of node's link(s) according to inside, outside and both (inside and outside) criteria, respectively.

Figures 1 to 4 show the evolution of global efficiency and the giant component (y-axis) during the process of attacks, represented by the fraction of nodes removed from the network (x-axis). The results stress the importance of links between communities, emphasizing that losing channels of communication between communities may be potentially harmful to the network connectivity. It means that those nodes responsible for linking communities may be the key elements for evaluating and mitigating topological states of vulnerability.

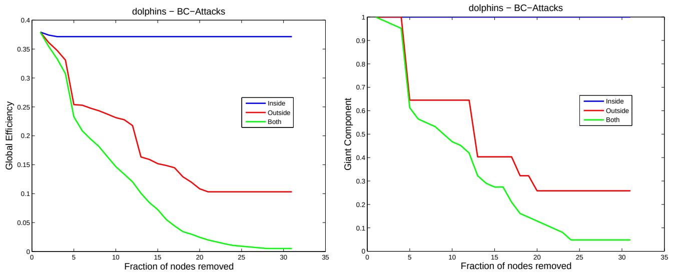

Figure 1. Global efficiency and giant component – BC attacks for dolphins network.

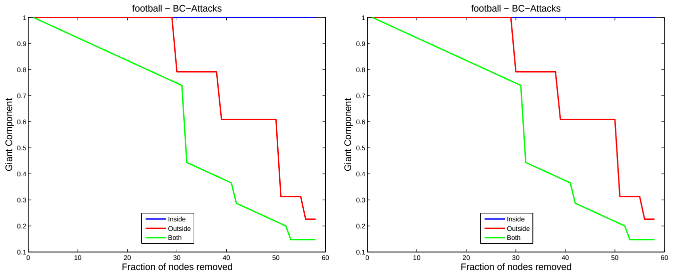

Figure 2. Local efficiency and giant component — BC attacks for football network.

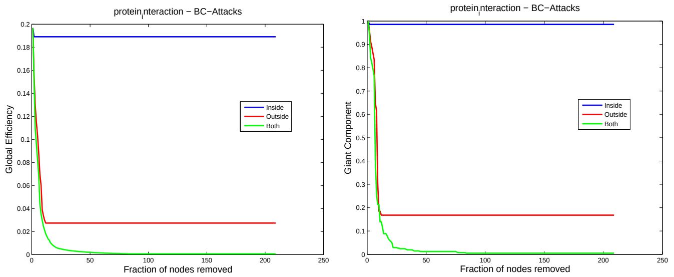

Figure 3. Global efficiency and giant component — BC attacks for Protein Interaction network.

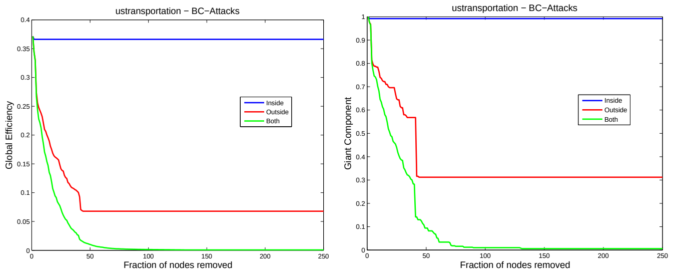

Figure 4. Global efficiency and giant component — BC attacks for UStransportation network.

Figure 5 illustrates the classification of links at each network state during the perturbation process considering the removal of both links (inside and outside) for the Dolphins network. The inter-communities bars represent the fraction of nodes that are connecting nodes from different communities. In turn, the intra-community bars represent those links that are connecting nodes belonging to the same community. They are computed taking into account the entire network (on the left) and the links that were removed from the network (on the right). Notice that at the beginning of the perturbation process, despite the fraction of intra-communities links considering the entire network is around 0.3, they were the majority of links lost. Furthermore, they were those which more severely affect the network connectivity (see Figure 1), highlighting the importance of these links to the network connectivity.

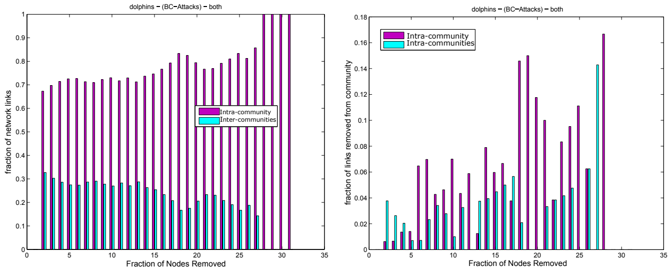

Figure 5. Link statistics — BC attacks for Dolphin network.

Figure 6 shows the results for BA and KE networks.

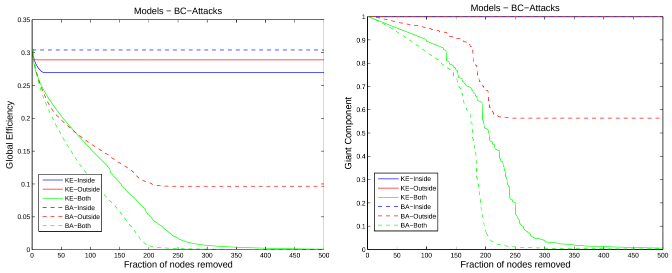

Figure 6. Global efficiency and giant component — BC attacks for BA and KE networks.

The property values of BA networks (dashed lines) showed the same pattern that real network topologies. On the other hand, for the KE networks the removal of both types of links were necessary to actually impact the network connectivity, thus deviating from most of the results achieved for inside and outside strategies, even for some other real networks not presented in this paper.

IV. Adaptive Mechanism

The approach proposed here is based on the previous work presented in [15]. It considers nodes as the main agents for controlling the necessary information and procedures responsible for evaluating and promoting changes in the network topology. For that, some nodes are assumed to be more likely to affect the network connectivity according to some likelihood status, and thereby, nodes in their neighborhood can be defined as in a vulnerable state. Here, the unit of analysis changes from nodes to a higher hierarchic structure according to the graph partition generated by the community detection technique [19]. A partition P is a division of a graph into clusters, such that each vertex is assigned to one and only one cluster. Even though some approaches consider vertices belonging to two or more clusters simultaneously [30], here each node takes part of a single community/cluster.

A community C is classified as in a vulnerable state if some specific property value is lower than expected. Two states are thus defined: vulnerable ($V_{C,t} = 1$) and not vulnerable ($V_{C,t} = 0$), according to:

$$V_{C,t} = \begin{cases} 1 & \text{if } \delta_{C,t} >= \gamma \\ 0 & \text{if } \delta_{C,t} < \gamma \end{cases}$$

where $\delta_{C,t}$ represents the target property value for community C at a specific time t. The threshold to set a community as vulnerable is given by the parameter γ. Both the target property and the vulnerability threshold can be set out as convenient for the vulnerability problem being handled.

The adaptation process is straightforward. It encompasses two main functionalities: the vulnerability assessment and the creation of new links. In compliance with the attacks and failures protocol, after each node removal, every community C assesses its vulnerability state. If applicable (*i.e*, when $\delta_{C,t} < \gamma$) new links are added in the network to try to reverse or minimize the adverse effects of the resulting topological configurations. For validation purposes three strategies were implemented:

- *inside*: adding connections between nodes belonging to the same community,
- *outside*: creating link(s) between node(s) from the vulnerable community to other(s) neighboring community(ies),
- *both*: the combination of inside and outside strategies.

The criteria for the definition of new connections are tied to the vulnerability property. According to the results discussed in [15], the local efficiency is a potentially good estimator for detecting and mitigating vulnerable states. Consider then the concept of local efficiency (5) at the community level:

$$E_{loc}(C_i) = \frac{1}{k_{in}(C_i)(k_{in}(C_i) - 1)} \sum_{l \neq m \in C_i} \frac{1}{d_{lm}}. \qquad (7)$$

where $k_{in}(C_i)$ is the number of nodes belonging to community C_i.

For new inside links, the non-connected nodes exhibiting the lowest and the highest local efficiency are connected. As the probability of sharing common neighbors is higher inside the community, this new connection tends to enhance the local community robustness.

The *outside* strategy considers that each vulnerable community (source community) should reinforce its connection with the neighboring communities with which it is weakly connected. Considering C as the set of communities in G and C_i the set of nodes belonging to community i, the neighboring of community C_i is $N(C_i) = \{(C_j \in C | e_{v,u} \in E \wedge v \in C_i \wedge u \in C_j\}$ and $k_{out}(C_{i,j})$ the number of times a community C_j appears in $N(C_i)$. For a vulnerable community C_i, the lowest community degree value $min(k_{out}(C_{i,j}) | C_j \in N(C_i))$ is the threshold to define the neighbor community(ies) to create a connection. It means that those neighboring communities with fewer connections are the targets for new connections, thus creating an alternative path between them.

The strategy to identify which nodes will receive new connections in both source and target communities is the same: the priority is for choosing nodes without any link with other communities. In the case of absence of nodes showing this feature, those nodes without connections with the target community are selected.

The *both* strategy combine the inside and outside procedures.

A. Results

For performance evaluation, the vulnerability threshold was set to $\gamma = E_{loc}(G) * 0.5$. This definition relies on the assumption that communities with local efficiency below the network local efficiency (see (4)) are more likely to be vulnerable.

Figures 7 to 12 present the adaptive mechanisms performance. Each line shows the evolution of global efficiency (on the left) and size of the giant component (on the right) during the process of attacks regarding different adaptation strategies: H is the original heuristic [15], $E_{loc}(outside)$, $E_{loc}(inside)$ and $E_{loc}(both)$ are for the *outside*, *inside* and *both* strategies, respectively. For benchmarking, G depicts networks without any running adaptive mechanism.

As expected, the improvements accomplished by the $E_{loc}(inside)$ strategy were irrelevant. The results for the original strategy demonstrate that its performance is related to the network local efficiency, mainly because the creation of links depends on the existence of non-vulnerable nodes. It means that vulnerable states can be detected, but the requirement to add links is not fulfilled. The evolution of both global efficiency and size of the giant component for Football, USTransportation and KE networks, which exhibit the higher scores for local efficiency (see Tables II and I), demonstrate that.

On the other hand, $E_{loc}(outside)$ and $E_{loc}(both)$ strategies produced significant results for all networks evaluated and were able to maintain the majority of nodes connected to the giant component. It is important to notice the influence of the initial network configuration regarding its sensitivity to attacks. For instance, the Football and Dolphins networks are less affected by attacks, so the adaptive community-based mechanisms were able to maintain the global efficiency and nodes in the giant component for most iterations, with the addition of a few links (see Figure 13). In turn, for more sensitive topologies, such as Protein Interaction and USTransportation networks, a small fraction of nodes was not able to be maintained in the giant component, despite the number of links created in the beginning of the adaptation process. Therefore, considering these networks sensitivity, the community-based heuristic improved the network robustness.

