

INSTRUMENTATION AND MEASUREMENT & ENGINEERING IN MEDICINE AND BIOLOGY JOINT CHAPTER

PROCEEDINGS OF THE 22ND PHD MINI-SYMPOSIUM

FEBRUARY 2, 2015



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

PROCEEDINGS OF THE 22ND PHD MINI-SYMPOSIUM

FEBRUARY 2, 2015 BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS BUILDING I



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

©2015 by the Department of Measurement and Information Systems Head of the Department: Prof. Dr. Ákos Jobbágy

Conference chairman: Béla Pataki

> Organizers: Balázs Renczes Gábor Wacha Bálint Ferencz Bence Bolgár

Homepage of the Conference: http://minisy.mit.bme.hu/

Sponsored by:

IEEE Instrumentation and Measurement & Engineering in Medicine and Biology Joint Chapter

Schnell László Foundation

FOREWORD

This proceedings is a collection of the lectures of the 22nd PhD Mini-Symposium held at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics. The main purpose of these symposiums is to give an opportunity to the PhD students of our department to present a summary of their work done in the preceding year. It is an interesting additional benefit, that the students get some experience: how to organize such events. Beyond this actual goal, it turned out that the proceedings of our symposiums give an interesting overview of the research and PhD education carried out in our department. The lectures reflect partly the scientific fields and work of the students, but we think that an insight into the research and development activity of the department is also given by these contributions. Traditionally our activity was focused on measurement and instrumentation. The area has slowly changed during the last few years. New areas mainly connected to embedded information systems, new aspects e.g. dependability and security are now in our scope of interest as well. Both theoretical and practical aspects are dealt with.

The proceedings will not be published in printed form, it has turned out that nowadays the web publication of symposium lectures is enough. This new form has some advantages, but it has some challenges as well. We hope that the advantages will dominate.

The papers of this proceedings could be sorted into two main groups: Model-based and Intelligent Systems; and Signal Processing and Data Mining. The lectures are at different levels: some of them present the very first results of a research, others contain more new results. Some of the first year PhD students have been working on their fields only for half a year.

There are two types of papers. One is a short independent publication, it is published in the proceedings. The other is simply a summary of the PhD student's work. This second one is intended to give an overview of the research done during the last year; therefore, it could contain shorter or longer parts of the PhD student's other publications. It does not necessarily contain new results, which have not been published earlier. It is clearly indicated in each paper that which category it belongs to. To avoid copyright conflicts, these papers are not published in the proceedings. Anyone interested, please contact the author.

During this twenty-one-year period there have been shorter or longer cooperation between our department and some universities, research institutes, organizations and firms. Some PhD research works gained a lot from these connections. In the last year the cooperation was especially fruitful with the European Organization for Nuclear Research (CERN), Geneva, Switzerland; Vrije Universiteit Brussel Dienst ELEC, Brussels, Belgium; Robert Bosch GmbH., Stuttgart, Germany; ERICSSON Magyarország Kft., Budapest; Complex Systems and Computational Neuroscience Group at Wigner Research Center, Budapest; National Instruments Hungary Kft., Budapest; IEEE Instrumentation and Measurement Society & Engineering in Medicine and Biology Society Joint Chapter, IEEE Hungary Section.

We hope that similarly to the previous years, also this PhD Mini-Symposium will be useful for the lecturers, for the audience and for all who read the proceedings.

Budapest, January, 2015.

Béla Pataki Chairman of the PhD Mini-Symposium

LIST OF PARTICIPANTS

Participant	Supervisor	Starting Year of PhD Course
Bolgár, Bence	Antal, Péter	2012
Bozóki, Szilárd	Pataricza, András	2014
Cserpán, Dorottya	Horváth, Gábor	2011
Darvas, Dániel	Majzik, István	2014
Debreczeni, Csaba	Varró, Dániel	2014
Dülk, Ivor	Kovácsházy, Tamás	2012
Hajdu, Csaba	Dabóczi, Tamás	2014
Pataki, András	Horváth, Gábor	2014
Renczes, Balázs	Kollár, István	2013
Salánki, Ágnes	Pataricza, András	2014
Semeráth, Oszkár	Varró, Dániel	2014
Szárnyas, Gábor	Varró, Dániel	2014
Tóth, Tamás	Majzik, István	2014
Virosztek, Tamás	Kollár, István	2014
Wacha, Gábor	Fehér, Béla	2013

PROGRAM OF THE MINI-SYMPOSIUM

Model-Based and Intelligent Systems	Chair: Jobbágy, Ákos		
Salánki, Ágnes	Identification of Dependability Models from Large Observation Sets	p8	
Tóth, Tamás	A Framework for Formal Verification of Real-Time Systems	p12	
Bozóki, Szilárd	Resiliency techniques for cloud-based applications	p14	
Dániel, Darvas	Requirements towards a formal specification language for PLCs	p18	
Bolgár, Bence	GPU-accelerated Metric Learning from multiple rela- tions	*	
Semeráth, Oszkár	Example and Counterexample Generation for Domain-Specific Languages	*	
Szárnyas, Gábor	Distributed Incremental Graph Queries	*	
Debreczeni, Csaba	Automated abstraction in model-driven engineering	*	

Signal ProcessingChair: Horváth, Gáborand Data MiningChair: Horváth, Gábor

Cserpán, Dorottya	Preprocession and Interpretation of Multielectrode	p22
	Array Recordings	
Pataki, András	Using Data Mining Techniques for Estimating Printed	p26
	Circuit Board Design Complexity	
Wacha, Gábor	Examination of the input dependence of data flow	p30
	graph clusterization	
Hajdu, Csaba	Design and Implementation of a Real-Time System	p34
	Survey for Beam Loss Monitoring Systems	
Renczes, Balázs	Accuracy Problems in Three- and Four-Parameter	*
	Least Squares Sine Wave Fit	
Virosztek, Tamás	Parameterization of Nonlinearity for Efficient Estima-	*
	tion in ADC Testing	
Dülk, Ivor	Parameter Estimation in Electromagnetic Devices by	*
	the Multilayered Medium and Model Based Approach	

The research reports indicated by * are not published in this volume. To gain access, please contact the organizers or the authors.

CONFERENCE SCHEDULE

Time	February 2, 2015
8:30	Opening speech
8:35	Model-Based and Intelligent Systems
10:35	Coffee break
11:00	Signal Processing and Data Mining

Identification of Dependability Models from Large Observation Sets

Ágnes Salánki Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary Email: salanki@mit.bme.hu

Abstract—Dependability analysis of complex infrastructures necessitates a sophisticated statistical analysis in order to estimate the causes of failures. The paper presents a combination of visual analytics and algorithmic methods specialized to rare event analysis. The main outcomes of the proposed analysis process are a statistical model of the system and the estimation of the main factors to be used in supervisory systems for an early detection of potential failures.

I. INTRODUCTION

By design, failures in high-availability systems are extremely rare. Resilience of such systems can be analyzed only empirically, because the diversity of the building components and the complexity of their interaction jointly exclude efficient usage of analytic methods. System models typically lack an exact statistical projection, thus, nonparametric analytics methods are needed.

Visual exploratory data analysis (EDA) performed by domain experts is a promising approach to support empirical construction of dependability model. However, application of analysis techniques can be challenging because of rare occurrence of fault events (*outliers, anomalies*). Nowadays, a variety of tools exist for detection of rare events in a multidimensional space. These automated methods without exact parameter tuning may retrieve spurious relationships instead of semantically correct results.

Visualization-based and algorithmic methods have the potential to complement each other [2], [3]. Visual analytics is intuitive, exploits the semantic interpretation skills of an expert typically without a demand for deep statistical knowledge. However, visual analytics can only provide rough estimations of clusters. On the other hand, computation-centric methods are able to select best-fitting boundaries to data clusters, find the best configuration, build models and test hypotheses.

Three types of integration are defined in [3]: a) *computationally enhanced visualization* techniques (e.g. artificial selection of ,,interesting'' scatterplots); b) *visually enhanced mining* (e.g. presentation of hierarchical clustering with dendograms) and c) *fully integrated visualization and mining* (e.g. interactive decision tree building in [2]), where the current approach also belongs to as well.

The goal of this paper is to present an approach for computationally enhanced visual detection and characterization of rare events. We find the synergistic integration of the András Pataricza Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary

Email: pataric@mit.bme.hu



Fig. 1. The visual outlier detection approach

semantic interpretation capability of an expert used in visual analytics with the precise processing of algorithmic methods very promising.

Parallel coordinates plots (PCPs) [8] were selected for visualization as the most sufficient means to present multidimensional data sets.

The user can select an initial set of candidate outliers on the PCP interactively, thereafter a dedicated algorithm refines the parametrization of the DB [10] detection algorithm. Thus, the processing of a large data set can be reduced to semisupervised learning by this outlier-focused relevance filtering. Results from the automated detection are back-annotated on the PCP to allow the user to refine the query.

II. RELATED WORK

A. Outlier detection

Outliers are data points deviating significantly from other observations. Accordingly, it is deduced that they were generated potentially by a different mechanism than the bulk of data [7]. Outliers are categorized into three main groups [4]: 1) *point outliers;* 2) *collective outliers* and 3) *contextual outliers*. Point outliers are observations individually differing from the rest of the data set, e.g. an extreme peak in CPU usage. Contextual outliers are observations of a context-dependent deviation, e.g., a large peak in CPU usage may be suspicious only at night. A collective outlier group includes a subset of observations, collectively deviating from the bulk of points, even if the individual observations themselves may not be outliers, e.g., continuously large CPU usage in a system where only short peaks are allowed. The two main groups of outlier detection algorithms are *distance-based* and *density-based* methods. Distance-based methods assume that outliers are far from the bulk of points or the center of the entire data set. This family of methods contains e.g. DB, which marks a node as outlier if it has only few neighbors around it in a hypersphere. Density-based approaches mark those data points as outliers whose density (e.g., average number of neighbors) is highly different of their *local* neighborhood (LOF or NNDB [12]).

B. Multi-dimensional visualization techniques

General-purpose multi-dimensional visualization techniques, including PCPs, scatterplot matrices or tableplots [13], all focus on a specific aspect of the observation set. Other methods, like MDS (*Multi-Dimensional Scaling*), PCA (*Principal Component Analysis*) plots or biplots project the multidimensional space into a subspace of a lower dimension and simultaneously highlight some special characteristics of the observation set. [9] presents specific transparency functions to control the visualization of outliers in PCP.

Interestingness measures and labels can be used as guidance of visual analysis by a preference attribute defined by the user. These metrics and annotations are analyzed simultaneously during the analysis as support in the subsequent steps for search space reduction or result ranking [5] [6].

III. ANALYSIS WORKFLOW IN IT INFRASTRUCTURE MANAGEMENT

This section summarizes the possible steps of a failuretriggered analysis workflow (Fig. 2). Its primary goal is root cause analysis while identification of trends or a fine diagnosis is only of secondary importance.

Usually the assurance of high availability, especially in our focus area of cloud computing, is a main engineering objective. Accordingly, coarse-grained diagnosis is sufficient to take mitigation actions. A detailed analysis serves only the fine-tuning and extension of automated monitoring and control mechanisms of the supervisory system.

Failure-triggered analysis of IT systems consists of two steps: 1) *analysis of phenomenological observations* (failure analysis); 2) *analysis of the monitored infrastructure* (root cause analysis).

The goal of failure analysis is to detect any sudden parasitic degradations in the high-level QoS metrics to avoid permanent SLA violation, assuming that our system works typically correctly and errors happen only very rarely. The corresponding analytics methodology here is outlier detection.

We assume that the number of high-level metrics is around 10 in our domain (e.g. [1]); root cause analysis still has to deal with a possibly large number of low level infrastructure metrics. There are several million observations, according to the typical sampling frequency and error latency, this may represent a day with time-triggered data acquisition methods applied in modern large systems.

Root cause analysis serves identification and elimination of faults in the system causing the observed QoS degradation, e.g. to find the resource acting as a bottleneck. Thereafter,



Fig. 2. Analysis workflow of a typical IT environment

new monitoring rules can be added to the supervisory system to detect, diagnose and mitigate such faults. The analytics method here is *classification*, being capable of exploring and localizing the data set subspaces corresponding to erroneous system states.

Root cause analysis sometimes fails to identify the fault. The main reason for this problem is that no complete a priori fault model can be constructed for such large-scale and complex systems. If the failure is caused by an unobserved factor (e.g., some change in the environment), then frequently no significant difference in resource utilization, operational domain, etc. can be found prohibiting a proper diagnosis. The prediction of the QoS degradation based solely on detecting changes in high-level observations before the rare event is still possible at the phenomenological level. *Dynamic time series analysis* methods are the most promising in this case.

IV. INTERACTIVE OUTLIER DETECTION WITH PCPs

Outlier detection is usually performed in the first phase of the analysis process. The domain expert faces two challenges at this point: 1) *very efficient multi-dimensional visualization* is needed requiring no exact knowledge about relevant factors in the operational domain under investigation; 2) *parametrization of automated outlier detection algorithms* is a hard problem, their indirect estimate is sometimes not intuitive.

Our approach supports an efficient offline outlier detection and characterization along the following workflow (Fig 1):

- Step 1 the user selects an initial subset of candidate outliers on the PCP plot and estimates the number of outliers in the data set: this number can be influenced by the domain, the time window, etc.;
- **Step 2** the DB algorithm is parameterized based on the previous user interaction; thereafter the algorithm determines the entire set of outlier candidates;
- **Step 3** the outliers (and simultaneously the dense clusters as point of reference) are visualized, where colors reflect the decision parameters of the algorithm and axes are reordered to highlight the relevant dimensions separating the outliers from normal points.

Individual steps are detailed in the following subsections.