Figure 13 shows the proportion of new links created at each iteration. Notice that for the Football network a few nodes were added to the network considering the community-based heuristic. Regarding the Protein Interaction network, the proportion of new links for *outside* and *both* strategies at the beginning of process are around 0.40 of the total number of links in the network.

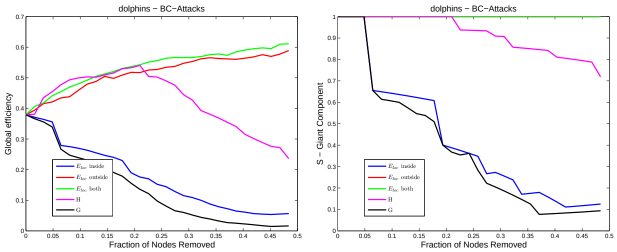

Figure 7. Global efficiency and giant component — Adaptation - Dolphin network.

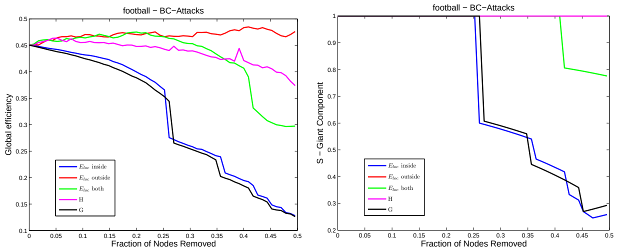

Figure 8. Global efficiency and giant component — Adaptation - Football network.

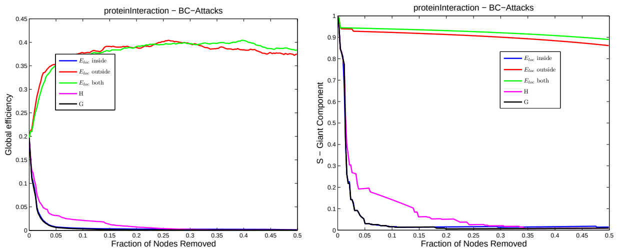

Figure 9. Global efficiency and giant component — Adaptation - Protein Interaction network.

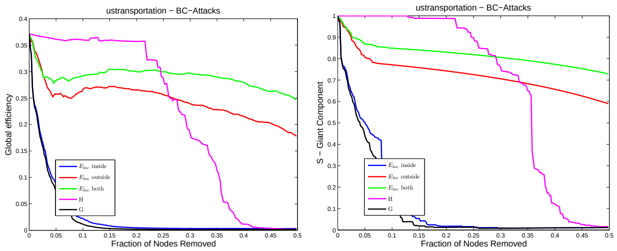

Figure 10. Global efficiency and giant component — Adaptation - USTransportation network.

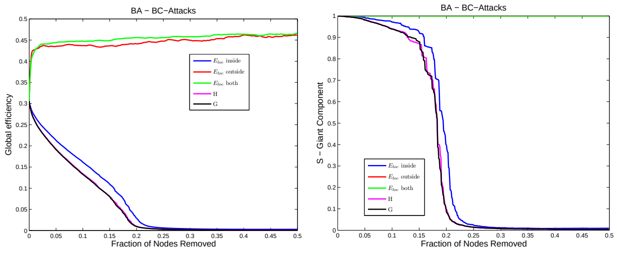

Figure 11. Global efficiency and giant component — Adaptation - BA networks.

ISBN: 978-1-61208-401-5

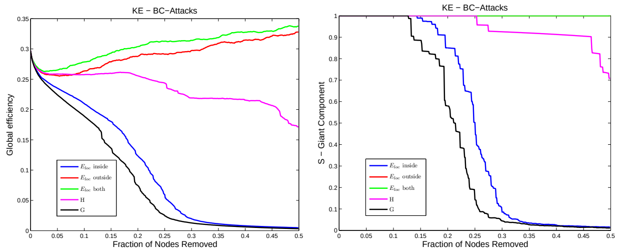

Figure 12. Global efficiency and giant component — Adaptation - KE network.

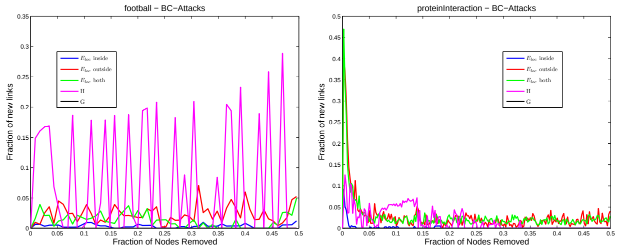

Figure 13. Global efficiency and giant component — Adaptation - USTransportation network.

As highlighted before, this network topology is quite sparse, exhibiting low scores for efficiencies and average degree, so it is necessary to create more links to provide a more robust network. However, after the initial adjustments, the network was able to accommodate perturbations and to maintain its global efficiency and the size of the giant component.

V. Conclusion

The first aspect highlighted here is that central nodes are probably those connecting communities and, therefore, information about the community structure can be worthwhile to design networks that are more resilient to failures and attacks. Taking this premise into account, community-based mechanisms to evaluate and mitigate vulnerable topological configurations were proposed in this paper. The solution comprises three main components: community identification, vulnerability detection and vulnerability mitigation. For the first component, a well-established method was applied [19]. For the second, a mechanism based on previous results from [15], but adapted to communities instead of nodes, was proposed. It considers as vulnerable those communities exhibiting local efficiency below the network local efficiency. Finally, the proposed heuristic to mitigate possibly vulnerable states relies on the creation of additional links between communities. For reinforcing the importance of the community structure, three different strategies were evaluated, considering creating links inside or outside the communities, or both. The $outside$ and $both$ community-based heuristics outperformed both the $inside$ community strategy and the original method based on node information. Furthermore, they showed less sensitivity to the network topological properties. Thus, the community-based heuristics showed to be a good prospect towards robust mechanisms to deal with the vulnerable topological configurations w.r.t. network robustness to attacks. Future works comprise the evaluation of local mechanisms for communities detection and parameter estimation, as well as the model validation considering larger datasets.

Acknowledgment

The authors would like to thank to FAPESP (procs. no.2012/25058-9, 2013/13447-3 and 2014/13800-8) for the financial support to carry out this research.

References

[1] A. Barabási and Z. Toroczkai, “Center for complex network research,” online, January 2010, network database. [Online]. Available: http://www.barabasilab.com/rs-netdb.php

[2] R. Albert and A. L. Barabási, “Statistical mechanics of complex networks,” Rev. Mod. Phys., vol. 74, no. 1, Jan. 2002, pp. 47–97. [Online]. Available: http://link.aps.org/doi/10.1103/RevModPhys.74.47

[3] E. Almaas and A. L. Barabasi, “Power laws in biological networks,” in Power laws, scale-free networks and genome biology, E. V. Kroonin, Y. I. Wolf, and G. P. Karev, Eds. Springer Science, 2006, pp. 1–11.

[4] M. E. J. Newman, “The structure and function of complex networks,” SIAM Review, vol. 45, no. 2, 2003, pp. 167–256. [Online]. Available: http://link.aip.org/link/?SIR/45/167/1

[5] M. Marchiori and V. Latora, “Harmony in the small-world,” PHYSICA A, vol. 285, 2000, p. 539.

[6] R. Albert, H. Jeong, and A.-L. Barabasi, “Error and attack tolerance of complex networks,” Nature, vol. 406, no. 6794, July 2000, pp. 378–382.

[7] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, “Efficiency of scale-free networks: error and attack tolerance,” Physica A: Statistical Mechanics and its Applications, vol. 320, Mar 2003, pp. 622–642.

[8] L. Dall’Asta, A. Barrat, M. Barthelemy, and A. Vespignani, “Vulnerability of weighted networks,” Journal of Statistical Mechanics: Theory and Experiment, vol. April 2006, 2006, p. P04006.

[9] M. Kurant, P. Thiran, and P. Hagmann, “Error and Attack Tolerance of Layered Complex Networks,” Phys. Rev. E, vol. 76, no. 026103, 2007, p. 026103.

[10] C. Ghedini and C. H. C. Ribeiro, “Rethinking failure and attack tolerance assessment in complex networks,” Physica A: Statistical Mechanics and its Applications, vol. 390, no. 23–24, November 2011, pp. 4684–4691.

[11] Z. He, S. Liu, and M. Zhan, “Dynamical robustness analysis of weighted complex networks,” Physica A: Statistical Mechanics and its Applications, vol. 392, no. 18, 2013, pp. 4181 – 4191. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037843711300410X

[12] J. Wang, C. Jiang, and J. Qian, “Robustness of interdependent networks with different link patterns against cascading failures,” Physica A: Statistical Mechanics and its Applications, vol. 393, no. 0, 2014, pp. 535 – 541. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437113007607

[13] M. Manzano, E. Calle, V. Torres-Padrosa, J. Segovia, and D. Harle, “Endurance: A new robustness measure for complex networks under multiple failure scenarios,” Computer Networks, vol. 57, no. 17, 2013, pp. 3641 – 3653. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613002740

[14] X. Wang, E. Pournaras, R. Kooij, and P. Van Mieghem, “Improving robustness of complex networks via the effective graph resistance,” The European Physical Journal B, vol. 87, no. 9, 2014. [Online]. Available: http://dx.doi.org/10.1140/epjb/e2014-50276-0

[15] C. Ghedini and C. H. C. Ribeiro, “Improving resilience of complex networks facing attacks and failures through adaptive mechanisms,” Advances in Complex Systems, vol. 17, no. 02, 2014, p. 1450009.

[16] Y. Yang, Z. Li, Y. Chen, X. Zhang, and S. Wang, “Improving the robustness of complex networks with preserving community structure,” PLoS ONE, vol. 10, no. 2, 02 2015, p. e0116551. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0116551

[17] K. Klemm and V. M. Eguíluz, “Growing scale-free networks with small-world behavior,” Physical Review E, vol. 65, no. 5, 2002, p. 057102.

[18] V. Colizza, R. Pastor-Satorras, and A. Vespignani, “Reaction-diffusion processes and metapopulation models in heterogeneous networks,” Nature Physics, no. 3, 2007, pp. 276–282.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, 2008, p. P10008. [Online]. Available: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008

[20] S. Wasserman, K. Faust, and D. Iacobucci, Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press, November 1994.