A. Parameter estimation

An object x in a dataset T is a DB(p, D) outlier if at least a fraction p of the objects in T lies from x greater than distance

D [10]. This is one of the basic, actively used concepts in the outlier detection community. Despite the simplicity of the intuitive definition, it is cumbersome to estimate the parameters p and D in the early exploration phase of the analytics process. However, the domain expert could have an exact vision about the proportion of outliers in the entire data set and, after exploring it, which observations seem most suspicious. Our approach presents a heuristic of estimation of p and D, based on the aforementioned two input parameters.

The parameters of the algorithms (Algorithm 1 and Algorithm 2) contain the data set T, the user-defined candidate subset O, the estimated proportion of outliers in the data set ε and the number of maximal allowed iterations n.

Algorithm 1 computes a (p, D) value pair based on the maximum proportion of neighbors to be considered *i*. First, the empirical distribution function is computed from the distances around each outlier candidate o (line 1). Secondly, a minimal distance D is defined, assuming that an outlier candidate has at most *i* neighbors in distance D (line 2). The final estimate for parameter p is computed from this distance to ensure that the selected candidate subset satisfies the DB outlier conditions (line 3). Finally, the size of the final outlier subset is determined (line 4).

Algorithm 1 Parameter estimation for a specific initial *i* quantile

Input: *T*: entire data set, *O*: outlier candidate subset, 1: maximum proportion of neighbors to be considered;

- **Output:** D_i : border distance of close neighbors,
- p_i : border proportion of close neighbors,
 - nn_i : vector containing the number of close neighbors
- 1: h_x : distance from other points for $\forall x \in T$

2: $D_i := \min_{o \in O} \operatorname{ecdf}_{h_o}^{-1}(i)$

- 3: $p_i := \max_{o \in O} \operatorname{ecdf}_{h_o}(D)$
- 4: $nn_i := |x : x \in T, \operatorname{ecdf}_{h_x}(D) \le p|$
- 5: return p_i, D_i, nn_i

Algorithm 2 computes possible (p, D) values in discrete points in $[0, \varepsilon]$, considering the maximal number of iterations n. The estimated values of p and D are calculated by Algorithm 1 in every iteration, which results in different outlier sets. No convergence can be assumed without a priori knowledge about distributions, thus, we calculate different outlier candidate sets at discrete points. At the end, an ideal parametrization is chosen to minimize the difference between the expected and the calculated outlier set size.

B. Visualization of outliers

This phase of analysis aims at visualization of the input data set; special coloring and axis reordering were chosen to reflect the DB results and highlight the characteristics of outliers.

Coloring: Any visual characteristics of a parallel coordinates line could be used to highlight outliers, e.g. transparency, color, size, texture, etc. Coloring was chosen in our visualization, because this is a non-additive characteristics: dense clusters do not amplify each others' value, as transparency, texture or size do.

The color of each observation is based on its number of neighbors. The breakpoints on the color scale partitions

Algorithm 2 Parameter estimation for DB algorithm

- **Input:** T: entire data set, ε : estimated proportion of outliers in the data set, O: outlier candidate subset, n: maximum number of iterations
- **Output:** p_i : border proportion of close neighbors, D_i : border distance of close neighbors
- 1: for i := 0 to ε do
- 2: D_i : border distance of close neighbors applying Algorithm 1
- 3: p_i : border proportion of close neighbors applying Algorithm 1
- 4: *nn_i*: vector containing the number of close neighbors applying Algorithm 1
- 5: $error_i := |nn_i \varepsilon|$
- 6: $i + = \varepsilon/n$
- 7: end for
- 8: $init := \arg\min_i error_i$
- 9: return p_{init}, D_{init}

the observations into three main groups: outliers (low values) are colored with red, normal points (high values) are blue, intermediate points are yellow. This is another supporting feature to highlight the separation quality of regions.

Axis order: Relevant dimensions to be monitored can be selected differently from one operational domain to another one. Thus, an appropriate order of coordinates axes should be computed to determine the best dimensions separating outliers from normal points. For this, the selected detection algorithm (e.g., DB) is run on every one-dimensional subset. Dimensions are ranked based on the correlation between their value and the final outlier score. The strength of the interdependence of two dimensions is evaluated by metrics, which can be used in the case of a non-linear relation.



Fig. 3. An initial user interface and results from different initial subsets

Results from two different subset selections are presented in Fig. 3. The typical cluster visualizations with low transparency (left) and outlier parallel coordinates plot (right) are present side-by-side, where the former serves as point of the reference and so supports outlier characterization. The first panel (top) presents the initial user interface, the other two (middle and bottom) show results of different subset selections. Although the set of outliers are different, identical axis reordering was performed.



a.ex.lata.pr.lat a.cap g.ex.latg.pr.lat g.cap s.ex.lats.pr.lat s.cap



a.cap g.cap g.pr.lats.ex.lats.pr.lata.ex.lata.pr.latg.ex.lat s.cap



V. CASE STUDY: OUTLIER DETECTION IN A STREAM PROCESSING ENVIRONMENT

Our outlier detection approach is demonstrated on a data set collected in the Storm [1] stream processing environment, while running an application computing the average delay of US flights. Data tuples in the experimental environment are being transferred between three types of nodes: *aggregator*, the *gatherer* and the *sweeper*, being responsible for aggregation by date, the computation on the aggregated data and the time handling.

Three metrics are stored about every component: *process latency*, *execution latency* and *capacity*. The first two reflect the time spent between start and end of processing of the incoming tuple, the time spent between start of processing and the formal verification of it, while the capacity is a derived metric in the range of [0, 1], reflecting how strongly the actual component can be considered as a bottleneck in the system. The environment was monitored by a 10 sec sampling time; the data set contained around 1800 observations in 9 dimensions in this small-scale pilot.

The initial user interface and the final result after selection of a tight subset is presented on Fig. 4. Examining the clustering and outlier plots simultaneously, we can notice that a large subset of outliers and intermediate observations show a typical behavior in every dimension but the capacity dimensions of the aggregator and gatherer node (*a.cap* and *g.cap*). These were chosen as two first axes: their high values show indeed a strange behavior. Thus, we can conclude that the most relevant factors to be observed in the future are the capacity of aggregator and the gatherer.

VI. CONCLUSION AND FUTURE WORK

The paper presented an interactive outlier detection and characterization approach for computing enhanced exploration of outliers.

Although PCP is an ideal visualization technique in terms of scalability and information compression, the method is unable to efficiently give to the users an idea about distances between points (contrary, e.g. to the scatterplot matrices). Coloring can partly eliminate this problem. Initial visual analysis approaches in our domain (e.g., [11]) suggest that





operational domains are frequently analyzed separately. Thus, use of distance based algorithms seems very promising in our domain.

Extensions of current user interface is planned to support visualization of object relationships typical in our domain: absolute and relative amount of resource metrics on different levels, possible interference between virtual machines, etc.

REFERENCES

- [1] Apache storm project. https://storm.apache.org/.
- [2] Mihael Ankerst, Martin Ester, and Hans-Peter Kriegel. Towards an effective cooperation of the user and the computer for classification. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00, pages 179–188, New York, New York, USA, 2000. ACM Press.
- [3] Enrico Bertini and Denis Lalanne. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration, pages 12–20, 2009.
- [4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3):15, 2009.
- [5] Tijl De Bie. Subjective interestingness in exploratory data mining. In Advances in Intelligent Data Analysis XII, pages 19–31. Springer, 2013.
- [6] Liqiang Geng and Howard J Hamilton. Interestingness measures for data mining. ACM Computing Surveys, 38(3):9–es, September 2006.
- [7] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.
- [8] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates. In Human-Machine Interactive Systems, pages 199–233. Springer, 1991.
- [9] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 125–132. IEEE, 2005.
- [10] Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov. Distancebased outliers: algorithms and applications. *The VLDB JournalThe International Journal on Very Large Data Bases*, 8(3-4):237–253, 2000.
- [11] András Pataricza, Imre Kocsis, Ágnes Salánki, and László Gönczy. Empirical assessment of resilience. In Software Engineering for Resilient Systems, pages 1–16. Springer, 2013.
- [12] András Pataricza and Ágnes Salánki. Detection of rare events. In Péter Antal, editor, *Intelligens adatelemzés*, pages 28–45. Typotex Kft., Budapest, 2014.
- [13] Martijn Tennekes, Edwin de Jonge, and PJH Daas. Visual profiling of large statistical datasets. In New Techniques and Technologies for Statistics conference, Brussels, Belgium, 2011.

A Framework for Formal Verification of Real-Time Systems

Tamás Tóth

Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary Email: totht@mit.bme.hu

Abstract—Formal methods have an important role in ensuring the correctness of safety critical systems. However, their application in industry is always cumbersome: the lack of expertise and the complexity of specification languages prevent the use of formal verification techniques. In this paper we take a step in the direction of making formal methods applicable by introducing a framework that enables efficient modeling and analysis of realtime systems.

I. INTRODUCTION

Nowadays, an ever increasing number of information systems are critical embedded systems that have a dedicated function in a specific, safety critical application environment (e.g., components of a railway control system). In case of such systems, failures may endanger human life, or result in serious environmental or material damage, thus ensuring conformance to a correct specification is crucial for their development.

To guarantee that a system operates according to its specification, formal verification techniques can be used. These techniques are based on the formal representation the system and its properties (requirements), which enables to apply mathematical reasoning to investigate their relationship. Moreover, these methods allow verification of systems in an early phase of the development life cycle prior to testing.

Since the behavior of safety critical systems is often time dependent, the notion of time has to be represented in their models. Timed automata are the most prevalent way to model timed systems. However, this formalism is only suitable to describe timed behavior with respect to constant values, thus its expressive power is insufficient to model systems with parametric behavior. Parametric timed automata, an extension of the original formalism, address this problem.

In this paper we introduce a formal modeling framework for supporting the efficient development of formal models based on parametric timed automata. The modeling language is based on the language of the well-know *Symbolic Analysis Laboratory* (SAL) framework¹ with extensions to simplify the work of the modelers. These extensions enable the modeling of time dependent behavior on language level.

1http://sal.csl.sri.com/

István Majzik

Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary Email: majzik@mit.bme.hu

II. RELATED WORK

Our work is inspired by the SAL model checker [5] and its language (our extensions are introduced in Section III). The SAL language enables compact modeling of systems in terms of (unlabeled) symbolic transition systems, however it does not support explicit modeling of time related behavior. The aim was to preserve compatibility so that the timed models of our extended language can still be transformed to the input of SAL. As another related tool, UPPAAL [1] is a model checker widely used for the verification of timed systems. It has a graphical interface and it provides efficient model checking algorithms to verify timed automata. UPPAAL models can also be transformed to our language with some restrictions: our formalism does not handle complex function declarations. Our approach has different strengths as the underlying Satisfiability Modulo Theories (SMT) technologies are efficient in reasoning over complex data structures of the modeled systems. In addition, complex synchronization constraints can be compactly expressed in our approach. The industrial case study of the paper was first introduced in [6], where the SAL model checker was used for the verification. Our paper summarizes the lessons learnt from that work.

III. THE SPECIFICATION LANGUAGE

The core of the specification language is a constraint language that enables declaration of uninterpreted constant symbols of complex data types and constraints over them. Supported data types include boolean, integer, real, enumeration, function, array, tuple and record types, and subtypes that are constrained by some formula. Constraints are sorted first order logic formulae in the combined theory of integer and real arithmetic.

The specification language enables the modeling of real time systems by means of symbolic transition systems with clock variables and parameters. The definition of a system consists of variable declarations, invariant and urgency constraints that constraint the traces of the system, and the description of the initialization and transition relations of the system with guarded commands. Systems can be composed synchronously or asynchronously to define more complex behavior. Properties are CTL* formulae over system variables.

For a more detailed description we refer the reader to [7].

IV. THE VERIFICATION WORKFLOW

The semantics of the language is provided by a series of simplifying model transformations, and a mapping to an SMT problem. The starting point of the workflow is a model in the above language given either directly, or as a result of a transformation from other timed formalisms, e.g. *UPPAAL* [1].

As a first step, the system is automatically flattened, that is, the result of a synchronous, respectively asynchronous composition is established. This is performed by merging the variables, invariant and urgency constraints, and initialization and transition definitions of the components. During this step, many of the constructed transitions can be eliminated by simply checking the satisfiability of their guards with a call to the underlying SMT solver [4].

In the next step, the model is automatically "untimed" by expressing the semantics of delay transitions explicitly. Here, a combined transition semantics [3] [5] is considered, where a transition merges the effects of a delay transition, followed by a discrete transition. For that purpose, a new input variable dis introduced to represent time delay. Such an untimed system model can easily be mapped to SAL or other intermediate formalisms. At the same time, transition (and initialization) definitions are completed, that is, assignments for unmodified variables (e.g., variables of asynchronous components) are made explicit. As a result, each variable (even those of some complex data type) is assigned a value in at most one assignment of a behavior definition.

The symbolic transition system represented by the model can then be easily expressed in SMT by transforming initialization and transition definitions of the system to predicates $I(\bar{x})$, respectively $T(\bar{x}, \bar{x}')$ as usual [2].

The tooling is implemented in *Eclipse*² using *Eclipse Modeling Framework*³ and relating technologies.

- *Abstract syntax.* The abstract syntax of the language is implemented as a metamodel in *EMF*. It is defined as an extension of the core language suitable for defining complex data types and expressions.
- *Concrete syntax.* The textual concrete syntax is defined by an *LL**-parsable grammar. The textual editor is then generated using the *Xtext*⁴ tooling.
- *Semantics*. Model transformations that define the semantics of the language are implemented in *Xtend*⁵.
- *Well-formedness rules*. Together with other validation constraints, algorithms for type checking and type inference are implemented for the type system of the language.

V. EVALUATION AND CONCLUSION

This paper is one step towards scalable formal modeling. We proposed a modeling language to provide better support for the designers of formal models by focusing on the aspects of data semantics, time dependent behavior, parametrization, and

²http://eclipse.org/

synchronous and asynchronous composition of components. Instead of manually coding flat transition systems, we provided automated model transformations from our extended language to more simple transition systems that can be directly mapped to the input of existing SMT solvers. This way an automated verification workflow is offered.