[21] V. Latora and M. Marchiori, “Economic small-world behavior in weighted networks,” The European Physical Journal B - Condensed Matter and Complex Systems, vol. 32, 2003, pp. 249–263.

[22] ——, “Efficient behavior of small-world networks,” Phys. Rev. Lett., vol. 87, Oct 2001, p. 198701. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.87.198701

[23] P.-Y. Chen and K.-C. Chen, “Information epidemics in complex networks with opportunistic links and dynamic topology,” in Global Telecommunications Conference (GLOBECOM). IEEE, Dec 2010, pp. 1–6.

[24] J. Wu, H. Z. Deng, Y. J. Tan, and D. Z. Zhu, “Vulnerability of complex networks under intentional attack with incomplete information,” Journal of Physics A: Mathematical and Theoretical, vol. 40, no. 11, 2007, pp. 2665–2671.

[25] M. Latapy and C. Magnien, “Complex network measurements: Estimating the relevance of observed properties,” INFOCOM. The 27th Conference on Computer Communications, April 2008, pp. 1660 – 1668.

[26] A. E. Motter and Y.-C. Lai, “Cascade-based attacks on complex networks,” Phys. Rev. E, vol. 66, Dec 2002, p. 065102.

[27] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Network robustness and fragility: Percolation on random graphs,” Physical Review Letters, vol. 85, no. 25, Dec. 2000, pp. 5468–5471.

[28] R. Cohen, K. Erez, D. B. Avraham, and S. Havlin, “Resilience of the Internet to Random Breakdowns,” Physical Review Letters, vol. 85, no. 21, Nov. 2000, pp. 4626–4628.

[29] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, “Giant strongly connected component of directed networks,” Phys. Rev. E, vol. 64, no. 2, 2001, p. 025101.

[30] S. Fortunato, “Community detection in graphs,” Physics Reports, no. 3-5, pp. 75 – 174.

Estimation of Job Execution Time in MapReduce Framework over GPU clusters

Yang Hung, Sheng-Tzong Cheng
Computer science and Information Engineering
National Cheng Kung University
Tainan, Taiwan
e-mail: stcheng@mail.ncku.edu.tw

Chia-Mei Chen
Information Management
National Sun Yat-Sen University
Kaohsuing, Taiwan
e-mail: cmchen@mail.nsysu.edu.tw

***Abstract*—The development of Graphic Processing Unit (GPU) makes it possible to put hundreds of cores in one processor. It starts a new direction of high-speed computing. In addition, in a Cloud computing environment, MapReduce over GPU clusters can execute graphics processing or general-purpose applications even much faster. It is crucial to manage the resources and parameters on GPU devices. In this paper, we study the execution time of MapReduce tasks over GPU clusters. We use Stochastic Petri Net to analyze the influence of GPU computing and develop SPN-GC model. The model defines formulas of every stage's execution time and estimates the execution time under different input data size. Our experimental result presents the comparison between the estimated execution time and actual values under different input data size. The error range is found out to be within 10%. This paper can be a useful reference when a developer is tuning the program.**

Keywords-GPU; MapReduce; Stochastic Petri Net; Estimation of execution time.

I. INTRODUCTION

As data is growing at an incredible speed, it is not easy to handle a huge amount of data to make timely decisions. The scale and variety of big data now challenges traditional computing paradigms. That is why Cloud Computing could come to help.

MapReduce is being considered as a programming model for large-scale parallel processing and an associated implementation for processing and generating large data sets. The benefits of this model include efficient resource utilization, improved performance, and ease-of-use via automatic resource scheduling, allocation, and data management. In addition, Graphic Processing Unit (GPU) was originally designed for graphics processing. But now, because of its parallel computing ability, GPU is more popular for scientific and engineering application.

Many algorithms and applications can be speed up by MapReduce, with the power of GPU computing [5]. To develop MapReduce over GPU clusters [6], programmers have to give thought to some performance-related issues. When running a MapReduce job, programmers cannot obtain any information about how the performance of jobs will be under their testing environment. In the past, programs were tuned by running a series of configurations based of past experiences and then waiting for jobs to complete for several times. If jobs' performance results can be estimated, programmers will be able to shorten their working hours by finding out the program's behavior in advance under a particular hardware specification and node configuration.

In Section II, Stochastic Petri Net [2] for MapReduce on GPU clusters model (SPN-GC) is developed to describe the detailed operations of MapReduce framework over GPU clusters [7]. SPN-GC estimates the execution time of MapReduce jobs with given parameters and returns the estimated results to programmers as a reference. In Section III, we validate the SPN-GC by running experiments. In Section IV, we conclude that programmers can use the proposed model to estimate and tune the programs in less time and with better performance.

II. SPN-GC: MODELING MAPREDUCE OVER GPU CLUSTERS

SPN-GC is divided into nine phases. Each phase demonstrates a specific operation in the MapReduce framework and is represented by the directed arcs along with the transitions from places to next places. Figure 4 illustrates a SPN-GC with three map functions and two reduce functions.

1) Load user program/Split input: When user starts an application, a job is assigned to Hadoop jobtracker and initialized. Jobtracker of the master node copies user program to each tasktracker on worker nodes. According to the user program and information, the jobtracker breaks input data into splits.

2) Read input: Every worker node runs its own tasktracker as a task manager. Tasktracker first reads input splits as input data of map function. Hadoop Disctributed File System (HDFS) always distributes input data over all nodes. In general, the map function will process the part of input splits that is stored on local disks for optimization. If the input split is on a remote server, a network transmission is initialized and the received data are stored, either temporarily in memory or on the disk if the data are too large, until the map function is completed.

3) Map function M/Read into GPU device: Map function of the user application takes over Compute Unified Device Architecture (CUDA) kernel function is initialized because CPU instructs the process to GPU, and data are copied from main memory to GPU device memory. Let the number of map tasks to be M. Note that map functions might not complete at the same time because every work node has its own processing speed and different delay time.

4) GPU computing: CUDA performs data high-speed parallel computing on GPU at this phase. Every stream processor is assigned to a thread. By Nvidia, 32 threads is called a warp. Threads read and write data from shared memory on GPU device at the same speed of cache.

5) GPU device to host: After CUDA finishes the work, data are copied from GPU device to host memory for the next process.

6) Sort/ Spill: The map function finishes an input pair and has to deal with the outputs from the map function. The outputs are called intermediate files in the MapReduce process.

7) Transmit/ Shuffle: After the first map task is finished, nodes of map function start to transmit intermediate files to nodes of reduce functions either locally or to remote nodes. Intermediate files are processed and sorted into the final output file for the use of the reduce tasks later. The final output file is separated into R partitions by a collector, and each partition is transmitted to the corresponding worker node that handles reduce tasks.

8) Sort/ Merge: All the data for the reduce function are pre-processing. Partitions are collected and sorted as in the map phase. After partitions are downloaded, sorted, and merged concurrently, a temporary file is prepared as the input data for reduce function.

9) Reduce function R: The reduce function is defined by the application requirements. According to the number of reduce tasks set by the user, R reduce functions should be executed in parallel, although all reduce functions may not start and end together because of varying processing speeds. All phases finish when the reduce tasks are completed.

A. Model Formulation

Let M to be the number of actual map tasks that is determined by the split size in Hadoop, and R to be the number of reduce tasks that can be configured directly in user program. The SPN-GC model can be defined as a marked stochastic Petri Net that is a 6-tuple (P, T, I, O, M_0, L), where $P = \{p_1, p_2, \dots, p_{n_p}\}$ is a finite set of n_p places, and

$$n_p = 5 * M + 3 * R + 2. \tag{1}$$

$T = \{t_1, t_2, \dots, t_{n_t}\}$ is a finite set of n_t transitions, where

$$n_t = 5 * M + 2 * R + 2. \tag{2}$$

$I \subseteq \{P \times T\}$ is a set of input arcs (flow relation), $O \subseteq \{T \times P\}$ is a set of output arcs (flow relation), and $M_0 = \{m_1, m_2, \dots, m_{n_p}\}$ is the set of initial markings where the generic entry m_i is the number of tokens in place p_i, $\mathrm{L} = \{\lambda_1, \lambda_2, \dots, \lambda_{n_t}\}$ is an array of firing rates where λ_j is the firing rate associated with each transition t_j. In SPN, each transition is associated with a random variable with exponential distribution that indicates the delay from the enabling to the firing of the transition.

There are many selections for firing rate which produces the elapsed time at each stage. In this paper, delay on timed transitions takes exponential distribution to describe the occurrence of events as a Poisson process. The exponential probability density function is defined in (3), where λ is the rate parameter of the distribution,

$$f_X(x) = \lambda e^{-\lambda x},\ x \geq 0. \tag{3}$$

The mean or the expected value of an exponentially distributed random variable X for a timed transition is given by (4), and can also be represented as the mean delay time in the set T on each timed transition, T_i .

$$E[X] = 1/\lambda. \tag{4}$$

After computing the mean delay time of each transition, the inverse can be obtained as a set of firing rates $L = \{\lambda_1, \lambda_2, \dots, \lambda_{n_t}\}$ where $\lambda_i = 1/T_i$, $\mathrm{i} = 1,2, \dots, n_t$, and the random time delay can be generated following an exponential distribution.

A big difference between MapReduce application and CUDA application is in the splitting of the input data. Programmers can select how to cut their input data in different sizes or in different ways depending on purpose for the input data. In Hadoop, programmers can set the block size by configuration. When the job is initialed, input of every task will fit the block size based on configuration. Different block size may produce unequal map task stages and result in various performance. It is similar in CUDA that programmers also have to adjust the input size of threads for their specific purpose or algorithm.

B. Model Analysis

The number of map tasks, *M*, is related to the number of CPU cores on each server which runs tasktracker. According to the configuration of Hadoop, the number of map tasks, *M*, is split by the size of blocks, as (5) shows below.

$$M = \lceil \text{Input data size } (D_{input})/\text{split size}(D_{split}) \rceil. \tag{5}$$

Data in HDFS is stored on a data node in which a tasktracker resides. Depending on data locations, the speed of accessing data locally or to a remote data node might be different. In our work, a random variable takes into account the ratio of the number of replications to the number of data nodes. For each map task *j*, a random rate of map input on local disk, *A_j*, is defined as below.

$$A_j = \frac{No.of\ replications}{No.of\ data\ nodes}, \text{ where } j = 1,2, \dots, M;\ 0 \leq A_j \leq 1.$$

In order to estimate the total execution time for a MapReduce job over GPU clusters, we need to derive the execution time spent in each phase. Starting from phase 1 till phase 9, the estimated execution in each phase is studied.