To evaluate the effectiveness of our language, the formal model of the introduced case study was developed in both our and the SAL languages. Comparing the results, the complexity of the developed models are the following:

The SAL model contains:

- 5 components for modeling the basic behavior consisting of 410 lines of code,
- 2 components for supporting the proper analysis of the temporal logic specification consisting of 105 lines of code,
- 3 components for recognizing the loops in the state space (required for the verification) consisting of 145 lines of code.

The formal model in our extended language contains 4 components and 235 lines of code, which demonstrates that the new language has its advantage. Moreover, it does not require additional components for the analysis.

Compared to UPPAAL timed automata, the main advantage of our language is the greater flexibility in handling clock variables and clock constraints. However, as UPPAAL provides a graphical modeling interface, in case of small models it makes the development of formal models more simple. Regarding the efficiency of verification, both approaches have their strengths and further investigations are needed.

In the future we plan to further improve our language with higher level modeling constructs. We also plan to develop new model checking algorithms based on induction techniques.

REFERENCES

- G. Behrmann, A. David, K. G. Larsen, O. Möller, P. Pettersson, and W. Yi. UPPAAL - present and future. In *Proc. of 40th IEEE Conference* on Decision and Control. IEEE Computer Society Press, 2001.
- [2] E. Clarke, A. Biere, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7– 34, 2001.
- [3] R. Kindermann, T. Junttila, and I. Niemel. Smt-based induction methods for timed systems. In M. Jurdziski and D. Nikovi, editors, *Formal Modeling and Analysis of Timed Systems*, volume 7595 of *LNCS*, pages 171–187. Springer Berlin Heidelberg, 2012.
- [4] L. Moura and N. Björner. Z3: An efficient smt solver. In C. Ramakrishnan and J. Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *LNCS*, pages 337–340. Springer Berlin Heidelberg, 2008.
- [5] L. Pike. Real-time system verification by k-induction. Technical Report TM-2005-213751, NASA Langley Research Center, May 2005.
- [6] T. Tóth, A. Vörös, and I. Majzik. K-induction based verification of realtime safety critical systems. In W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, editors, *New Results in Dependability* and Computer Systems, volume 224 of Advances in Intelligent Systems and Computing, pages 469–478. Springer International Publishing, 2013.
- [7] T. Tóth, A. Vörös, and I. Majzik. Verification of a real-time safety-critical protocol using a modelling language with formal data and behaviour semantics. In A. Bondavalli, A. Ceccarelli, and F. Ortmeier, editors, *Computer Safety, Reliability, and Security*, volume 8696 of *Lecture Notes in Computer Science*, pages 207–218. Springer International Publishing, 2014.

³http://eclipse.org/modeling/emf/

⁴http://eclipse.org/Xtext/

⁵http://eclipse.org/xtend/

Resiliency techniques for cloud-based applications

Szilárd Bozóki

Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary Email: bozoki@mit.bme.hu

Abstract— Resilience of runtime environments provided for cloud based systems is a prerequisite of deploying mission critical applications. It requires a tradeoff between the applied redundancy and its dependability increase. Offering cheap, on-demand redundancy is one of the most attractive benefits of cloud computing which makes even high redundancy based solutions affordable. The current paper revisits traditional fault tolerant design patterns form the cloud perspective.

Keywords— Resiliency, redundancy, fault tolerance, cloud, design patterns

I. INTRODUCTION

Migration to the cloud became an appealing alternative for manifold use cases and applications due to the prodigious development of cloud computing. Even mission critical and SLA (*service level agreement*) bound applications started to migrate from proprietary dedicated servers to commercial cloud services with no special platform SLA guarantees because of the huge economic benefits. For instance, telco solution providers, traditionally confined by strict carrier grade requirements, are yet to embrace the benefits of cloud computing through Network Function Virtualization [1].

Cloud computing has unique features that generally affect resiliency. For instance, offering cheap, on-demand redundancy is one of the most attractive benefits of cloud computing. In order to utilize the true potential of cloud computing based runtime environments the age old and proven fault tolerant design patterns need fundamental rethinking.

The logic interface (*cloud stack*) hides the majority of details of the underlying platform. However, assurance of user level SLAs is highly dependent on the underlying platform, thus a measurement and calculation methodology of platform KPIs (*key performance indicator*, such as performance profiles of the virtual network and computing power etc.) is imperative.

These KPIs can guide the architecture design phase of system development from the extra functional requirements point of view, like design for dependability.

The practice of design for fault tolerance searches for a proper tradeoff between the assurance of the designated dependability of the target application and the used redundancy.

Currently there are three known main aspects to revisit:

1. What are the *required level* and *appropriate structure of redundancy* (e.g. a spare pool of VMs) guaranteeing the continuous fulfilment of SLA requirements?

András Pataricza

Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary Email: pataric@mit.bme.hu

- 2. What is the proper way to adopt these solutions to different cloud platforms in order to cost efficiently match an application level SLA?
- 3. What is the proper characterization of the designated user service in order to efficiently fit it to the fault tolerance design pattern?

The current paper tries to answer these questions from cloud user point of view by examining what are the pros and cons of the different resiliency techniques, with focus on downtime and cost.

The rest of the paper is organized as follows. Section 2 introduces the main factors influencing design for resiliency. Section 3 revisits fault tolerant design patterns for architecture design. Section 4 presents the analysis methodology of design patterns in more detail. Section 5 concludes the paper by presenting an outlook on future work.

II. FOUNDATIONS OF CLOUD COMPUTING

A. Cloud computing

NIST defines cloud computing as follows [2]:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

1) Essential characteristics of cloud computing [2]

- *On-demand self-service* and *rapid elasticity* facilitate active resource management by the user and the applied fault tolerant mechanisms.
- *Broad network access* led to the integration of geographically remote resources into a single virtual platform. However, fault tolerance necessitates to take into account the distributed nature of cloud computing among others by the vulnerability of the interconnection fabric.
- *Resource pooling*, a.k.a. resource sharing, is the fundamental mechanism for an excellent utilization of resources, however, it opens new surfaces for error and intrusion propagation, even between functionally independent applications of different owners. The quality of the provisioned resources varies depending on tenant activity.

(e.g through noisy neighbors) This variance places an extra layer of complexity on resiliency.

- *Measured service* and pricing models of the providers play a vital role in the exact quantification of the cost of resiliency.
 - 2) Service models of cloud computing: [2]

The idea of the *service stack* needs to be summarized, before explaining the service models.

The service stack contains the layers and components which cooperate in a service provisioning application.

Generally, the following main layers of a service stack can be identified from bottom up (Fig. 1): hardware, virtual machine monitor (VMM, aka hypervisor in many cases) usually residing in a host operating system, virtual machine (VM), guest operating system, application container and the application.

The three primary service models of cloud computing, namely *Software as a Service* (SaaS), *Platform as a Service* (PaaS) and *Infrastructure as a Service* (IaaS), are based on the cutting point of the service stack at which provisioning is offered to the user. The subpart below the cutting point is the cloud domain solely under control of the cloud provider, whilst from the interface point on the service belongs to the cloud user. Fig. 1 depicts the relation between the service stack and the different service models, as well.

The service model has a huge impact on the responsibilities and the available possibilities cloud users have regarding fault tolerance, as he has no influence on the services below the cutting point, if it is a control domain of a separate provider.

3) Deployment Models of cloud computing: [2]

The deployment models define the nature of the cloud provider. Self-operating a cloud (*Private cloud*) or part of it (*Hybrid cloud*) gives insight and control into the cloud providers domain, thus enables fault tolerant activities in a larger part of the service stack. On the other hand, cloud environments operated by an organization (*Public cloud*) or a group of organizations (*Community clouds*) lack of such extended observability and controllability into the underlying layers.

B. Software-Defined Network [3]

In essence, *Software-Defined Network* (SDN) relies on two core ideas: separation of the control and data planes of the data transmission, and abstracting network control into application programming interfaces. These two concepts allow higher level programming of networks and give greater flexibility over traditional networking solutions.

SDN greatly enhances the agility of network management. It fundamentally complements the designated cloud computing characteristics of on-demand self-service and rapid elasticity, providing even greater freedom in resource configuration.

C. Scope of the paper

The IaaS service model deployed on a public cloud is the primary scope of this paper. This means that a cloud provider is solely responsible for maintaining the physical infrastructure hidden from the users, who submit requests regarding the resources.

Network virtualization [4] allows cloud users to run their own *network operating systems*, offering full control over the virtu-

al networks they define and use. Simultaneously, cloud providers can hide the peculiarities of their physical infrastructure by mapping it to virtual networks.

III. FAULT TOLERANT DESIGN PATTERNS

Software design patterns extract the reusable abstract design information of proven solutions to recurring common problems within a context. They also highlight the reasons why the chosen solution is appropriate.

Patterns for fault tolerant software target improved *resiliency*. They offer general purpose solutions applicable at different levels of the architecture to ensure undisrupted processing. [5]

A. Physical cloud layout

Cloud resources are usually hosted in multiple locations named *regions* providing geographic diversity. [6] *Availability zones* are physically isolated locations within regions interconnected through low-latency links. Inter-region communication is done via Internet, implying higher cost, lower security and quality.



Fig. 1. Service stack, cloud service models and alternatives for the redundancy architectural pattern

The *redundancy architectural pattern* relies on the use of multiple separate copies of a resource in order to ensure fault tolerance. It can be applied at several layers as depicted in Fig. 1. For instance, hardware redundancy mostly resides in the domain of the cloud provider. In contrast to this, the replication of VMs relying on the same design pattern is within the domain of both the cloud user and cloud provider.

B. Phases of fault tolerance

Fault tolerance has four phases: *error detection*, *error recovery*, *error mitigation* and *fault treatment*.

- Error detection aims at finding errors within a system state.
- Error recovery aims at returning an erroneous system to an error free state by applying correcting actions that eliminate the error.
- Error mitigation aims at adapting a system to maintain operation in the presence of errors.

• Fault treatment aims at fixing the system by removing the faults, the causes of errors.

Patterns for fault tolerant software are grouped into five categories. Four groups are aligned along the four phases of fault tolerance, while the fifth group is based on architectural design.

C. IaaS level fault tolerance patterns

A few examples are given in this short section to showcase the high level applicability of fault tolerant patterns.

VMs are among the primary units of resources in IaaS, therefore they represent a natural candidate for the role of unit of fault tolerance. (*Units of Mitigation architectural pattern*)

Operating multiple VMs concurrently (*Redundancy architectural pattern*) and switching between them (*Failover error recovery pattern*) is also an obvious idea. Moreover, an example of the multiple applicability of the redundancy architectural pattern was already depicted in Fig. 1.

Storing information independent from the VMs is a regular cloud service, thus the *Remote Storage error recovery pattern* is also a natural candidate.

IV. REDUNDANCY AND FAILOVER PATTERNS

Fault tolerance aims at the maintenance of the continuous operation of a service despite of faults. Failover assures fault masking by substituting a faulty resource with a fault free backup one. This substitution should be seamless and invisible to the services utilizing the corresponding resource.

Failover can be realized in several ways: resource reallocation, task migration or the re-arrangement of the system topology.

For instance, an external entity could act as a *proxy* [7] (loadbalancer [8]), initiating a failover by task reallocation upon detecting an error. However, fast network reconfiguration facilitated by SDN is a new candidate for fault tolerance. In this case the network can execute the failover by rearranging the network routing topology. A similar idea, named *network topology transparency*, could be found within a draft of Network Function Virtualization. [9]

The tradeoff between the different redundancy and failover solutions should be resolved using a cost analysis and the loss function associated with downtime (service outage).

A. Availability considerations

Downtime (service outage) is the duration when the service is unavailable. Fig 3 depicts the downtime to be the duration of repair, the time between the crash of the primary virtual machine and the activation of the secondary virtual machine.

The execution time of resource reconfiguration becomes extremely important in systems based on dynamic resource and redundancy allocation. Although on-demand self-service and rapid elasticity are essential features of cloud computing, the level of agility, the speed of reconfiguration (scaling) is far from deterministic. For instance, typical virtual machine startup may vary between 100 and 800 seconds [10]. Moreover, a more detailed study [11] aimed at modeling the overhead of launching virtual machines concluded that the total overhead has an even larger variation. The reconfiguration time of SDN nodes also have huge variation. The installation of a new *forwarding rule* generally takes between 2 and 12 milliseconds, but the overall duration can reach the order of seconds. This can occur due to the latency of *packet in* and *packet out* messages in case of a saturated control plane. [12]

The cumulative effect of these variation factors may lead in worst case to critical degradations in service due to the delayed failover actions despite of the availability of a sufficient redundancy.

1) Stateless failover

Stateless failover does not need transmission of any state information from the failed to the backup node, thus the main contributors to downtime are network reconfiguration and activation of the backup node. The use of hot (active) backups eliminates the latter one.

2) Stateful failover

Stateful failover necessitates the transmission of state information from the primary to the backup node. This state transfer significantly affects downtime. Standard virtualization has a significantly varying impact on network, thus on state transfer performance [13]. State information can be transferred directly between VMs or indirectly by means of an intermediate shared resource.



Fig. 2. State transfer alternatives

Direct state transfer (synchronization) requires active backup VMs, thus operates with a continuous overhead of running redundant nodes. (Fig. 2) This solution eliminates state transfer during a failover.

Indirect state transfer requires a mediating entity, usually a remote storage (*Remote storage error recovery pattern*, Fig. 2). This solution does not require an active backup, thus avoids the ongoing expense of the active backup nodes, but the duration of state transfer is significant. Naturally, any combination of the methods is feasible.