ISBN: 978-1-61208-401-5

Phase 1 is for loading program and splitting input data. Therefore, T_{phase1}, the estimated execution time of phase 1, is derived in (6). After loading a program, SPN-GC estimates the time of data uploading to HDFS and splitting into blocks.

$$T_{Phase1} = T_{load} + T_{upload}. \tag{6}$$

Phase 2 is for reading input from local disk and remote disk via network. Map-worker node download split from the input split locations. Every split data must be read from disk of HDFS and transmitted to map-worker node and then written into host memory. Therefore, T_{Phase2} is the maximal downloading time of all the nodes that do the map tasks.

$$T_{Phase2} = \max_{1 \le j \le m} \{\frac{Dsplit}{Disk\_r}(1 - A_j) + \frac{Dsplit}{Dnetwork}(1 - A_j) + \frac{Dsplit}{Disk\_w}(1 - A_j)\} \tag{7}$$

Phase 3 is about Mapping to GPU. Tasktrackers start map functions and copy input data into GPU devices. The execution time of map function can be estimated by the rate of data split size, CPU capability, and the time to copy data into GPU memory.

$$T_{Phase3} = \frac{Dsplit}{Disk\_r} + \frac{Dsplit}{Mem\_r} + \frac{CPU_{test}}{CPU_i} * T_m + \frac{Dgpu\_block}{GPU\_Mem\_w} \tag{8}$$

Phase 4 accounts for GPU memory read and GPU computing. Input split data is being read into GPU device cache from GPU device memory. The execution time of GPU computing can be estimated by the rate of GPU block size and GPU capability.

$$T_{Phase4} = \max_{1 \le i \le n}(\frac{Dgpu\_block}{GPUmem\_r} + \frac{D_{gpu\_block}}{D_{test}} * \frac{GPUtest}{GPU_i} * T_{m\_GPU}),$$
$$\text{where } n = D_{split} \,/\, D_{gpu\_block}. \tag{9}$$

Phase 5 is for GPU device to host. After GPU computing, output data is copied from GPU device memory to Host memory. Key-value pair is ready to sort.

$$T_{Phase5} = \frac{Dsplit}{GPU\_Mem\_r} + \frac{Dsplit}{Disk\_w} + \frac{Dsplit}{Mem\_w}. \tag{10}$$

Phase 6 is Spill/ Merge. Spills generated either by the metadata buffer or by the sort buffer could reach a specific limit that can be evaluated. The metadata size is 16 bytes per key-value record, while $D_{metadata}$ is the metadata size for all records in the map tasks, which can be evaluated as $16 * D_{input}/(D_{test} * M) * Map_{spiller\_records}$ bytes. In addition, $D_{metabuffer}$ is equal to (sort.mb * sort.record.percent * sort.spill.percent).

D_{data} is all key-value data size of the map task. D_{data} is equal to $(D_{input}/D_{test)}) * (D_{map}/M) - D_{metadata}$.

$D_{databuffer}$ is equal to [sort.mb*(1-sort.record.percent)* sort.spill.percent]. While, D_{spill} is equal to $(D_{metadata} + D_{data})$, meaning that all data sizes must be processed in this phase.

$$\text{No. of spill}(= N_{spill}) = \max(\left\lceil\frac{D_{metadata}}{D_{metabuffer}}\right\rceil, \left\lceil\frac{D_{data}}{D_{databuffer}}\right\rceil).$$

$$\text{No. of merges } (= N_{merge}) = \left\lceil\frac{N_{spill}}{sort.factor}\right\rceil.$$

$$T_{Phase6} = N_{spill} * \left(\frac{D_{spill}}{MR_i} + \frac{D_{spill}}{HW_i}\right) + N_{merge} * D_{spill} * (\frac{1}{HR_i} + \frac{1}{HW_i} + \frac{1}{MR_i} + \frac{1}{MW_i}) \tag{11}$$

Phase 7 is used to transmit and shuffle. The product of $\frac{D_{input}}{D_{test*R}}$ and $D_{mapoutput}$ equals the estimated map output size serving to be the input data of each reduce-worker node. The input data of reduce-worker nodes can be stored on a local disk (the reduce-worker node also acts as a map-worker node) or on remote map-worker nodes that must transmit data through the network.

$$T_{Phase7} = \frac{D_{input}}{D_{test*R}} * D_{mapoutput} * (\frac{A_j}{HR_i} + \frac{1-A_j}{Network_i} + \frac{1-A_j}{HW_i}) \tag{12}$$

Phase 8: Sort/ Merge. The data size of reduce input is expressed as $D_{shuffle}$, which is $\frac{D_{input}}{D_{test}} * D_{mapoutput} * \frac{Mtest}{R}$. In shuffling, the downloaded map output is buffered in memory first. When memory buffer is filled at a certain level of usage, the data are written to the disk, as specified in spilling. The time to merge all reduce input data and sort them can be estimated as disk read-write and memory read-write on $D_{shuffle}$ data size. Therefore,

$$T_{Phase8} = D_{shuffle} * \left(\frac{1}{MR_i} + \frac{1}{MW_i}\right) + \left\lceil\frac{D_{shuffle}}{Heap_{red} * shuffle.buffer.percent * shuffle.merge.percent}\right\rceil * D_{shuffle} * (\frac{1}{HR_i} + \frac{1}{HW_i} + \frac{1}{MR_i} + \frac{1}{MW_i}) \tag{13}$$

The final phase, Phase 9 is about Reduce. In this phase, programs may use CUDA for GPU computing. Data in memory must be read first, then the elapsed time of reduce function is estimated by multiplying T_r with the rate of $D_{shuffle}$ and the test map output-data size,$D_{mapoutput}$. After read and reduce, the output of the reduce function is written into HDFS. Hence,

$$T_{Phase9} = D_{shuffle} * \frac{1}{MR_i} + \max_{1 \le i \le n}(\frac{Dgpu\_block}{GPU\_Mem\_w} + \frac{D_{gpu\_block}}{D_{test}} * \frac{GPUtest}{GPU_i} * T_{m\_GPU} + \frac{Dshuffle}{GPU\_Mem\_r}) + \frac{D_{shuffle}}{D_{mapoutput}} * \frac{CPU_{test}}{CPU_i} * T_r + \frac{D_{input}}{D_{test*R}} * D_{red} * \left(\frac{1}{HW_i} + \frac{1}{MR_i}\right) \tag{14}$$

C. Notations and Default Setting

Major notations used in this paper are summarized in Table I. Default values are suggested as well. Detailed description of each parameter could be found in [9]. System and application related notations used in this paper are summarized in Tables I and II, respectively. Default values are suggested as well. Detailed description of each parameter could be found in [9].

TABLE I. SYSTEM NOTATIONS AND SETTINGS

Job Phase		
Parameter	*Notation*	*Default*
Input split size	D_{split}	64 MB
No. of referred map tasks	M_ref	4
No. of reduce tasks	R	1
mapred.tasktracker.map.tasks.maximum	Max_map	4
Map Phase		
Parameter	*Default*	*Description*
sort.mb	100 (MB)	The amount of buffer space to use when sorting streams.
sort.spill.percent	0.8	The amount of sort buffer used before spilled to disk.
sort.record.percent	0.05	The amount of metadata buffer used in spilling.
sort.factor	10	The number of map output partitions to merge at a time.
Reduce Phase		
Parameter	*Default*	*Description*
Max Heap size of reduce task ($Heap_{red}$)	1024 (MB)	Max heap size that can be used by reduce task.
parallel_copies	5	The number of map output partitions to merge at a time.
shuffle.buffer.percent	0.7	The amount of buffer space to use when sorting streams.
shuffle.merge.percent	0.66	The amount of the sort buffer used before data spill to disk.

III. VALIDATION OF SPN-GC

Platform-Independent Petri Net Editor 2(PIPE2) v4.3.0 [3] is an open-source tool that supports the design and analysis of Stochastic Petri Net models. PIPE2 uses the “xml” format that is easy to describe the form of a Petri Net. The SPN-GC is validated to conform the regulations of Stochastic Petri Net by PIPE2. Our SPN-GC simulator is constructed using Java and is expected to be released as a package in PIPE2 soon.

A. Simulation Settings

In our experimental environment, we run Hadoop-1.2.1 as MapReduce framework [1] of four worker nodes with one physical server each. Each physical server is equipped with GPU of model NVIDIA Tesla C2050 [4]. The details of hardware specification and software version can be found in [9].

The speed of memory is measured by using the “dmidecode” command on Linux. The system measures the elapsed time when creating 100 of 1000 bytes blocks to read in and read out. The speed of disk is measured in the same way. GPU device information can be queried from NVIDIA system management interface.

Three GPU computing benchmarks are studied in our experiments: converting side-by-side (SbS) video to depth video, matrix multiplication, and K-mean clustering. Due to the paper length, here we present the experimental results of 3-D video case and K-means only. Interested readers are recommended to study further [8][9].

For 3-D video conversion, the program is to transfer side-by-side video into depth video based on tremendous graphic processing. Every frame in the SbS video is composed of two almost-identical pictures except a little different angle of camera view. The depth video can be played on a 3-D monitor and delivers the 3-D visual effect. The data set was collected from Internet and the file format is any SbS video of .mp4 file.

TABLE II. APPLICATION RELATED NOTATIONS

Parameter	**Notation**
Input data size of the estimated job	D_{input}
Input data size of the test small job	D_{test}
Program loading and data split time of the test small job	T_{load}
Exec. time of map function in the test small job	T_m
Exec. time of reduce function in the test small job	T_r
Size after executing map function of the test small job	D_{map}
Size after executing reduce function of the test small job	D_{red}
Size of map output which is equal to reduce shuffle bytes	$D_{mapoutput}$
Data size of GPU block bytes	D_{gpu_block}

We use these SbS videos to compare the accuracy between the actual job execution time and the estimated execution time by SPN-GC. Since most of the GPU computing applications are not utilizing the MapReduce framework yet, we need to port the program written in CUDA to MapReduce framework. This justifies our major contribution in the area. To collect data, ten sets of different data size are fed in to each program to test the performance under different input configurations.

The second experiment is about K-means clustering, which is a method of vector quantization, originally from signal processing. A popular cluster analysis in data mining, K-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The problem is known to be computationally difficult (i.e., NP-hard). In our experiment, the observations are generated by random variables using time values as seed.