Fig. 3. Downtime and reconfiguration durations

d1: backup creation duration

d2: network reconfiguration duration

d3: state transfer duration

Fig. 3 presents the relation between *downtime* and the different *reconfiguration actions*. Backup VM creation and state transfer run sequentially, whilst network re-configuration runs in parallel with them. With an active backup d1 becomes 0, without state transfer d3 becomes 0 too. (Table I) The total recovery time is heavily application dependent and the best mechanisms are subject of further research.

Redundancy / Mode	Downtime		
Stateless / none	Max (d1,d2)		
Stateless / 1 active	d2		
Stateful (synchronizing) / 1 active	d2		
Stateful (synchronizing via remote	d2		
storage) / 1 active			
Stateful (remote storage) / none	Max (d1+d3,d2)		
Stateful (remote storage) / none	Max (d1+d3,d2)		

TABLE I.	DOWNTIME EVAULATION OF REDUNDANCY MODES
----------	---

3) State to be transferred

State can be transferred many ways with different limitations: application data, the application, and the virtual machine just to name a few. For example, *offline migration* of VMs is possible among different architectures, while *live migration* of VMs is not, as transferring of memory pages requires binary compatibility.

B. Pricing model

The efficiency of redundancy mechanisms increases with the independence of the primary and backup resources. For instance, a backup deployed in another region is insensitive to faults affecting the region of the primary one. However, state storage and transfer between primary and backup nodes may differ essentially depending on the resource allocation from the cost point of view.

An initial investigation of the basic features of the IaaS pricing model is presented in this section, because a low *total cost of ownership* (TCO) of redundancy is uttermost desirable.

The *pay per use* pricing model is generally used, but with different pricing policies by the different cloud providers. The pricing model of Amazon EC2 [14], an industry leader, is selected as a reference.

Basically, Amazon EC2 has four categories of virtual machine setups: general purpose, compute optimized, memory optimized, storage optimized. The costs of the resources scale linearly within each category. However, the relative costs of resources between categories differ.



Fig. 4. Storage and data transfer cost comparison

Data transfer within the same availability zone is either free or low cost; inter availability zone data transfer has moderate cost, while external data transfer to the Internet, including another cloud region, is more expensive.

Remote storage pricing is based on the total amount of stored data per month. However, provisioned storage costs extra for the guaranteed input/output operations per second available to the user. Remote storage is tied to an availability zone and inter zone migration is charged as inter zone network data transfer. Fig. 4 depicts the price of a gigabyte of data stored for a month and the same amount transferred out to the internet. The extra price for provisioned storage is omitted. It is visible, that storing a gigabyte of data for one month costs similar to transferring it out to the internet.

C. Cost analysis

 TABLE II.
 Cost factors (0=Not Applicable, 1=Should be applied)

Redundancy / Mode	VM Cost	State Storage 0,05-0,15 [\$/GB-month]	State Transfer 0-0,2 [\$/GB]
Stateless / none	0	0	0
Stateless / 1 active	1	0	0
Stateful (synchronizing) / 1 active	1	0	1
Stateful (remote storage) / 1 active	1	1	1
Stateful (remote storage) / none	0	1	1

The different redundancy modes have different ongoing costs. Having an active VM costs hourly. Storing and transferring state across availability zones and regions also costs money.

V. CONCLUDING REMARKS AND FUTURE WORK

The current paper introduced the problem, highlighted cloud computing features relevant from resiliency point of view, and evaluated a selected set of patterns respective to downtime and cost. As future work, a deeper analysis of the selected patterns is planned along with the investigation of other patterns.

REFERENCES

- [1] Network Functions Virtualisation Introductory White Paper
- [2] Peter Mell, Timothy Grance, The NIST Definition of Cloud Computing
- [3] Software-Defined Networking: The New Norm for Networks
- [4] OpenVirteX (2015.01.29): http://ovx.onlab.us/
- [5] Robert Hanmer "Patterns for Fault Tolerant Software"
- [6] Amazon AWS documentation (2015.01.29): http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regionsavailability-zones.html
- [7] HA Proxy (2015.01.29): http://www.haproxy.org/
- [8] ELB (2015.01.29): http://aws.amazon.com/elasticloadbalancing/
- [9] gs_NFV-REL001v010101p Resiliency Requirements (2015.01.29): http://docbox.etsi.org/ISG/NFV/Open/Published/
- [10] Ming Mao, Marty Humphrey, "A Performance Study on the VM Startup Time in the Cloud"
- [11] Hao Wu, Shangping Ren, Gabriele Garzoglio, Steven Timm, Gerard Bernabeu, Seo-Young Noh, "Modeling the Virtual Machine Launching Overhead under Fermicloud"
- [12] Brent Stephens, Alan Cox, Wes Felter, Colin Dixon, John Carter "PAST: Scalable Ethernet for Data Centers"
- [13] Guohui Wang, T. S. Eugene Ng "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center"
- [14] EC2 pricing model (2015.01.29): http://aws.amazon.com/ec2/pricing/

Requirements towards a formal specification language for PLCs

Dániel Darvas^{*†}, István Majzik^{*} and Enrique Blanco Viñuela[†] *Budapest University of Technology and Economics, Department of Measurement and Information Systems Budapest, Hungary, Email: {darvas,majzik}@mit.bme.hu [†]European Organization for Nuclear Research (CERN), Engineering Department Geneva, Switzerland, Email: {ddarvas,eblanco}@cern.ch

Abstract—One of the main obstacles of using formal verification for complex PLC (Programmable Logic Controller) programs is the lack of formal requirements. There are no widely used specification methods that could serve as input for formal verification; also that could help the developers to capture the behaviour and handle the complexity of these programs.

The goal of this research is to bring formal specification closer to the PLC domain in order to help the development, verification and maintenance. This paper aims to briefly overview the particularities of the PLC domain and the state of the art in formal specification. Then it collects the requirements towards a PLC-specific formal specification language based on general works, comparative case studies and own experiences at CERN. Also, it draws up a sketch of a possible specification method that follows the collected requirements.

I. INTRODUCTION AND BACKGROUND

Programmable Logic Controllers (PLCs) are robust industrial computers providing standard solutions for control systems. For this discussion, these computers can be treated simply as computers with limited resources and a large number of low-level inputs and outputs. Physical inputs and outputs are represented as input and output variables in the programs. The execution of the user program is mainly cyclic: in each scan cycle (1) the input values are sampled and stored in the memory, then (2) the user program is executed, next (3) the computed output values are written to the physical outputs. The values of the input variables and physical outputs are both stable in the memory during the computation phase. The user program can also store states, thus the outputs are depending both on the current and previous input values. The programming languages are defined in the corresponding IEC 61131-3 standard [1].

A. Motivation

The motivation of this work is coming from CERN (European Organization for Nuclear Research) where most research facilities, like the Large Hadron Collider (LHC) rely on a multitude of PLCs. To cope with the complexity and to reduce the maintenance needs, most of the PLC programs at CERN are developed using the UNICOS framework [2]. This framework provides a design methodology, code generation tools and a library of *base objects*. These base objects are not further decomposed in the implementation, but their complexity is still high. Therefore the (re-)use of these base objects needs a precise specification. Besides the usage, as all the systems at CERN are depending on these objects, their modification

is a critical task that requires high level of understanding of the underlying logic to avoid the negative collateral effects. These reasons imply a vital need for a suitable specification method that is neither available yet at CERN, nor in the PLC community in general. Previous work aiming to apply formal verification to these base objects [3] also emphasized the need for a good, formal specification method. Currently the main obstacle to use model checking on these PLC programs is the lack of precise, formal requirements to be checked, not the lack of available efficient verification tools.

The rest of the paper is structured as follows. Sect. II overviews the main requirements towards the specification language to be proposed taken into account the peculiarities of the PLC domain and based on the literature overview. Sect. III discusses already existing specification methods related to this work. Sect. IV sketches up the concepts of a new PLC specification language. Finally, Sect. V summarizes the paper.

II. PLC SPECIFICATION LANGUAGE REQUIREMENTS

The goal of this research is to overcome the previously mentioned issues by proposing a suitable PLC specification language. First the domain-specific requirements are summarized, then previous work is discussed to gather requirements towards a good, useful formal specification language.

A. Domain-specific requirements

The motivation of this work is a real insufficiency in the PLC domain, therefore it is indispensable to address the existing problems and particularities. Many specification languagerelated requirements can be extracted from the previously developed programs and by discussing the problems with the developers. The main domain-specific needs and challenges are summarized below based on the experience gained at CERN.

a) Events: Although the execution of PLC programs is cyclic and not event-triggered, the concept of events still exists (in a latent way). In fact, many Boolean inputs represent events or external actions aiming to modify the internal state of the controller. Events should be treated as "first-class citizens" in the specification.

b) Event semantics: It is also important to adopt a semantics that is appropriate for the PLC domain. Due to the cyclic behaviour, we cannot have the assumption usual in event-triggered systems that all previous events are fully handled before a new event is triggered. As in a PLC multiple

events can happen simultaneously, the priority of events has to be defined. If multiple *contradictory* events happen, the one with the highest priority should suppress the events with lower priorities, but several *independent* events can trigger in the same cycle.

c) Clean core logic: PLC programs work directly with physical input and output signals, therefore a significant part of the programs has to perform *input and output handling*. While this task is unavoidable, decoupling the I/O-handling helps to focus on the core logic. As the PLCs have limited resources, the developers try to minimize the number of variables, but this approach is not needed to be followed in the specification. In the specification, it is important to use "concepts" rather than expressions, i.e. it is better to *define internal variables* (variables for specification purposes only) defined by expressions on the input variables, and to use these internal variables in the definition of the logic.

d) Hierarchical, modular structure: To support reuse and the abstract design of specification, the method should provide a hierarchical, modular structure. Hierarchy and modularity helps to "divide and conquer", to have a simple logic in the leaf modules (i.e. modules that are not further decomposed), and to avoid duplicated specification of same submodules.

e) Multiple formalisms: The leaf modules of the hierarchical structure should be specified by a specification language adapted to the behaviour of the current module, thus multiple module definition formalisms are needed. Among others, control-oriented, data-processing-oriented, as well as timing-related behaviours shall be considered. State machines and logic circuits are widely used in the development and specification of PLC program modules, but these can only be considered as "explanatory doodling", not as precise specification, as their formal semantics are not defined. New, formal languages should be proposed that are adapted to the specialities of the domain and based on the existing knowledge and practice of the developer community.

f) Limited expressivity: Rich languages can provide rich features, but also more space for problems and they can require longer training period. The expressivity of a PLC specification language should be restricted. In some cases, some of the restrictions can be explicitly relaxed, but then the verification of these parts should be carried out with special attention.

g) Time-dependent behaviour: As PLCs can have timedependent behaviour, the proposed formalism should support timed models. This behaviour is generally captured by timers in PLCs. Three kinds of timers are defined in the corresponding standard [1]: TP (signal pulsing), TON (on delay), TOFF (off delay). Each has a different semi-formal semantics. These standardised timers should be part of the language as modules, because they are widely used and well-known by the automation engineers.

B. Requirements from the Literature

Besides the experience from practice, previous work in the literature can point to necessary requirements and best practices of developing a new, domain-specific specification language. In this part some general work are summarized. In the first place, a (formal) specification method should satisfy obvious general requirements, e.g. it has to be correct, unambiguous, consistent, verifiable [4].

In 2000, van Lamsweerde published a survey [5] on existing formal specification methods and a roadmap for the future. The conclusion of the paper is that "formal specification techniques suffer a number of weaknesses". Such weaknesses are e.g. the (1) limited scope (i.e. the specification can only capture a part of the system), (2) poor separation of concerns (i.e. the intended properties, the environmental assumptions and the properties of the application domain overlap), (3) too low-level ontologies, (4) high cost, and (5) poor tool support. The author states that future formal specification methods should be *lightweight* (i.e. not requiring deep formal methods expertise), at least partially domain-specific, structured, multiparadigm and multiformat (i.e. integrating multiple languages and letting the specifier use the best for the current needs, thus for each subsystem the most appropriate language shall be chosen, or different languages might be necessary for specifying functional and extra-functional requirements).

Knight et al. approached the question "Why formal methods are not used widely?" more practically. In [6] they designed an evaluation framework to assess formal specification methods. Furthermore, they selected three specification languages (Z, PVS, Statecharts) and applied them for a subsystem of a nuclear reactor. Then, the specifications were assessed by developers not expert in formal methods and by nuclear engineers. After a short training period, the general idea and the main advantages of formal specification were welcomed and understood. From the point of view of the nuclear engineers, Z and PVS were too complex, but Statecharts were claimed to be effective for communication and easy to learn, although difficult to search and navigate. The authors emphasize that often overseen features are also recommended to help the industrial usage. These comprise the support for documentation and readability, e.g. by including free-text annotations connected to the elements of the specification.

A possible reason why Statecharts [7] are often welcomed by the non-computer engineers can be the fact that it was not developed in a purely academic environment, but in strong collaboration with avionics engineers, taking their habits and knowledge into account [8].

A good example for Statechart-based languages is the RSML formalism. It was designed to help the specification process of the TCAS II avionics system. In [9] the authors discuss some lessons learned, like simplicity and readability are "extremely important" [9]. Based on the experience from RSML, a new language (SpecTRM-RL) was created, but both seem not to be used widely. Also, the provided solutions do not fit to the PLC domain and using only a Statechart-like formalism does not provide a convenient method for specifying PLC programs, e.g. behaviour of modules with many numeric state variables are difficult to be captured.

A recent work on the topic collects requirements towards a model-based requirements engineering tool for embedded systems [10]. In their survey the highest ranked requirements were the support for various different representations, document generation, expression of non-functional requirements, and for maintaining traceability links; not formal verification or automated code generation.

The literature overview briefly summarized above pointed out that a good specification should be *lightweight* [5], [6]. This also implies the need for *domain-specificity*. The specification should be adapted to the concrete (sub)system being specified, therefore a "*portfolio of languages*" is needed, of which the most appropriate can be chosen for each submodule [5]. Similarly, the functional and extra-functional properties need different languages.

III. EXISTING, RELATED SPECIFICATION METHODS

The motivation for a new specification language is coming from CERN, but we believe that the problem is more general. There is no widely-accepted, formal or semi-formal specification method for PLC programs. The IEC 60848 standard [11] defines a specification language based on finite state machines called *Grafcet*. This language can be seen as a safe Petri net extended with guards and variable assignments. Although in some cases Grafcet can capture the behaviour of some submodules, it is not applicable generally. Also, the semantics of the language is complex and sometimes not intuitive, furthermore it is confusing that a standardised PLC implementation language, the SFC uses similar syntax with different semantics [12].

Other work also targeted the formal specification of PLC programs. In [13] the authors describe a specification method called *ST-LTL* based on LTL (Linear Temporal Logic). In fact, this is a formal language that can be easily checked on the implementation. On the other hand, the formalism is far from the automation engineer's general knowledge, also it can be difficult to scale with the growing size and complexity, furthermore it is not obvious to see if the specification is inconsistent or incomplete.

The formal specification methods are not widespread in the PLC domain yet, but many formalisms are widely used in computer engineering, e.g. the B Method, the VDM-SL, the Z notation, or temporal logics. An obvious solution would be to use one of them. However, it is not really a suitable solution, as these methods are typically (1) too difficult to be used for engineers not trained specially in formal verification, and (2) the usage is even more difficult in the case of PLC programs, as the methods are not adapted specifically to the PLC semantics, even the general properties of the PLCs (e.g. cyclic execution) have to be explicitly defined, therefore it is difficult to address the domain-specific requirements discussed in Section II-A.

A good example for formal specification from the industry is the SCADE Suite by Esterel Technologies/ANSYS. It provides a model-based environment to support the development of critical embedded software, including an automated code generator compliant to various industrial standards. However, even if its cyclic computation model is close to the PLC computation model, this toolset does not provide solutions specifically for IEC 61131-compliant PLCs.

The B, Grafcet, SCADE, Spec-TRM, ST-LTL, VDM-SL, and Z methods are compared in the Table I based on some of the main requirements discussed in Section II-A and II-B. As can be seen, none of the tools/methods provide a solution

		B / Z	Grafcet	SCADE	Spec-TRM	ST-LTL	VDM-SL
	Precise meaning	+	+	+	+	+	+
re	Lightweight	-	+	+	+	-	-
sral	Specific for the PLC domain	-	+	-	-	+	-
ene	Support for documentation	-	-	+	+	-	_1
9	Available tool support	+	+	+	-	-	+
	II-A a) Events	_1	+2	+	+2	-	_1
•.	II-A b) Event priorities, suppressions	-	-	_	-	-	_
req	II-A c) Clean core logic	-	_	$+^{2}$	-	-	_
SL	II-A d) Hierarchical, modular structure	+	$+^{2}$	+	+	+2	_
Ä	II-A e) Multi-formalism	-	-	+	-	-	_
	II-A g) Time-handling	-	+	+	$+^{2}$	+2	_1
1		-					

 1 Not supported in the original formalism, but supported in one of its extensions. 2 With restrictions.

that is adapted to the PLC domain, satisfies most of the domain-specific and the general requirements at the same time. Therefore we think that none of these methods could be integrated to a PLC development process with low cost.

IV. MAIN CONCEPTS OF THE PROPOSED SOLUTION

Based on the requirements and experiences discussed before, a first version of a specification language suitable for PLC programs is defined here. Only a short sketch of the method is provided, not an extensive description. The starting point of the proposed solution is the Statechart formalism, as it is intuitive and generally close to the (automation) engineers. However, it has to be extended and altered, for the following reasons.

- 1) The Statechart formalism is not a complete specification method, it just captures the behaviour of a part of the system. It was intended to be embedded in a broader framework [8].
- 2) The common semantics definitions do not fit entirely to the PLC properties. For example, the "events" are represented by input variables and the program execution is cyclic, therefore it cannot be assumed that only one event will be processed at a time, as it is assumed for example in the semantics of the UML (Unified Modeling Language) State Machines.
- 3) Not every stateful behaviour can be captured efficiently by state machines, e.g. modules with integer or floating-point state variables. This is common in process control programs, thus a Statechart-based formalism should be accompanied by other languages.

A. Structure (High-level Syntax)

The base element of the proposed solution is the *module*. A module is either a *composite module* that is refined by submodules, a *leaf module* capturing the behaviour of a part of the system, or an *alternative module* that helps to choose between multiple implementations (e.g. timed or non-timed implementation) based on some parameters.

In every module it is possible to define internal variables based on the input variables and outputs based on the current state of the module. In leaf modules events can also be defined. This decoupling of I/O-handling and computation helps to keep the *core logic* as small as possible. The scope of the internal



Fig. 1. Metamodel of the module structure

variables is by default restricted to the defining module and its descendants, but this restriction can be explicitly relaxed. The core logic of the leaf modules can be defined using one of the three defined formalisms, adapted to the logic it is implementing. The first possibility is to use a well-defined, restricted *statechart* describing the logic that allows to use hierarchical states (without regions), transitions (with guards or external triggers but without variable assignments, actions and internal events), and history nodes.

If it is not efficient or not possible to use statecharts, because for example the module is using integer state variables, an *input-output connection module* can be used. The representation contains all input variables that can be used in the computation and all output variables that have to be assigned by the given module. The logic description consists of defining different stateless computation blocks and the directed connections between the variables and block inputs and outputs. In this way the assignment rule of each output variables can be given graphically. Furthermore, it is easy to enforce that each output variable should be assigned exactly once in each cycle. The module structure explained above is summarized by its metamodel in Fig. 1.

It is necessary to give multiple possible syntax not only for describing the core logic, but also for describing the input and output definitions. In simple cases, ordinary Boolean expressions can be used. In more complex cases, the AND/ORtables (like the ones defined in [9]) are more efficient and less error-prone.

In the proposed solution, every element can be annotated by the user to make easier the understanding for example by providing important explanations. The annotations can also help to use the specification (or an artefact generated from the specification) as documentation for maintenance and reuse.

B. Execution (Informal Semantics)

The execution of each module consists of 4 phases. It is started by (1) computing the defined input expressions and the (2) enabled events. Next, the (3) logic of the module is executed. The execution of each module is finished by (4) computing the values of the defined output variables.

The execution of the logic depends on the applied formalism in the case of leaf modules. For composite modules, this means the *sequential* execution of the submodules, in a predefined order. The execution of an alternative module consists of executing the module corresponding to the evaluated condition value. In a statechart module, the execution means the exhaustive firing of all enabled transitions that are not triggered by any event. Then at most one event is selected that is enabled and not suppressed by any higher priority event, and at most one transition fires that is triggered to this selected event. After that all enabled non-triggered transition fires. The execution of an input-output connection module is the iterative evaluation of all defined signals starting from the input variables and the previous values of output variables, until the new values can be assigned to the outputs.

V. SUMMARY AND FUTURE WORK

This paper described a first step towards a new formal specification language for complex PLC programs. The goal of this research is to provide a specification method tailored to the PLC domain. This paper sketched up the main concepts of a potential specification language based on (1) the general requirements towards formal specification methods, (2) the specialities of the PLC domain, and (3) the experienced challenges and difficulties.

Future work includes the design and description of formal syntax and semantics of the language. This should be followed by providing tool support to construct the specification, to verify its consistency and to check the conformance of PLC programs in regard to the specified behaviour. After an experimental phase and collecting feedback from its users the specification method is intended to be introduced in the PLC development workflow of CERN.

REFERENCES

- IEC 61131-3:2013 Programmable controllers Part 3: Programming languages, IEC Std., 2013.
- [2] E. Blanco Viñuela et al., "UNICOS evolution: CPC version 6," in Proc. of the 12th Int'l Conf. on Accelerator & Large Experimental Physics Control Systems, 2011, pp. 786–789.
- [3] B. Fernández Adiego, D. Darvas, J.-C. Tournier, E. Blanco Viñuela, and V. M. González Suárez, "Bringing automated model checking to PLC program development – A CERN case study," in *Proc. of the 12th Int'l Workshop on Discrete Event Systems*. IFAC, 2014, pp. 394–399.
- [4] IEEE Std 830-1998 Standard, IEEE Computer Society Std., 1998.
- [5] A. van Lamsweerde, "Formal specification: A roadmap," in *Proc. of the Conf. on The Future of Software Engineering*. ACM, 2000, pp. 147–159.
- [6] J. C. Knight, C. L. DeJong, M. S. Gibble, and L. G. Nakano, "Why are formal methods not used more widely?" in 4th NASA Langley Formal Methods Workshop, 1997, pp. 1–12.
- [7] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231–274, 1987.
- [8] D. Harel, "Statecharts in the making: A personal account," in Proc. of the Third ACM SIGPLAN Conf. on History of Programming Languages. ACM, 2007, pp. 5–1–5–43.
- [9] M. Heimdahl, N. Leveson, and J. Reese, "Experiences from specifying the TCAS II requirements using RSML," in *Proc. of the 17th AIAA/IEEE/SAE Digital Avionics Systems Conf.*, vol. 1, 1998, pp. C43/1–C43/8.
- [10] S. Teufl, M. Khalil, and D. Mou, "Requirements for a model-based requirements engineering tool for embedded systems: Systematic literature review and survey," fortiss GmbH, White Paper, 2013.
- [11] IEC 60848:2013 GRAFCET specification language for sequential function charts, International Electrotechnical Commission Std., 2013.
- [12] J. Provost, J.-M. Roussel, and J.-M. Faure, "A formal semantics for Grafcet specifications," in *IEEE Conf. on Automation Science and Engineering*, 2011, pp. 488–494.
- [13] O. Ljungkrantz, K. Åkesson, M. Fabian, and C. Yuan, "A formal specification language for PLC-based control logic," in *Proc. of the* 8th IEEE Int'l Conf. on Industrial Informatics, 2010, pp. 1067–1072.

Preprocession and Interpretation of Multielectrode Array Recordings

Dorottya Cserpán^{*}, Zoltán Somogyvári[†], Gábor Horváth^{*} *Budapest University of Technology and Economics, Department of Measurement and Information Systems, Budapest, Hungary. Email: {cserpan,horvath}@mit.bme.hu [†]Wigner Research Center for Physics, Department of Theory, Budapest, Hungary. Email: somogyvari.zoltan@wigner.mta.hu

Abstract—The recent blooming of the Multi-electrode array (MEA) technics made it possible to measure the extracellular potential with a tens of micrometers and few tens of kHz spatial and temporal resolution. While most common use of these recordings is to analyse the local field potential or the current source density (CSD) distribution and to get the spiketrains of neurons, our goal was to build a tool for estimating a detailed spatiotemporal description of the electro-physiological processes in a single cell level. The motivation of this work is the lack of electrophysiological imaging methods at a whole cell-level. We present two methods depending on whether the morphology of the cell is known or not. By applying these methods on simulated or experimental data we show their performance and limitations.

I. INTRODUCTION

Almost 100 years passed since the recordings of the first human electroencephalography were carried out by Hans Berger. Since then the electrode technology has improved, electrodes with much higher temporal and spatial resolution are available, both for in vivo and in vitro, invasive and noninvasive experiments. The potential recorded by an electrode placed into the extracellular space in the brain or in a slice is called extracellular potential (EC). The EC at a point is the superposition of the potentials generated mainly by the membrane currents of neurons, so it has big significance in investigating neural activity. Frequencies less than 300 Hz, the local field potential (LFP), typically are related to synaptic activities, neural oscillations, the higher ones to firing activities, which is called the multi-unit activity regime (MUA). Further divisions of the lower frequency neural oscillations exist, which emerge as population activities and are connected to various cognitive states. Although the extracellular potential is very informative in many senses, it is not straightforward to interpret how it is connected to the synaptic membrane currents. In case of multi-layer structures, as the neocortex or the hippocampus, the Current Source Density (CSD [1]) method connects the the quantities. The inverse CSD [2] and kernel CSD [3] methods generalized and expanded the applicability of the CSD method. The spike CSD (sCSD [4]) method was the first to raise the question of restoring the membrane current source density distribution on a single cell level. This method is well applicable to neurons possessing an elongated morphology, like pyramidal neurons. In this paper two other methods will be shown with similar aims as the sCSD. One of them, the spherical CSD, is a method to use in case of neurons with spherical shape with the soma in the center, like thalamical relay cells. As the palette of the morphologies of neurons are very colourful, the other method is developed for arbitrary morphology, but it has to be known. The feasible technology to reconstruct the morphology after recording is done by colouring a cell close to the electrode with flourescent dye and doing automated recovery providing the coordinates later on. Unfortunately due to the lack of appropriate experimental data, the ksCSD method was only tried on simulated data, but the spherical CSD method was applied on real recordings as well.

II. PREPROCESSION

As the spherical and kernel spike CSD methods calculate the membrane current source density distributions of single cells, the extracellular potentials related to these need to be extracted from the raw data. It is widely accepted, that the so called spike, which is the extracellular potential evoked by the firing activity at the soma, looks slightly different for each cell, but the same for one cell while firing, hence is a good basis for clustering. Each cluster shall contain spike shapes belonging to one neuron. To detect the spikes, the raw data was band-pass filtered between 500-3000 Hz by using a 3rd order Butterworth filter and a detection threshold was set. Clustering was performed on the first 6 PCA components of the spike shapes and by using the Gaussian mixture model algorithm called Mclust in implemented in R [5].

A spike-triggered potential average can be calculated and if there are multiple electrodes, this gives a spatial distribution also. The current data was recorded by a linear probe with 32 electrodes each 50 μ m from each other and a 20 ms time window was used resulting 1600 um x 20 ms spatio-temporal potential map.

Furthermore coherence values in several frequency regimes between the different extracellular potential time series recorded by various electrodes were calculated to identify the functionally connected regions.