Our SPN-GC simulator is following the nine-phase execution and calculating the mean delay time to estimate the mean elapsed time in each phase and return the total estimated execution time. Exponential distributed random variables are used. Every run of simulation was performed 2000 times. The average of all simulated values is returned as the estimated execution time for the program.

B. Experimental Results

Figure 1 shows the results of converting SbS videos into depth videos. The execution time of actual test and estimated time by SPN-GC for input video length from 30, 60, 90, 120, 150, 180, 210, 240, 270, 300 seconds are shown respectively. We can see that the average of estimated execution time and actual test values were very close.

Figure 2 illustrates the results of the K-Means execution time of actual test and estimated execution time by SPN-GC. The input data size ranges from 0.5, 1, 2, 3, 4, 5, 6, to 7 millions. Every block reads 20 thousands as input. The application randomly chooses 10 points as the centers, then partition *n* observations into 10 clusters with the nearest mean to the cluster. We can see that the estimated execution time of each input data is very close to the actual test value.

To evaluate the significance of SPN-GC model, we compare the time taken by SPN-GC and by actually executing the test job, Time Cost Ratio is calculated to reflect how much time saving is obtained by SPN-GC estimation. SPN-GC is proved to be able to save lots of time in cluster selection or performance tuning.

$$\text{Time Cost Ratio} = \frac{time\ spent\ by\ SPN{-}GC}{Actual\ job\ execution\ time} * 100\%$$

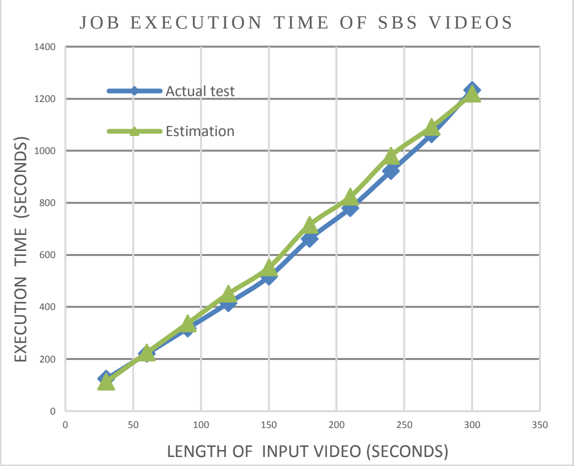

Figure 1. Job Execution Time of Side-by-side Videos.

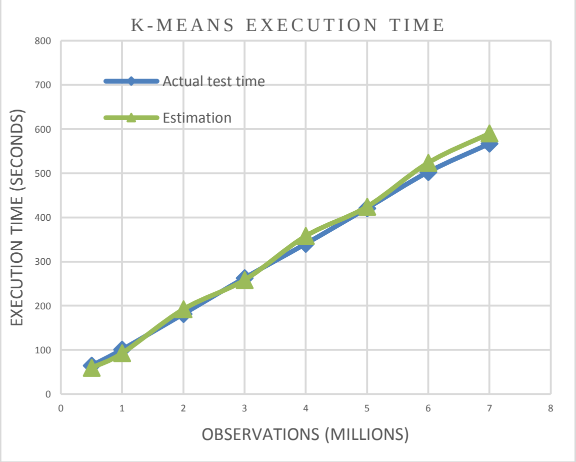

Figure 2. K-Means Execution Time.

Figure 3 draws the time cost ratio of SPN-GC to actual time cost for K-means clustering. The input range is from 0.5 to 8 millions of observations. We can see that, as the data size grows, the ratio drops to 0.7% approximately. It can be identified that SPN-GC would be able to perform an accurate estimation of the execution time for a job with a very small time cost. The benefit becomes more significant when the size of the problem increases.

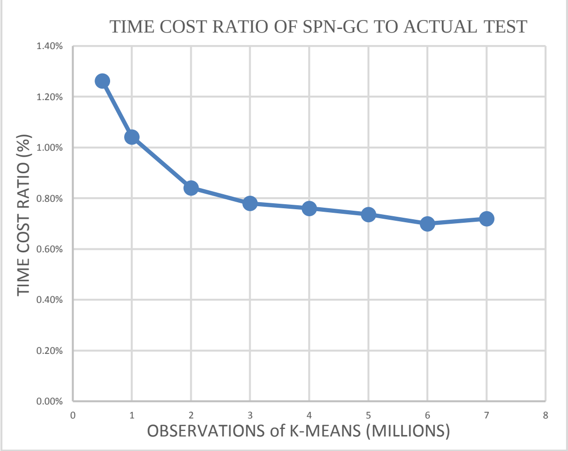

Figure 3. Ratio of SPN-GC Time Cost.

IV. CONCLUSIONS

In this paper, we develop the SPN-GC model to estimate the execution time of a job under the MapReduce framework over GPU clusters. Job execution time is an important performance indicator that provides crucial information for cluster evaluation. The considered environment is that input data are split into Hadoop block sizes and then spilt it again into the blocks for CUDA in GPU computing. There is also a problem that when GPU is computing graphic processing, every block gets different data and therefore each block has its own complexity. The data complexity and mean delay time are solved under the assumption of exponentially distributed random variables. In experimental results, SPN-GC is validated by PIPE2 and compared the estimation execution time with actual data test under three applications. The average error range of estimation execution time was found to be within 10%. SPN-GC can be a reference to evaluate GPU clusters performance.

ACKNOWLEDGMENT

This work is supported in part by Ministry of Science and Technology, Taiwan, R.O.C. under the grant number “102-2221-E-006-086-MY3” and by Ministry of Economic Affairs, Taiwan, R.O.C. under the grant number “103-EC-17-A-02-S1-201” respectively.

REFERENCES

[1] G. Wang, A. R. Butt, P. Pandey, and K. Gupta, “A Simulation Approach to Evaluating design decisions in MapReduce setups,” IEEE Symposium, Modeling, Analysis & Simulation of Computer and Telecommunication Systems, pp. 1-11, 2009.

[2] M. K. Molloy, “Performance Analysis Using Stochastic Petri Nets,” IEEE Transactions, Computers, vol. C-31, issue 9, pp. 913-917, 1982.

[3] N. J. Dingle, W. J. Knottenbelt, and T. Suto, "PIPE2: A Tool for the Performance Evaluation of Generalized Stochastic Petri Nets," ACM SIGMETRICS, Performance Evaluation Review, vol. 36, issue 4, pp. 34–39, 2009.

[4] E. Lindholm, J. Nickolls ,S. Oberman, and J. Montrym, "NVIDIA Tesla: A Unified Graphics and Computing Architecture," IEEE conference, pp. 39-55, 2008.

[5] Y. Guo, W. Liu, G. Voss, and W. Mueller-Wittig, "GCMR: A GPU Cluster-based MapReduce Framework for Large-scale Data Processing," IEEE conference, High Performance Computing and Communications & Embedded and Ubiquitous Computing, pp. 580-586, 2013, doi:10.1109/tencon.2013.6719008.

[6] J. A. Stuart and J. D. Owens, "Multi-GPU MapReduce on GPU Clusters," IEEE Intl. Symp. Parallel & Distributed Processing, pp.1068–1079, 2011,.

[7] H. Gao, J. Tang, and G. Wu, "A MapReduce Computing Framework Based on GPU Cluster," IEEE Conference, High Performance Computing and Communications & Embedded and Ubiquitous Computing, pp. 1902-1907, 2013.

[8] H. Wang, "Using Petri Net to Estimate Job Execution Time in MapReduce Model," master thesis, National Cheng Kung University, Taiwan, 2013.

[9] H. Yang, "Estimation of Job Execution Time in MapReduce on GPU Clusters," master thesis, National Cheng Kung University, Taiwan, 2014.

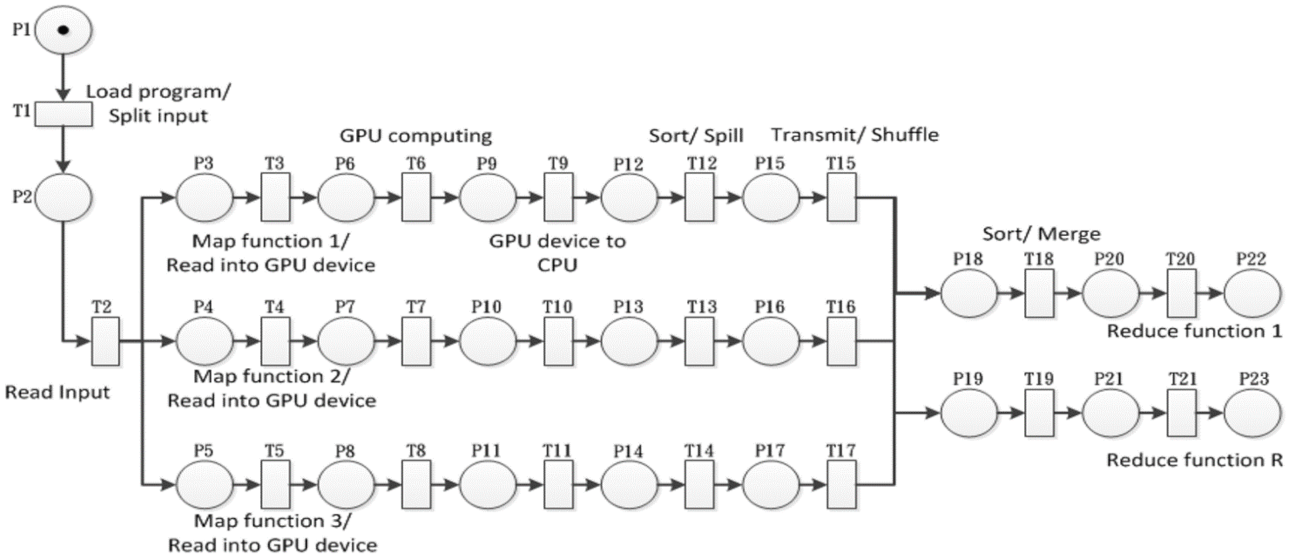

Figure 4. Nine phases of SPN-GC (with *M* =3, *R* =2).