III. CURRENT SOURCE DENSITY METHODS

The relationship between the current sources and the generated potentials is given by the Poisson-equation:

$$\sigma \nabla^2 \Phi(\mathbf{r}, t) = -C(\mathbf{r}, t), \tag{1}$$

where σ is the electrical conductivity of the extracellular medium, Φ is the extracellular potential, **r** refers to the position, and $C(\mathbf{r},t) = \sum_{n=1}^{N} I_n(t) \delta^3(\mathbf{r} - \mathbf{r_n})$ is the current source density, the summation goes over all point sources,the

position of the n^{th} current source is ${\bf r_n}$. There are several methods which use different assumptions for solving the above mentioned equation.

A. Traditional CSD

The traditional CSD method uses the recorded extracellular potential from a laminar electrode, which is placed perpendicularly (z direction) to the layers of the cortex. Based on this setup and assuming that the layers are infinite and homogeneous, the current source densities of each layer can be given:

$$C(z_j) = -\sigma \frac{\Phi(z_j + h) - 2\Phi(z_j) + \Phi(z_j - h)}{h^2}$$
(2)

where z_j is the position along the z-axis of the j^{th} electrode and h is the inter-electrode distance.

B. Kernel spike CSD

The CSD distribution at point \mathbf{x} can be expressed as the sum of the M current sources:

$$C(\mathbf{x}) = \sum_{j=i}^{M} a_j \tilde{b}_j(\mathbf{x})$$
(3)

 \tilde{b} are the basis functions overlapping each other and are distributed evenly along the morphology of the cell, a_j is a multiplication constant. The morphology of the cell was described with several curves, so called "branches". The curves in the 3 dimensional space can be parametrized with variable t.

$$x = f_x(t)$$

$$y = f_y(t)$$

$$z = f_z(t)$$
(4)

Let us consider Gaussian basis functions:

$$\tilde{b}_i(t') = e^{-\frac{(t'-t_i)^2}{R^2}}$$
(5)

where both $t, t_i \in [0, d]$ are parameters on the same branch which has a length of d. Here R is the double of the variance of the Gaussian function. The connection between the current source densities and potentials Φ is introduced by the Aoperator($A : \tilde{F} \to F$)

To determine the CSD distribution in arbitrary positions(x), the following kernel functions were introduced:

$$K(\mathbf{x}_k, \mathbf{x}_l) = \sum_{i=1}^M b_i(\mathbf{x}_k) b_i(\mathbf{x}_l)$$
(6)

$$\tilde{K}(\mathbf{x}_k, \mathbf{y}_l) = \sum_{j=1}^M b_j(\mathbf{x}_k) \tilde{b}_j(\mathbf{y}_l)$$
(7)

Using the simulated or measured extracellular potentials (V) and assuming \tilde{K} is invertible the solution for C is straightforward.

$$C(\mathbf{x}) = \tilde{\mathbf{K}}^T(\mathbf{x})\tilde{\mathbf{K}}^{-1}\mathbf{V}$$
(8)

Original Current Density Distribution















Fig. 1. Simulation results showing the performance of the ksCSD method. Y-shaped morphology was used with 4x4 recording sites with a 50 μ m cell-to-electrode distance. The 3 upper colormaps show the original current source density distribution, its spatial smoothed version in case of the best parameter settings. The second plot from the bottom shows the error in different time instances in case of the best parameter setup. The plot in the bottom indicates, that using more basis functions (M) improves the performance with an optimal width (R), which is around 30 μ m in this case.

The cell and the electrode



Fig. 2. Setup for the LFPy simulation. The red lines mark the morphology of the cell, black squares show the spatial position of the electrodes. A multicompartmental neuron model with dynamics was used to calculate the membrane currents and the extracellular potentials. Artificial synaptic connections were used to mimic the synaptic activity. Besides the Y-shaped morphology, a simpler (ballstick) and more complex ones were also used.

C. Testing the method on simulated data

As there are several parameters used in the simulations, like the width and number of used base functions, the number and arrangements of the electrodes, the cell-electrode distance and the morphology of the cell, it is impossible the study all cases. Therefore some special cases in terms of morphology were chosen for studying the behaviour of the method. Unfortunately experimental data for the validation of the ksCSD method is not available yet, I used the LFPy [6] simulation environment, which given a certain cell morphology and parameters describing the electrophysiological properties of the cell, simulates the membrane potential, membrane currents and extracellular potential both in space and time. On the simulated extracellular potential I use the ksCSD method to give an estimation for the membrane currents, as these are known from the simulation, it is possible to calculate the error of the method with the following error measure:

$$\epsilon = \frac{\sum_{i} \sum_{t} |C_{i,t} - C_{i,t}^{0}|^{2}}{\sum_{i} \sum_{t} |C_{i,t}^{0}|}, \qquad (9)$$

where C^0 denotes the membrane currents calculated by LFPy, C stands for the ksCSD estimated current, the summation goes for all segments *i* and all time snaps *t*. One of the main results is the comparison of the estimated currents with the the original membrane currents by using the above mentioned measure, which is shown on Figure 1. On the simulated data random inputs arrive onto various segments, part of the dendrite tree. It was clear from the beginning, that the ksCSD method is not able to reach that high spatial resolution, in induces spatial smoothing. Still, making a difference between completely bad an not as good reconstruction is needed, so a spatially



Fig. 3. An example for thalamical cell with spherical symmetry and a schematic plot for the spherical shell model

smoothed version of the simulated data was also calculated for the comparison. It is visible, that the reconstructed colormap shows a high level of similarity to the previously mentioned smoothed version. As expected, applying the ksCSD method for cells with simpler morphologies and using more electrodes improves the performance. It seems, that there is an optimal width for the basis functions, but increasing the number of them does not decreases the error significantly in many cases.

D. Spherical CSD

The thalamical cells typically have a spherical symmetry with the soma in the center. Proximal and distal parts on the dendrite tree of the cells are distinguishable based on the location of the synaptic inputs. Using this apriori information, we built a spherical shell model to solve the inverse problem. The main assumptions are, that the charge current source density distribution can be described with concentric shells. It is possible to write the potential in the form of the multipole expansion series. As a benefit of the spherical symmetry, the spherical harmonics can be used in the description, and calculating with these make easier to solve the Poisson-equation. Let's assume that each shell is measured at 2 points, which will provide enough information for calculating the monopole and dipole part of the the multiple expansion series, allowing asymmetry within a shell. The first shell is a sphere, the radius is equal with the distance of the soma center and the nearest electrode. Getting further from this sphere, each electrode-pair marks the border of a new shell, so on so on. The mathematical derivation will not be presented more deeply. An example can be seen on Fig. 3. The spike was detected on the 37th channel, the numbering of the channels in this case goes from 34 till 65, so 200 μ m spatial vicinity in both directions measured from the soma are shown, which is placed to 0. On the potential map, red mean a negative potential change compared to the reference. On the spherical CSD colormap the red colour marks flow of positive ions into the cell. In both colormaps the intense red are marks the firing activity. In some clusters not shown here, the dendritic forward- and backpropagations are observable. The patterns visible on the colormaps highly depend on the cell-to-electrode distance, on the amplitude of the spike and on the goodness of the cluster.

IV. CONCLUSION

The methods presented above aim to estimate the current source density distribution of single cells while using different apriori information. The ksCSD method assumes the



Fig. 4. An example for estimating the current source density distribution on a cell assuming, that it has a spherical symmetry. The raw, spike triggered average potential is shown at the top,the bottom plot is the spherical CSD. Event in the 200 μ m vicinity of the soma are shown in both direction paralell to the electrode. The main difference between the plots is the intense blue area at the spike (10 μ s) on the current map, which is a counter current of the current marked with red area. The yellow areas in the lower spatial location before the spike might indicate the synaptic excitations.

morphology is known, so the spatial resolutions of the result is much more detailed. The method is a little sensitive to the initial choice on the basis number and much more to the width of the basis function. Based on the simulation results, the method is able to capture the key patterns of the inputs, although it is limited in the spatial resolution. Higher resolution can be achieved with a denser electrode grid, but also the complexity of the neuron morphology matters. The simulations were noiseless, further investigation of the effect of noise is needed. Probably currents with not enough high amplitude or located far from the recording sites are not to be reconstructed. My future plan is to validate the method on experimental data once it is available.

The spherical CSD method can be used for cells with spherical symmetry, like the relay neurons located in the thalamus. Evidence for electrophysiological pattern emergence was shown. The future direction of this work is to investigate the behaviour of these cells under various stimuli, for example whisker or flash and compare the results. An other question to answer is what changes during up- and down states, which are different functional states in the neocortex strongly influencing this brain region.

ACKNOWLEDGMENT

I am indebted first of all to my supervisors Gabor Horvath and Zoltan Somogyvari and to my many colleagues from the Complex Systems and Computational Neuroscience Group at Wigner Research Center who supported me, furthermore to Daniel Wojcik who gave me the the opportunity to work in his Laboratory of Neuroinformatics at Nencki Institute. Furthurmore I am grateful to Laszlo Acsady and Peter Bartho for providing the data.

REFERENCES

- C. Nicholson and J. A. Freeman, "Theory of current source density analysis and determination of conductivity tensor for anuran crebellum," *J. Neurophysiol.*, vol. 38, no. 2, pp. 356–368, 1975.
- [2] K. Pettersen, A. Devor, I. Ulbert, A. M. Dale, and G. T. Einevoll, "Current-source density estimation based on inversion of electrostatic forward solution: Effects of finite extent of neuronal activity and conductivity discontinuities." *Neurosci. Methods*, vol. 154, no. 1-2, p. 116133., 2006.
- [3] J. Potworowski, W. Jakuczun, S. Leski, and D. Wojcik, "Kernel current source density method," *Neural Comput.*, vol. 24, no. 2, pp. 541–575, Feb. 2012.
- [4] Z. Somogyvari, D. Cserpan, I. Ulbert, and P. Erdi, "Localization of single-cell current sources based on extracellular potential patterns: the spike csd method." *Eur J Neurosci.*, vol. 36, no. 10, pp. 3299–3313, Nov. 2012.
- [5] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis and density estimation," *Journal of the American Statistical Association*, vol. 97, pp. 611–631, 2002.
- [6] H. Lindn, E. Hagen, S. Leski, E. S. Norheim, K. H. Pettersen, and G. T. Einevoll, "Lfpy: A tool for biophysical simulation of extracellular potentials generated by detailed model neurons," *Frontiers in Neuroinformatics*, vol. 7, no. 41, 2013.

Using Data Mining Techniques for Estimating Printed Circuit Board Design Complexity

András Pataki

Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary Email: apataki@mit.bme.hu

Abstract—Printed circuit boards (PCB) are indispensable parts of today's electronic equipments. Due to the customer's demands the size of PCBs has been shrinking, while the number of components to be mounted on these boards is still increasing. Because of highly competitive environment, it is necessary to deliver products on time and within budget. Besides that reliability and quality are the most important technical objectives. As a result of extremely high volumes (~1Mio pieces/year) the appropriate choosing of printed board technology can yield significant savings and profit caused by the cost reduction of PCB. These facts make the design and the manufacturing of boards even more complex and difficult. To reach the previous mentioned objectives in our approach we want to exploit the advantages of artificial intelligence techniques by extracting knowledge from projects in the past. The model will incorporate versatile PCB metrics. As a first step this article introduces the basic considerations of PCB complexity and the statistical evaluation of versatile designs.

Keywords—PCB complexity; Data mining

I. INTRODUCTION

Designing Electronic Control Unit (ECU) is always a very challenging task. It is especially challenging in automotive industry. The customers (for instance car manufacturers) are choosing such Original Equipment Manufacturers (OEM) they are able to provide the best quality and the cheapest product, which have to comply even with the strictest standards. OEMs also have to pay attention to their internal processes. Due to the higher customer expectations, the number of functions to be integrated into an ECU is increasing, while board size is shrinking. This results in more and more complex schematic circuits and denser printed board layouts. Because of the numerous requirements such as desired layout area, placement restrictions (high current or high frequency components), mechanic constraints, electromagnetic emission limits and immunity levels (commonly EMC) the routing becomes time consuming and very difficult. Unfortunately some of the rules are excluding the others making impossible the optimization of every parameter. [1] Usually a trade-off analysis is done to define which requirements are more important and which one is less, so the routing paths and topology can be accordingly prioritized. It must be investigated, whether the layout is feasible or it is possible to reduce PCB fabrication cost by

Gábor Horváth

Department of Measurement and Information Systems Budapest University of Technology and Economics Budapest, Hungary Email: horvath@mit.bme.hu

using less demanding technology. At extremely high volumes the development costs content in unit price -compared to the manufacturing costs- are very low. Every achievement for example wiring in 6 layers instead of 8 layers causes significant unit price reduction. Unit price must cover the profit and the following main costs:

- Schematic and Layout design
- Mechanic design and tooling
- Design validation and simulations
- PCB Manufacturing and Assembly and Testing

The following points represent different activities, most of them were earlier independent from the others, but it has changed principally. Today they are mutually influencing each other. The financial success of project depends on two factors: Firstly on the previously mentioned fabrication costs, secondly on the design costs. Although the design costs are relative low at high volumes compared to the fabrication costs, they must be kept as low as possible. This imposes the optimization at different design stages.

- Schematic designers are optimizing the cost and performance of different circuit solutions. This leads generally to cheaper circuit solution, which uses discrete components instead of integrated circuits. In turn the board will be more complex.
- Layout designers are trying to minimize net lengths and use as simple structures as possible and they must reduce the layout design time too (Internal costs).
- Mechanic designers must simplify the construction and reduce the amount of material needed (metal or plastic case). The mechanical stability may affect the routing channels.