Towards Assessing Visitor Engagement in Science Centres and Museums

Wolfgang Leister, Ingvar Tjøstheim, Trenton Schulz
Norsk Regnesentral
Oslo, Norway
{wolfgang.leister, ingvar.tjostheim, trenton.schulz}@nr.no

Göran Joryd
Expology
Oslo, Norway
goran@expology.no

***Abstract*—Currently, it is difficult to assess the engagement of visitors in science centres and museums for specific installations. We intend to measure how well individual installations work by using non-intrusive assessment technologies. This paper lays out the assessment framework for this goal. The article presents the Visitor Engagement Installation profile that characterises installations along six dimensions. An assessment framework that consists of four layers is presented and explained. First findings of the assessment of a selected installation are presented.**

***Keywords*—assessment; installations; science centres; museums; visitor engagement.**

I. Introduction

Science centres and museums present exhibitions, installations, and educational programmes that are supposed to engage visitors for self-education on a subject and to inspire the visitors to learn more. However, there is little data showing how well these installations perform regarding their goal to transfer knowledge to the visitors other than the use of longitudinal studies [1]. Similarly, there is little data to determine whether adjustments of installations have the wanted effect on a visitor's engagement.

The main objective of our work is to measure the performance of installations, but we cannot, in general, measure this directly. Instead, we assess the experience of visitors and groups of visitors while they use the installation and retrieve parameters and objective data from the installation and its context. We also want to avoid time-consuming observations by the museum staff and keep intrusive methodologies, such as questionnaires, to a minimum.

In our research, we argue that we can assess dimensions of engagement towards an installation by means of subjective assessment and automated observations of technical data from the installations, physiological data of the visitor, camera data, behaviour, etc. These data are used to estimate the performance of the installation, and whether adjustments of such installations contribute to a better engagement and experience.

First, we present an overview of related work, showing both the installation-centric and visitor-centric view of studies (Section II). Then, we show the approach of our proposed framework for assessing engagement (Section III). We present the Visitor Engagement Installation (VEI) profile to characterise installations using six dimensions (Section IV). An assessment of a selected installation follows (Section V). Finally, we present our conclusion (Section VI).

II. Related Work

Science centres are informal learning environments [2] that are distinct from classrooms because they offer free-choice learning [3][4], i.e., visitors can choose which activities to participate in and they can leave at any time.

Lindauer [5] presents a historical perspective of methodologies and philosophies of exhibit evaluations. Lindauer mentions only a few methods that perform measurements using simple metrics of counting or measuring time. In the literature, the majority of evaluations in science centres deals with the assessment of learning, often using a longitudinal approach, i.e., observing a subject or installation over time. Šuldová and Cimler [6] suggest that engagement can be assessed more instantaneously and be used as a part of learning assessment, supporting Sanford's [7] claim that "some compelling evidence links visitor engagement to learning".

We align the literature along two axes, as illustrated in Figure 1: the vertical axis denotes the span between longitudinal and instantaneous assessment; the horizontal axis denotes whether the assessment is *visitor* or *installation*-centric. In general, assessing an installation also needs to take an assessment of the visitor into account.

A. Visitor-Centric View

Dierking and Falk [8] present the Interactive Experience Model, which is a visitor-centric model. They define the interactive experience influenced by three contexts: *1*) the personal context, *2*) the physical context, and *3*) the social context. Falk and Storksdieck [9] use the principle of identity-related motivation that places visitors into five identity types: *1*) the explorer; *2*) the facilitator; *3*) the professional and hobbyist; *4*) the experience seeker; and *5*) the spiritual pilgrim. This line of visitor studies has been extensively studied [10][11].

Barriault and Pearson [12] present frameworks that analyse the learning experience more instantaneously by identifying learning-specific behaviour observed by cameras and microphones installed within an installation. Šuldová and Cimler [6] refine these methods, but still depend on manual analysis.

B. Installation-Centric View

In the installation-centric view, the science centre assesses installations rather than the visitors. The developers of installations need to consider the aspects of attractiveness, usability, being educational, etc. Young [13] suggests that developers need to be an advocate for the visitors and think as a visitor and recommends a cyclical development process. Allen [14]

ISBN: 978-1-61208-401-5

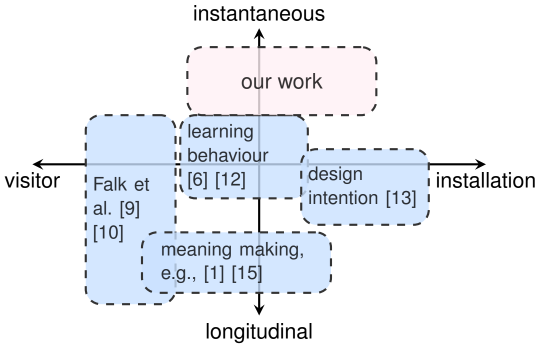

Figure 1: Classification of selected work in visitor studies

presents a study of three different versions of an exhibit for the purpose of studying dimensions of interactivity.

In longitudinal visitor studies, observations and sense-making [15] are often used. In sense-making, qualitative mental models, understanding events, and interpretation of situations in an iterative process (e.g., the data/frame theory of sensemaking [16][17]) are in the foreground whereas we are interested in concrete measurements and quantitative and descriptive data based on machine-retrievable data and questionnaires that allow us to get an instant result.

III. Approach

Installations in museums and science centres are complex systems that need to perform in their context together with the visitors. We take an installation-centric approach over a visitor-centric approach since we are interested in how the installations and potential changes of installations will perform. Also in the installation-centric view, it is important to observe visitors, study what they do, and determine whether the installations work as intended.

To assess the engagement for an installation, we developed an assessment framework that takes various types of data into account. While we are creating an estimation model, we need all available data. After we've created a suitable estimation model, our intention is to abstain from intrusive data collection as much as possible. We use a machine learning approach [18] to establish the model.

Developers and owners of installations are interested in what to change once an installation is assessed. This can be achieved by using the VEI profile presented in Section IV. The idea is to characterise an installation along six dimensions that one can adjust. Whether such adjustments are successful can be evaluated in a new assessment.

A. Assessment Framework

We propose an assessment framework that uses objective assessment, physiological responses, and estimation models to derive evidence of how a visit is perceived for individuals and groups of subjects.

An important requirement is that the assessment methods are not perceived as being intrusive. Intrusive assessment methods are usually only applicable in a lab setting, as they reduce the quality of experience (QoE) and, thus, impact the result of an assessment negatively.

Engagement and visitor experience cannot be measured directly. They are latent constructs. From measurable data and an estimation model trained by our machine learning approach we intend to derive a measure of experience of the visitors using an installation. It is similar to a satisfaction index and can be used to evaluate an installation.

B. The Layers of the Assessment Framework

Our assessment framework (Figure 2) consists of four layers: *Layer I*: the *Scenario Layer* presents the artefact, the subject, the action or interaction of the subject, other subjects, and, to some extent, observers; *Layer II*: the *Data Collection and Observer Layer* describes which data are collected from the elements of the scenario. *Layer III*: the *Assessment Layer* describes the types of assessment performed; and *Layer IV*: the *Assessment Process Layer* describes how the assessed data are processed further for the evaluated properties.

C. The Data Collection and Observer Layer

From a technical perspective, we classify whether these data in the Data Collection and Observer Layer (Layer II) as *1*) are automatically retrieved and processed, e.g., log files, technical parameters, event lists, sensor data, or physiological data; *2*) are data from surveys and questionnaires; these data are often coded and analysed after the visitors have left the site, and the answering process might be intrusive; *3*) are observations by an external observer; or *4*) are static data that are stored, available, or known, e.g., from databases, or historical data.

D. The Assessment Layer

For defining the categories used in the Assessment Layer (Layer III), we adapt the assessment categories presented by Leister and Tjøstheim [19] into the following components: *a*) *subjective assessment* based on questionnaires and ratings; *b*) *objective assessment* based on measurements at the object; *c*) *physiological assessment* based on sensor data from a subject; *d*) *behaviour and interaction assessment* based on observations of the subject and the subject's behaviour and interaction with both the object and other subjects; *e*) observation of the subject and interaction with other visitors; and *f*) objective and subjective context information, including visitor type.

E. The Assessment Process Layer

The Assessment Process Layer (Layer IV) describes how the data from the Assessment Layer are processed. In Figure 2, the impact of these data is shown with bold arrows. Additionally, values with dashed lines could be taken into consideration. Data that are visualised with dotted lines are used in the calibration process when creating the estimation model or for evaluation purposes. Most of these data cannot be automatically processed and need human intervention of some kind.

Layer IV contains the following elements:

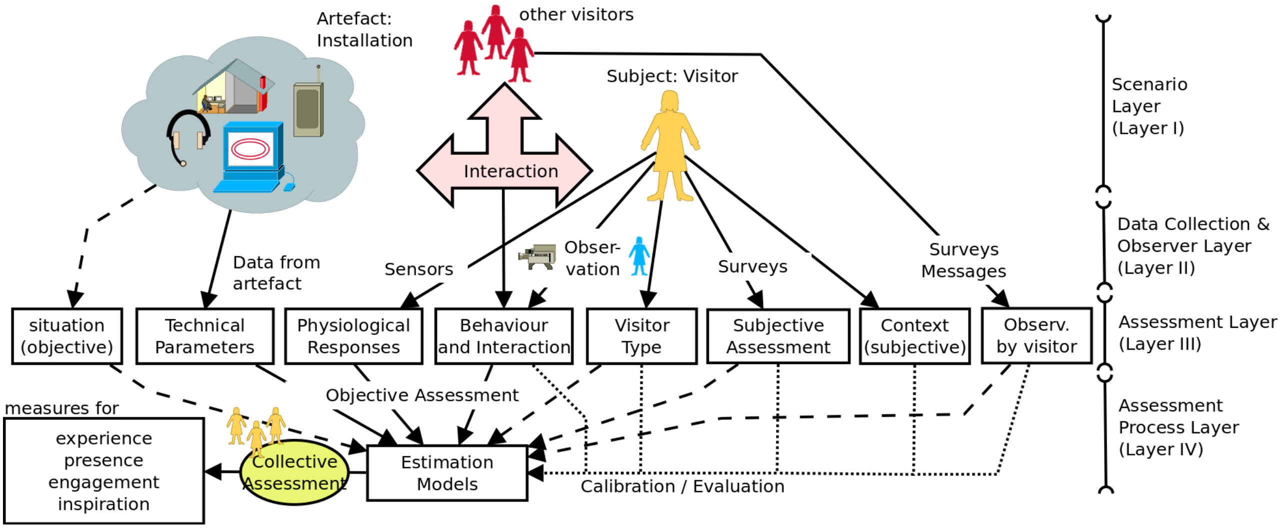

Figure 2: Four-layer assessment framework for engagement of visitors using installations in science centres and museums.

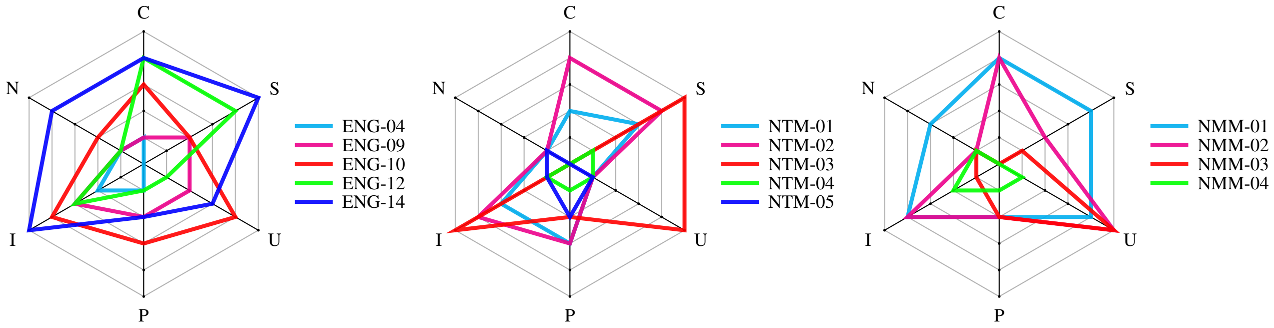

Figure 3: The VEI profile for selected installations in three science centres.

1) Estimation Model: The estimation model is a mathematical model that takes measurable assessment data as input and returns estimated values expressed in suitable metrics. The estimation model usually returns an estimated value for one subject at a time since personal data specific to the subject are involved in the calculation. Machine learning approaches [18] can be used to implement the estimation model.

2) Collective Assessment: Collective assessment presents the rating for one installation based on the individual assessments by many subjects.

3) Measures for evaluated properties: The result of the assessment process consists of measures for the evaluated properties. This can be a vector of values that will be used in the process that requires such assessment data.

IV. The Visitor Engagement Installation (VEI) Profile

To characterise installations, we developed the VEI profile in an iterative process with three sciences centres: the Engineerium (ENG), the Norwegian Museum of Science and Technology (NTM), and the Norwegian Maritime Museum (NMM).

Most studies that evaluate installations in science centres evaluate the impact of one dimension, such as interactivity, on the visitor. For this, observations of visitors are performed with various degrees of the dimension in question. However, we did not find a profile that characterises installations in multiple dimensions directly from an objective perspective, i.e., from only evaluating the installation.

The VEI profile was developed from a set of requirements for a well-working installation given by the participating science centres. From these requirements, we selected a set of dimensions that we considered sufficiently orthogonal and tried these on a set of fourteen selected installations (see Figure 3). We performed several iterations of this process until the requirements for common science centre installations were covered. We are aware that other dimensions could have been used. If necessary, our profile can be extended with more dimensions, such as immersion or degree of difficulty.

A. Defining the VEI Profile

The VEI profile classifies installations in their dimensions of competition (C), narrative (N), interaction (I), physical (P), visitor (user) control (U), and social (S). Each of these dimensions can have a value from 0 to 5; the higher the value, the more a dimension is present in an installation. TABLE I presents the description of the values for each dimension.

The dimensions of the VEI profile are described as follows:

TABLE I: EXPLANATION OF THE VALUES USED IN THE VEI PROFILE.

	0	1	2	3	4	5	
C	visitor observes only; no competition element.	inst. has several components; result must be achieved to proceed or succeed.	visitor receives a score; competition with the installation (machine).	competition with other visitors asynchroneously.	competition with other visitors in real-time.	challenge in team; influence on other players' result.	**C**
N	no narrative; object can only be observed.	installation is used in a specific sequence; chronological succession of events.	installation is built up in sequences; conditions must be met to proceed to next phase.	installation designed for multiple visitors; visitors may cooperate; multiple parallel narratives.	multi-player game or simulation; visitors cooperate to achieve a final result.	visitor develops narrative.	**N**
I	no interaction with object; observe only.	primarily no interaction; visitor can do something with the installation.	some interaction, such as “continue”, “stop”, “yes/no”; installation reacts.	moderate degree of interaction; choices influence outcome.	high degree of interaction; choices have consequences; content is stored.	visitor creates some of the content.	**I**
P	no physical activity; observation only.	push buttons; touch screen; hold or touch object.	visitor moves betw. parts of installation; enter installation; guided tour.	some activity, e.g., operating pumps; throwing balls.	full body-motion; longer physical activity.	full body motion over time; performing physical task in real setting.	**P**
U	controlled; visitor is observer; linear structure.	controlled with some degrees of freedom; mostly linear structure.	combination of controlled and free flow; choices can be made.	visitor can make choices; receives feedback on right or best choices.	visitor controls flow, but installation limits its choices.	visitor has high degree of control; creative process.	**U**
S	single visitor.	single visitor, others observe.	several installations used independently from each other.	single visitor while others observe and engage and cheer.	installation intended for several simultaneous visitors.	multi-visitor installation; visitors must cooperate.	**S**
	0	1	2	3	4	5	

1) Competition: the degree of competition in an installation.

2) Narrative: the degree of active participation in the underlying narrative.

3) Interaction: the degree of interaction between the visitor and the installation.

4) Physical: the degree of physical activity the visitor must perform when using the installation.

5) Visitor control: the degree a visitor can control the use of the installation.

6) Social: the degree of social interaction between visitors.

B. Applying the VEI Profile To Measure Engagement

We applied the VEI profile to installations from the three science centres: five at ENG, five at NTM, and four at NMM. The VEI profiles of these installations are shown in Figure 3.

We assessed installations with visitors. We wanted to determine whether a change in one dimension of the VEI profile from x to y will result in a change of the visitor's engagement. For example, the assumption that a change in an installation with a C-factor (competition) of 3 to 4 would increase the visitor engagement could be tested by measuring the visitor engagement with the originally designed installation, make changes in the installation to increase the C-factor (e.g., making the competition with other visitors happen in real-time), and then measure the visitor engagement for the altered installation. We are interested in the relative changes of the assessed engagement-related values when testing installations with modified versions that have a different VEI profile.

C. Characterising Exhibitions Using the VEI Profile

Besides single installations, the VEI profile can be used to characterise exhibitions or groups of installations. For example, the graphical representation of the VEI profile for selected installations in Figure 3 suggests that physical activity is characterised as low for these installations. Also the N-dimension seems to be low, with the exception of two recently developed installations that are based on longer narratives. We also observe differences between the three sites regarding their overall profile characterised by mean values and variance of the respective VEI profiles.

V. ASSESSMENT OF A SELECTED INSTALLATION

We are doing assessments to analyse the correlations between the various data in Layer III of our assessment framework. These assessments will be used to build the structure and parameters of the estimation models in Layer IV.

In the following, we present preliminary results of an assessment that has the assumption that the C-dimension of the VEI-profile has an impact. We compare subjective data of winners, losers, and single players of a quiz game.

A. Experiment Setup

The installation *Footprint eQuiz* at the Engineerium, here denoted as ENG-12, shall challenge the visitors with questions about different environmental perspectives, show how the oil and gas industry takes responsibility, and how they work to minimise the negative impact on the environment. The installation provides an understanding of different ways we can lower our energy consumption to reduce the environmental impact.

ENG-12 is a game where up to two players compete by answering questions related to energy and the environment. There are two levels available, beginner and expert. The installation consists of two stations with two large buttons each, an orange one and a blue one. ENG-12 starts with a short introduction before ten questions are shown on the screen in sequence. As a question is shown, a timer starts counting down

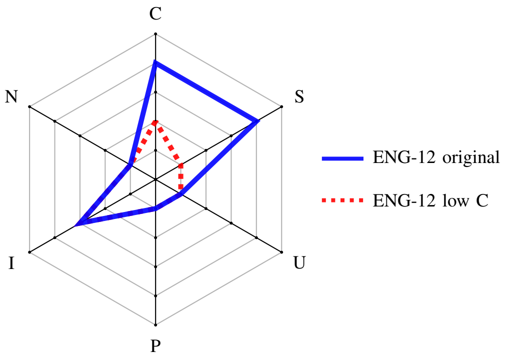

Figure 4: VEI profiles of the ENG-12 installation when two players compete (solid line) and a single player version (dashed line); the single player version has lower values for C and S.

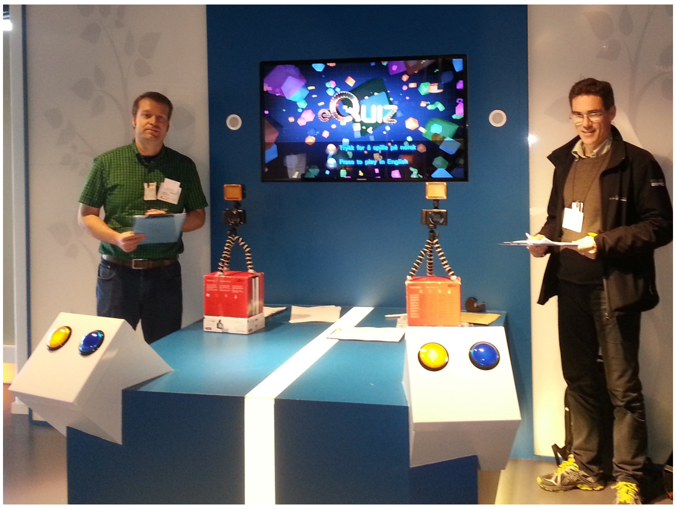

Figure 5: The installation ENG-12 at the Engineerium during the assessment.

to zero. Players answers by pressing either button within the countdown time. Players receive points for a correct answer and bonus points based on how quickly they answered. A player answering incorrectly loses points but can't go below zero. After the ten questions, a summary with the number of points scored for each player is presented.

In terms of the assessment model, the Scenario Layer contains the installation as the artefact (object) under observation, the visitors are the subjects, and the main action is to answer the questions by pressing buttons. The group of other visitors is the peer player. In the Data Collection and Observer Layer, we observe technical parameters from the installation, use a face reader and human observers to interpret emotions, and use surveys. Thus, in the Assessment Layer, technical parameters, physiological responses, and subjective assessment are employed. Since we are early in our investigations, the Assessment Process Layer is not yet fully implemented.

Figure 4 shows the VEI profile of ENG-12 with the solid line. We also show a version where only one player answers questions with the dotted line. This change lowers the values of both the C-dimension and the S-dimension.