II. RELATED WORK

The human factor is very important in routing automotive mixed signal circuits. Analytic approaches are also known. The two main groups are: qualitative and quantitative. [2], [3], [4], [5]. These groups can be further divided into subgroups. Qualitative approach says that the similarities between the new

project data and the existing experience in the past give the basics of prediction of layout design time, PCB cost or design complexity. Niazi et al. compare, classify and discuss the different models in [2]. It is common in all approaches that they create design metrics; try to find similarities between recent project data and historic designs. They give some details on regression analysis, which assumes linear relationship. Back-Propagation neural networks (BPNN) on the other hand can be used to overcome the disadvantages of regression analysis. These methods are analogical. Intuitive methods like expert systems, fuzzy systems are trying to store the knowledge of experts in form of rule bases. Quantitative methods like analytical cost estimation techniques are keeping in focus that both design process and manufacturing are made of different activities. Utilizing designer's experience along with computer aided technologies would be a powerful method in exploring the hidden difficulties and in decision making. Choosing the appropriate technology is not trivial and the following questions are to be answered:

- Which methods and PCB metrics are suitable?
- Is the PCB routable?
- If not, which constraints are to be changed?

III. DATA SOURCES AND PREPROCESSING PCB FILES

The goal is to extract knowledge from a set of PCBs. It is essential to find a PCB description format suitable to store electrical and geometric objects. There are several file types like Gerber (currently RS274X), ASCII, CCZ, Unidat, etc. Each of them has advantages and disadvantages as well. Gerber files contain the whole panel data, that necessary for the manufacturing, but there are no hints related to the netlist or to the design considerations. We need information in a broad range from tracks to design rules for extracting knowledge from the past projects. In Fig. 1 some basic shapes are shown. Since PCBs are composed of complex copper areas and tracks, these objects can be described with lines and arcs. Syntaxes are shown in simplified form:



To build a knowledgebase the individual PCB files must be analyzed. Robust software procedures ensure data integrity by checking the files if the most important data structures have been initialized. The following main points must be checked:

A. Empty or only Header files

The basic task is to filter these files. There is already some information in the file but the description is still incomplete. For example: units, layer stack up, panel outline are known, but there are no routing data or pick and place tags. This kind of files can be simply ignored.

B. Unions of poligons

In some cases the copper areas are split into many partially overlapping regions. It can be later confusing even if the overlapping rate is minimal, so a possible solution is to make the union of the regions.

C. Area of polygons and the influence of slots

Sometimes outlines are concave and contain holes and slots. They are constructed from custom sequences of polygons as seen in Fig. 2. Practically they can be represented with arrays of polygons. In such cases the polygons are described with the same data syntax regardless whether it is a hole or a copper area. We must therefore identify the particular areas, after that it will be possible to find the maximum area (A_{max}). Knowing that, the signs of the individual polygons in the array can be determined. Area of Slots and holes (Poly2 and Poly3 in Fig. 2) will be negative.



Fig. 2 Outline of a printed circuit board with slots

D. External Objects

Because of manufacturing reasons several additional objects like drills, patterns, codes and -in special caseselectrical components will be placed outside of outline. At this stage we must clip the whole PCB region with the outline.

E. Invalid Area, Excessive pad density

From technical point of view pad density must have an upper limit. As in Fig. 3 an example shows: a 0.5mm pitch BGA footprint has 324 bumps per cm². Suppose two side placement of components we get $2\times324=648$ PAD/cm² as limit. If the value exceeds that empirical limit, it is a sign that the unit system must be false. Depending on PCB dimensions (for instance 5mm×5mm board dimensions are not real.) the real unit must be either cm or inch.



IV. INITIAL CONSIDERATIONS

Practical way to determine the complexity is the placement study (see Fig. 4): placing all the components to the board to see if enough space remains for the routing.

VIA (e.g. interconnections between layers) density maps can be generated to see if there are any hotspots on the panel. In Fig. 4 at the right side the critical regions were marked with red. There are many other types of maps (Trace direction change density, wiring capacity). It would be interesting to know which patterns can be detected most frequently.



V. PRELIMINARY RESULTS

As first attempt we tried to find some general linear relationship between the number of pads (horizontal axis) and VIAs (vertical axis) (Fig. 5). Number of pads is definite at schematic design phase, while VIAs only at the end of the layout design.



Next we tried to find relationship between the number of pads and board layers (Fig. 6). Because of its poor information content Layers, PAD count, board area dependence was checked. (Fig. 7) The plot shows the situation of different PCB stack-ups. The meanings of colors blue, cyan, yellow and red are 2, 4, 6 and 8 layer PCBs.



Fig. 7 PAD-Area-Layers diagram

Previously several PCBs have been evaluated based on general statistical properties. To explore the structural composition of the schematic, the netlist must be considered and evaluated as a graph. Analyzing the schematic based on its electrical modules would be a powerful method to find further similarities between projects. A metric or semi metric is desired which takes into consideration the different component groups and the structural features of the circuit. Therefore, we calculated the node distribution and the number of components in the groups like resistor, capacitor etc. It was found that generally GND and VCC have the most nodes. Care must be taken to handle this, since the high number indicates just the possibility of supply net. Additional methods are required to validate it.

The Jaccard, Cosine, Euclidean metrics were also investigated. To test the performance of each metric four different configurations of project sets were created:

- Different versions of a product (product1)
- Different versions of another product (product2)
- A mixture of 150 different projects
- A mixture of 300 different projects

From statistical point of view we represent the projects with <key, value> pairs. The set of projects called P. Making the Cartesian product $PR \in \{P \times P\}$ it is possible to compare them. The Jaccard distance is the fraction of the intersection and union of keys but it does not take into account the values. P_k and P_1 are project k und project 1:

$$S_{J}(P_{k}, P_{l}) = \frac{|Keys(P_{k}) \cap Keys(P_{l})|}{|Keys(P_{k}) \cup Keys(P_{l})|}$$
(1)

Cosine metric takes the inner product of values in numerator. If a value is zero the numerator will be zero as well.

$$S_{C}(P_{k}, P_{l}) = \frac{\sum_{i \in Keys(P_{k}, P_{l})} Values(P_{k}^{i}) \times Values(P_{l}^{i})}{\|Values(P_{k})\| \|Values(P_{l})\|}$$
(2)

Jaccard and cosine metrics are inappropriate to show real similarities between projects because of their limitations. Euclidean distance takes into consideration not only the values of common keys but the values of disjoint keys too.

$$S_E(P_k, P_l) = 1 - \frac{\sqrt{\sum_{i \in Keys(P_k, P_l)i} \left(Values(P_k^i) - Values(P_l^i) \right)^2}}{d_{E_{max}}} \quad (3)$$

Some results are shown in **Fig. 8**. The results are arranged in a matrix and rows and columns represent the individual projects and the color in particular row and column the similarity between them. A linear gradient from blue to red means the values in a range of [0; 1]. Another interesting possibility would be to detect known electrical modules like CAN, LIN, DCDC converter, filter circuits.



Fig. 8 Similarity between projects a)Jaccard, b)Cosine, c)Euclidean

VI. SIMPLE MODELL TO ESTIMATE PARAMETERS

Talking about PCB complexity means the estimation of the board's routability at given constraints. Firstly we choose board area, number of pads, resistors, capacitors, ICs, transistors, diodes as inputs and number of layers and VIAs as outputs like in Fig. 9. We have data of approximately 8000 boards having different sizes and parameters. Recall that, the distribution of the projects (see Fig. 7) span over multiple decades in each parameter (area, PADs). Simple way could be to use multidimensional linear regression. Intuitive approach is to choose the following parameters as inputs: number of layers (shortly #Layers) and similar #resistors, #capacitors, #inductors, #diode, #PADS, board area. M is the number of model parameters.

$$y(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \, x_i \tag{4}$$

The number of possible training sets $is\binom{n}{k}$. In our case n=8000 and k is the number of points in the training set. The samples were taken randomly with uniform distribution from

the whole data set and no preliminary assumptions were made. Since different products generally composed of different electric modules, so it was seen during the visual data analysis that the variance of the component count is relative high. To get an insight for the distribution of weight k=300 points were taken and the corresponding weights were determined. This process was 800 times repeated and the distribution in Fig. 10 was calculated.



Fig. 9 Simple Estimation model

It was observed that two weights: area and number of pads were dominant in the model and were not affected by the size of training set. The other parameters depend on the particular project, therefore, their variances are higher.



VII. CONCLUSION

This paper was dealing with the aspects of the complexity analysis of printed circuit boards and gave a brief overview on PCB data processing. Some problems were detailed concerning polygons and dimension. There are different methods which are capable to estimate some aspect of a new design using design metrics. Several metrics were used to find similarities between netlists. We have shown some results at different metrics. PCB complexity is composed of diverse tasks, since the accuracy of the prediction is determined by the similarities of schematic, outline, requirements etc.

The topic was divided into three sections:

- Manufacturer's and printed board technology features must be identified
- Based on priorities and preliminary netlist information we tried to find similarities and predicting how complex the layout would be.
- By post processing the layout we would like to collect experience on what was good or wrong in the past.

The potential design risks can be estimated by trying to find similarities between knowledge and schematic. Artificial intelligence based methods are being tested to extract useful information.

REFERENCES

- [1] IPC-2226 "Design Standard for High Density Interconnect (HDI) Printed Boards" 10-2000, pp. 6–9.
- [2] A. Niazi, J. S. Dai, S. Balabani, L. Seneviratne "Product Cost Estimation: Technique Classification and Methodology Review" *Journal of Manufacturing Science and Engineering*, Vol. 128, May 2006, pp. 563–575
- [3] R. Kastner, E. Moshe, "Predictability for PCB Layout Density" http://www.ipcoutlook.org/mart/49710.shtml, Apr. 2012
- [4] Y. Kwon, O. A. Omitaomu, G. N. Wang, "Data mining approaches for modeling complex electronic circuit design activities" *Computers & Industrial Engineering*, Vol. 54, July. 2007, pp. 229–241
- [5] C. Bazeghi, J. Renau, "Printed Circuit Board Time Estimation" Workshop on Complexity-Effective Design, June. 2006, pp. 4–10

Examination of the input dependence of data flow graph clustering

Gábor Wacha Department of Measurement and Information Technology Budapest University of Technology and Economics Budapest, Hungary

Email: wacha@mit.bme.hu

Abstract—Control- and data flow analysis of a given software can give insights on the internals of the algorithm. With the generation of a data flow graph based on runtime measurements, it is possible to separate data independent parts of a given program. However, the result of the runtime measurements is dependent on the input of the software.

In this paper we introduce a method to estimate the impact of the input variables on the data flow graph, in particular on the different clusters of the graph. We test our approach with the JPEG image compression algorithm.

I. INTRODUCTION

With manycore architectures becoming more common, embedded application developers face a new problem – task distribution on the different processor cores.

Profile analysis can help to find an even load distribution on the different cores. Load balance aware task partitioning algorithms exist, for example in [1], however, the resulting system is likely to suffer performance drawbacks in cache usage: passing large amounts of data between the partitions assigned to different cores can result in more miss in the processor local cache.

The data flow analysis and spectral clustering method introduced in [2] can be used to find data independent partitions of the algorithm under test. From run-time measurements the amount of data passed between specific functions of the software is recorded into an Aggregated Dynamic Data Flow Graph (ADDFG) [3]. The ADDFG is a directed graph. The nodes of the ADDFG correspond to the software functions in the algorithm. Each edge has a weight which corresponds to the amount of data transferred between the functions during a given execution.

A spectral clustering method can show data-independent partitions, where communication between the functions residing in different partitions is kept as minimal as possible. With the explored partitions, a cache-aware task scheduling can be achieved.

However, the analysis relies on run-time measurements. One of the problems of run-time measurements is the input sensitivity of the result. The ADDFG – on which the clustering and the explored partitions are based – can be different on every execution of the algorithm.

We propose a method to examine the impact of the different input variables on the ADDFG, and particularly on the Béla Fehér

Department of Measurement and Information Technology Budapest University of Technology and Economics Budapest, Hungary Email: feher@mit.bme.hu

explored partitions. First, we introduce the assumptions we made on the software under measurement in Section II. Then we describe our analysis workflow in Sections IV, V and VI. Finally, we demonstrate our methods in Section VII.

II. GENERALIZED SOFTWARE MODEL

In this paper, we regard the software as a nonlinear, deterministic function, which takes input parameters, and returns a feature as output. This is different than the "classical" software, which takes input parameters, and returns a calculated output value, since a feature is a more abstract object: it can also be a call graph, a coverage result, or a given hardware state of an embedded system.

The software will be regarded as a gray box: every internal state and executed instruction is observable, but the activity (e.g. adding two values) of a specific processor instruction is unimportant.

A. Software parameters

Most algorithms have input parameters which influence the execution. We classify these input parameters into the following groups:

- *Main dataflow*: the input parameter is part of the data on which the software does its main calculation. For example, raw image data in the JPEG compression algorithm is part of the main dataflow.
- Data length descriptor: the input parameter describes the length of the main dataflow. This parameter is often not given explicitly, it is inferred from the size of the main dataflow. For example, the data length descriptor parameter is the image size in the JPEG compression algorithm.
- *Behavior selector*: the input parameter selects from different sub-algorithms. Different algorithms result in significantly different behavior. The different behavior typically manifests in different program states or different call graphs. For example, in a JPEG handler software, the input which selects between image compression or decompression is a behavior selector parameter.

We assume that the value of every input parameter is available at the start of the execution of the given software, however, in many embedded software the timing of an actual input (e.g. the time when an interrupt arrives) can influence the output.

To model the time course, we can regard these inputs as value-time pairs. With this, timing information is included in the software parameters. The concrete analysis of timed input parameters is out of the scope of this paper.

B. Software output

As stated in Section II, we regard the output of the algorithm as a feature. In usual software profiling cases the feature can be an execution time, a call graph or a memory footprint. In our paper the output of the software is the ADDFG of the given program execution. We view the ADDFG as a graph adjacency matrix, with each element of the matrix is one output.