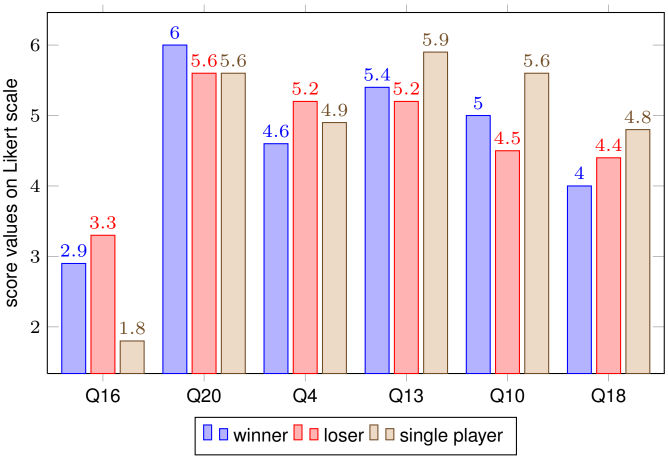

Figure 6: Response scores on a Likert scale for winners ($n = 29$, mean game score: 1966), losers ($n = 30$, mean game score: 1354), and single players ($n = 6$, mean game score: 1837) for the subjective constructs Q16 (too difficult), Q20 (engagement), Q4 (intention to answer again), Q13 (fun), Q10 (concentration), and Q18 (intention to learn more).

Figure 5 shows the installation ENG-12 during the assessment. In addition to the installation, we have installed two cameras that observe each of the players, one camera that observes the scene from behind, and, for each player, a human observer makes notes. The video footage is used both for manual analysis and automated analysis of facial expressions using the Face Reader software by Noldus [20]. We also made changes to the installation's software to log all events (e.g., which button is pressed, and score values) with timestamps.

The observers note visitor's mood using a simplified valence tracker [21], i.e., whether the visitor is excited-positive, excited-negative, or calm-neutral for each quiz question. These values are compared with the outcome of the Face Reader software. The self-reported data by the visitors consist of a self-developed questionnaire for ENG-12 and a 20-item PANAS scale [22]. Since we are interested in the the positive affect, i.e., the PA of the PANAS, we omitted factors that express negative emotions (e.g., *guilty* or *scared*) that hardly can be an impact from the use of the installation.

We performed tests to ensure that the preliminary technical setup is in place and working. This includes logging the events from the installation (objective data), interpretation of the video footage and light conditions, usefulness of the questionnaires and valence tracker, and conformance with the Norwegian privacy laws. Still, challenges need to be addressed, such as lighting problems or adjustments in the questionnaires (some items of the PANAS adjectives seem not to be understood by the target group; as a consequence, we did not use these items).

B. Results

We asked students from school classes that visit the Engineerium to use ENG-12 with our assessment equipment and observed them as described above. In four sessions between October 2014 and January 2015 we assessed data from 29 winners, 30 losers, and 6 single players. The data from one of

TABLE II: PANAS SCORES FROM THE EXPERIMENT.

PANAS	Pos.	Neg.
winners ($n = 29$)	34.0	16.5
losers ($n = 30$)	31.5	18.5
single players ($n = 6$)	34.0	20.3
std. dev. ($n = 65$)	6.8	5.0

the winners was discarded due to an irregularity (he played the game twice). We are aware that the number of single players is too low to give a significant result, and one of the single player responses is an outlier. So, we refrain from interpretations of the single player data. We show results from the subjective answers the players gave after having played ENG-12 with six selected questions in Figure 6. In TABLE II, we show the mean values of the positive and negative PANAS scores for the three groups and the mean value. We note that the standard deviation is in a similar range as published by Watson et al. [22] for assessments in the moment.

In our experiments, the automated face expression recognition fails in about 50% of the cases. The reasons for these failures include lighting problems (the light settings in science centres are often problematic for such analysis) and positioning of the cameras (these should be installed so that they do not obstruct essential parts of the installation). Given the achievable data quality of the data sets (14 winners and 17 losers), we registered about 70% smiles when an incorrect answer was given and about 40% smiles when a correct answer was given, independently whether they turned out to be winners or losers. Note that the smiles occur before the players know their ranking (winner or loser). The data from the valence tracker were only used to verify whether the assessment from the face reader is viable.

C. Interpretation

The interpretation of these data show rather small differences between winners and losers. However, a trend is visible: losers find the quiz questions somewhat more difficult (Q16). While they show lower engagement (Q20), their intention to answer again (Q4) and to learn more (Q18) is higher. They also report less fun (Q13) and less concentration (Q13). The PANAS scores show a similar trend, i.e., winners have a higher positive score while losers have a higher negative score. Note, however, that the differences are rather small. We also note that the trends in these responses are as expected between winners and losers. The data for the single players are not as expected, but due to low data quality we refrain from an interpretation.

For evaluating the impact the C-dimension in the VEI profile to the QoE, we do not yet have sufficient data quality, specifically for the single players. The fact that winners and losers show different values in the expected manner, both for the questionnaire and for the PANAS, shows that the C-dimension has an impact; else the two groups would not have shown differences.

The result concerning the number of smiles after each question suggests that the smiles might have a different social functionality than expressing enjoyment. However, the high number of smiles, specifically when answering incorrectly, show that the visitors are engaged and show emotions; that is that they are not indifferent. This also shows that it, in fact, is feasible to register engagement automatically.

VI. CONCLUSION

We presented the VEI profile to characterise installations at science centres, and a framework for assessing visitor engagement for installations. The goal is to assess engagement using measurable values from the installation, sensors, cameras, and so on, instead of using long-term observations and interpretation methods. Our current work shows the principles how to achieve this goal.

Currently, we have performed some preliminary assessments with ENG-12 with the metrics described here. The experiments so far have shown that registering engagement automatically is feasible. We need to perform more assessments with ENG-12 to get better data quality, as well as assessing other installations, the impact of other dimensions of the VEI profile, and the measurement of other data types in Layers II and III of our framework. While the goal is to develop a suitable estimation model in Layer IV of our framework, the collected data are not yet sufficient to apply machine learning methods.

VII. ACKNOWLEDGMENTS

The work presented here has been carried out in the project VISITORENGAGEMENT funded by the Research Council of Norway in the BIA programme, grant number 228737. The authors wish to thank Hege Røsto Jensen for her involvement in the assessment process.

REFERENCES

[1] P. Pierroux and A. Kluge, “Bridging the extended classroom: Social, technological and institutional challenges,” Nordic Journal of Digital Literacy, vol. 6, no. 3, 2011, pp. 115–120.

[2] A. Hofstein and S. Rosenfeld, “Bridging the gap between formal and informal science learning,” Studies in Science Education, vol. 28, no. 1, 1996, pp. 87–112.

[3] J. H. Falk and L. D. Dierking, “Learning from the outside in,” in Lessons without limit: How free-choice learning is transforming education. New York: Altamira press, 2002, pp. 47–62.

[4] J. Wang and A. M. Agogino, “Cross-community design and implementation of engineering tinkering activities at a science center,” in Proc. FabLearn 2013: Digital Fabrication in Education, Stanford, 27-28 October 2013, pp. 1–4. [Online]. Available: http://fablearn.stanford.edu/2013/papers/ [Accessed: 2 Feb 2015].

[5] M. Lindauer, “What to ask and how to answer: a comparative analysis of methodologies and philosophies of summative exhibit evaluation,” museum and society, vol. 3, no. 3, November 2005, pp. 137–152.

[6] A. Šuldová and P. Cimler, “How to assess expirience – the new trend in research technique, use in nonprofit sector of entertainment and educational industries,” Marketing a obchod, vol. 4, 2011, pp. 115–124.

[7] C. W. Sanford, “Evaluating Family Interactions to Inform Exhibit Design: Comparing Three Different Learning Behaviors in a Museum Setting,” Visitor Studies, vol. 13, 2010, pp. 67–89.

[8] L. D. Dierking and J. H. Falk, “Redefining the museum experience: the interactive experience model,” Visitor Studies, vol. 4, no. 1, 1992, pp. 173–176.

[9] J. Falk and M. Storksdieck, “Using the contextual model of learning to understand visitor learning from a science center exhibition,” Science Education, vol. 89, 2005, pp. 744–778.

[10] J. H. Falk, “The impact of visit motivation on learning: Using identity as a construct to understand the visitor experience,” Curator, vol. 49, no. 2, 2006, pp. 151–166.

[11] J. H. Falk, J. Heimlich, and K. Bronnenkant, “Using identity-related visit motivations as a tool for understanding adult zoo and aquarium visitors’ meaning-making,” Curator: The Museum Journal, vol. 51, no. 1, 2008, pp. 55–79.

[12] C. Barriault and D. Pearson, “Assessing Exhibits for Learning in Science Centers: A Practical Tool,” Visitor Studies, vol. 13, 2010, pp. 90–106.

[13] D. L. Young, “A phenomenological investigation of science center exhibition developers’ expertise development,” Ph.D. dissertation, The University of North Carolina at Chapel Hill, 2012, 174 pages.

[14] S. Allen, “Designs for learning: Studying science museum exhibits that do more than entertain,” Sci. Ed., vol. 88, no. S1, 2004, pp. S17–S33.

[15] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card, “The cost structure of sensemaking,” in Proc. of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems. New York: ACM, 1993, pp. 269–276.

[16] G. Klein, B. Moon, and R. R. Hoffman, “Making sense of sensemaking 1: Alternative perspectives,” IEEE Intelligent Systems, vol. 21, no. 4, Jul. 2006, pp. 70–73.

[17] ——, “Making sense of sensemaking 2: A macrocognitive model,” IEEE Intelligent Systems, vol. 21, no. 5, Sep. 2006, pp. 88–92.

[18] C. M. Bishop, “Pattern Recognition and Machine Learning (Information Science and Statistics)”. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[19] W. Leister and I. Tjøstheim, “Concepts for User Experience Research,” Norsk Regnesentral, NR Note DART/14/2012, 2012.

[20] V. Terzis, C. N. Moridis, and A. A. Economides, “Measuring instant emotions during a self-assessment test: The use of facereader,” in Proc. Measuring Behavior 2010, A. J. Spink, F. Grieco et al., Eds., Eindhoven, The Netherlands, 2010, pp. 192–195.

[21] G. Chanel, K. Ansari-Asl, and T. Pun, “Valence-arousal evaluation using physiological signals in an emotion recall paradigm,” in Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, oct 2007, pp. 2662–2667.

[22] D. Watson, L. A. Clark, and A. Tellegen, “Development and validation of brief measures of positive and negative affect: the PANAS scales,” Journal of Personality and Social Psychology, vol. 54, 1988, pp. 1063–1070.