III. OUTLINE OF THE ANALYSIS

Our method to analyze the impact of the input variables on the ADDFG can be summarized in the following steps:

- 1) *Input generation.* During this step we prepare the list of possible inputs for the algorithm under test. The main difficulty is to find a finite set of input combinations, which reflects the infinite set of possible inputs. We show how to handle the different kind of inputs introduced in Section II-A.
- 2) Algorithm execution and ADDFG generation. We execute and observe the algorithm with the inputs generated in Step 1. to create the output. The execution results in a set of ADDFGs. The main difficulty is to monitor the memory and register access of every executed instruction, which is needed for the ADDFG generation.
- 3) *Error estimation.* From the set of ADDFGs generated in Step 2., we give an estimation of the possible error of the clusterization method. The main difficulty is to find an estimation, which can be used with the ADDFG, and which also gives practical results.

IV. INPUT GENERATION FOR THE ALGORITHM UNDER TEST

Since we do not know a direct relationship between the input parameters and the ADDFG, to examine the impact of each input variable, we have to generate the ADDFGs for different inputs.

If the domain of all input parameters is a finite set, for every combination the ADDFG can be generated. However, most algorithms do not have this property. The solution can be to generate the ADDFG to a finite set of input combinations, while taking care that input data for the measurements reflects the input expected in the real world application.

In the following sections we assume that every input is already classified into the groups introduced in Section II-A. This assumption is valid, since a real world algorithm is a gray box: we have *a priori* knowledge what each input parameter does.

The main problem is that for each different kind of input the ADDFG can be different. This means that different kinds of input parameters should be regarded differently. To simplify the input dependence analysis, we have the following assumptions:

- Behavior selector variables have finite number of combinations in real world applications. This means we can generate a set of ADDFGs for each behavior selector variable combination, thus partitioning the original problem into independent subproblems.
- The impact of the data length descriptor variables can be examined as follows. The control and data flows of a real world algorithms usually have a relationship with the data length describable in a closed form expression. This allows us to examine only a finite set of input lengths and deduce the input dependency over the data length from the results.

With these assumptions, only the impact of the main dataflow has to be examined, we can constrain the data length and the selected behavior to fixed values.

For the main dataflow input variables we need *a priori* knowledge on the use case of the software. We have to generate inputs with a fixed data length, which reflects real-usage input data.

To clarify, for example in a JPEG handling software we constrain the behavior as compression, and we generate different, but fixed size input images, where the image is from real world data.

V. ALGORITHM EXECUTION

To analyze the impact of the different input parameters on the ADDFG, the software should be run in an observeable, reproducible environment. To generate the ADDFG for each input generated according to Section IV, the memory or register access of each executed CPU instruction has to be traced.

If we trace only the instruction counter, the address of the accessed memory and the direction of the data transfer for each executed processor instruction, we can estimate the necessary bandwidth of the trace data.

In a standard 3-operand instruction set (like in the case of the ARM), one instruction can have up to 3 data accesses. Calculating with 32 bit memory addresses, a 32 bit instruction counter, and a processor running on a relatively low clock frequency of 100 MHz, the necessary bandwith is in the magnitude of at least 1 GByte/s.

This bandwidth is not available on trace hardware, in consequence, for tracing a processor simulator is used. The simulator executes each instruction, and stores the trace information. Because the effect of every executed instruction on the registers or on the memory is traced, modern techniques – such as binary translation – can not be used effectively.

The trace information for one execution – if we estimate the simulation time with 1 real world second – stores approximately a hundred million records. This data is inserted in a database, the trace log of each executed instruction is stored. Insertion of each instruction separately has strong impact on the performance of the simulation, since for every executed instruction, a database query should be generated, sent to the database server and parsed. Prepared statements can achieve better performance, however the best performance is achieved with bulk loading the trace data into the database.

The following table shows the results of our measurements of the average simulation speed in MIPS using the PostgreSQL database server and OVP Microblaze simulator:

Insertion strategy	Simulated MIPS			
Query	0.2			
Prepared statement	0.3			
Bulk insert	0.5			
Without database insertion	0.6			

According to the results, the simulation is two magnitudes slower than the real hardware with 100 MIPS. However, the simulator performance is enough to observe the execution of the embedded system and generate the ADDFG.

VI. ERROR ESTIMATION OF THE SPECTRAL CLUSTERING METHOD

A. Finding the clusters of the ADDFG

The clustering algorithm is the following [4]. Let G be the weighted adjacency matrix of the directed ADDFG. The clustering algorithm is only usable on symmetric matrices, the undirected ADDFG should be defined. We assume that on each data transfer between functions the performance loss is the same regardless of the direction of the transfer. With this assumption, we can define the undirected ADDFG, where the direction of the edges can be ignored. The adjacency matrix A of the undirected ADDFG is calculated from G as stated in Eq. 1.

$$A = (G + G^T) - diag(G).$$
(1)

A is a symmetric matrix, on which the spectral clustering method [4] can be used. Let L be the unnormalized Laplacian matrix of A [5]:

$$l_{ij} = \begin{cases} deg(i) & \text{if } i = j \\ w(i,j) & \text{if } i \neq j, \\ 0 & \text{otherwise,} \end{cases}$$
(2)

where deg(i) is the degree of the *i*-th vertex, and w(i, j) is the weight of the edge between the *i*-th and *j*-th vertex. When there is no edge between *i* and *j*, w(i, j) is 0.

The unnormalized Laplacian matrix can be calculated as D-A, where D is the diagonal degree matrix of the weighted symmetric adjacency matrix A:

$$D_{ij} = \begin{cases} deg(i) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$
(3)

The eigenvectors of L hold information about the clusters of A. All coordinates of the eigenvector corresponding to the smallest eigenvalue of L are the same, the eigenvalue is 0, resulting that the Laplacian matrix is singular. All the other eigenvalues are real [5].



Fig. 1. Plot of the eigenvector coordinates corresponding the second smallest eigenvalue of L in a demonstration algorithm

The coordinates of the eigenvector of the second smallest eigenvalue can be used to find the clusters of A. Comparing the coordinates (which are real numbers, so a comparison can be done) of the eigenvector to a given threshold value can give two clusters of the graph. The *i*-th vertex of the ADDFG is in the first cluster if the *i*-th coordinate of the eigenvector is less than the threshold value, in the second cluster otherwise. 0 is often the most straightforward choice for two clusters [6].

Fig. 1. shows the plot of the coordinates of the eigenvector generated from the ADDFG of a demonstration algorithm. On the x axis the function name corresponding to the given coordinate is given. We can separate two main clusters if we select the threshold value as 0.

B. Error estimation

For every execution, the ADDFG – and consequently the Laplacian matrix L – can be different depending on the input. If we use the partitions explored resulting from a given execution for software parallelization, it is possible that for other inputs the difference in the ADDFG results in different clusters. Very different clusters will result in suboptimal parallel execution and cache usage.

The partitioning algorithm introduced in Section VI-A finds the clusters based on the eigenvectors of the graph Laplacian of the ADDFG. The problem of error estimation can be reworded as estimation of the sensitivity of the eigenvectors regarding perturbations on the Laplacian.

The perturbation theory of eigenvalues and eigenvectors is a remarkably active area of research. Many methods exist, see in [7], [8], [9] and [10]. However, because of the properties of the Laplacian matrix defined in Eq. 2, most approaches give unusable results: the Laplacian is singular, not positive definite, and its eigenvalues are not unique.

L. Huang et al. in [9] show a practically usable estimation on the clustering error. They define η , the *mis-clustering rate* as the proportion of samples that have different cluster memberships after data perturbation.

The first step of their method is to approximate the misclustering rate in case of eigenvector perturbation. Let L the Laplacian of the ADDFG, \tilde{L} a perturbed Laplacian, dL =



Fig. 2. Histogram of the edge weight between *encode_one_block* and *emit_bits_s* in the JPEG compression algorithm

L-L. Let v_2 and \tilde{v}_2 be the eigenvectors corresponding to the second smallest eigenvalue of L and \tilde{L} respectively.

An estimation of η is given as:

$$\eta \le \|\widetilde{v}_2 - v_2\|^2 \tag{4}$$

To calculate η , an estimation to $\|\widetilde{v}_2 - v_2\|$ is necessary. According to [11], the norm of the difference of v_2 and \widetilde{v}_2 can be estimated as

$$\|\widetilde{v}_2 - v_2\| \le \frac{\|4dL\|_F}{\nu - \sqrt{2}\|dL\|_F},\tag{5}$$

where ν is the difference between the second and the third eigenvalues of L.

The Frobenius-norm $||dL||_F$ is calculated as the square root of the sum of the absolute squares of the elements of dL. With uncertainity evaluation, the worst case value of $||dL||_F$ can be calculated from the measured ADDFGs.

VII. RESULTS

We tested the approach on the libJPEG compression. With different, 32x32 sized inputs, we generated the ADDFG of each execution.

The generated ADDFGs had the same edge weigths between the nodes, except *encode_one_block* and *emit_bits_s*. This is not unexpected, since the JPEG compression algorithm works on fixed size blocks. The only part of the algorithm which has different input sizes is the Huffman coding, since the Huffman coding is a variable length coding.

Fig. 2. shows the histogram of the edge weight of 100 executions. The edge weights of the ADDFG were normalized so that the trace of the ADDFG is 1.

As we stated in Section II, the software is an unknown nonlinear function of the input parameters resulting in the ADDFG. This means, that the distribution of the edge weight is unknown. Normality tests show that the edge weight can not be modeled as a normal distribution. To give a confidence interval on the average edge weigth, Chebyshev's inequality can be used. We used a 90% confidence interval to examine the mis-clustering rate. With the worst case perturbation (where the Frobenius norm of the error of the Laplacian matrix was pessimal), the misclustering rate was calculated as 0.01. This means only 1% of the 256 functions in the JPEG algorithm can be in the wrong cluster in case of an unfortunate input.

VIII. CONCLUSION

In this paper, we introduced a method to estimate the error of the clusterization of an ADDFG of a given algorithm. We described how the different kinds of input parameters can influence the generated ADDFG. A method was given to enable the fast simulation and observation of the algorithm under test. We estimated the mis-clustering rate with methods described in [9]. We tested our approach with the JPEG compression algorithm.

In future research, the impact of the timed input parameters can be examined. Also with input dependency analysis it could be possible to enable better profiling, directed code coverage measurement and algorithm optimization.

REFERENCES

- J. Kang and D. Waddington, "Load balancing aware real-time task partitioning in multicore systems," in *Embedded and Real-Time Comput*ing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on, Aug 2012, pp. 404–407.
- [2] G. Wacha and B. Fehér, "Examination of algorithms using dynamic data flow graphs," in *Proceedings of the 21st PhD Minisymposium*, 2014, pp. 44–47.
- [3] I. Szabó, G. Wacha, and J. Lazányi, "Aggregated dynamic dataflow graph generation and visualization," *Carpathian Journal of Electronic* and Computer Engineering, vol. 6, no. 2, pp. 50–54, 2013.
- and Computer Engineering, vol. 6, no. 2, pp. 50–54, 2013.
 [4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. MIT Press, 2001, pp. 849–856.
- [5] B. Mohar, "The Laplacian spectrum of graphs," in *Graph Theory*, *Combinatorics, and Applications.* Wiley, 1991, pp. 871–898.
- [6] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11222-007-9033-z
- [7] W. L. Xiao Shan Chen and W. W. Xu, "Perturbation analysis of the eigenvector matrix and singular vector matrices," *Taiwanese Journal of Mathematics*, vol. 16, pp. 179–194, 2012. [Online]. Available: http://journal.taiwanmathsoc.org.tw/index.php/TJM/ article/viewFile/1588/1263
- [8] R. Mathias, "Spectral perturbation bounds for positive definite matrices," SIAM J. Matrix Anal. Appl, vol. 18, pp. 959–80, 1997.
- [9] L. Huang, D. Yan, N. Taft, and M. I. Jordan, "Spectral clustering with perturbed data," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 705–712. [Online]. Available: http: //papers.nips.cc/paper/3480-spectral-clustering-with-perturbed-data.pdf
- [10] F. M. Dopico, J. Moro, and J. M. Molera, "Weyl-type relative perturbation bounds for eigensystems of Hermitian matrices," 2000.
- [11] G. W. Stewart and J. Guang Sun, *Matrix Perturbation Theory*. Academic Press, 1990.

Design and Implementation of On-line System Supervision for Beam Loss Monitoring Systems

Csaba F. Hajdu*[†], Tamás Dabóczi[†], Christos Zamantzas*

*European Organization for Nuclear Research (CERN), Geneva, Switzerland. Email: {cshajdu,czam}@cern.ch [†]Budapest University of Technology and Economics, Department of Measurement and Information Systems, Budapest, Hungary. Email: {chajdu,daboczi}@mit.bme.hu

Abstract—The strategy for beam setup and machine protection of the particle accelerators at the European Organisation for Nuclear Research (CERN) is predominantly based on beam loss monitoring (BLM) systems. These systems were designed to protect the machines against unintended energy deposition by particles lost from the beams and escaping the beam lines, to assist in diagnosing machine faults and to provide feedback of the machine behavior to the control room and other systems.

The aim of the PhD thesis project related to this paper is to design and implement a process providing a continuous and comprehensive surveillance of the entire beam loss monitoring signal chain from the detector to the processing electronics, with particular emphasis on the connectivity and functionality of the detectors. At present, no particle accelerator features such a procedure.

Sections I-III of this paper elaborate on the background of the project. Section IV summarizes the motivation of the project, briefly reviews the current best implementation and discusses the state of the development. In particular, Section IV-A surveys the hardware features of the system relevant to the development and Sections IV-B and IV-C relate the measurements I conducted to assess the transfer characteristics of the signal chain with the modulation of the high voltage power supply as input. Section V anticipates the next steps of my development work.

I. INTRODUCTION

The particle accelerator complex dedicated to fundamental research in physics at the European Organisation for Nuclear Research (CERN) consists of several distinct accelerators [1]. Specifically, particles are accelerated to increasingly higher energies through a succession of these before being injected into the current flagship particle collider and highest energy machine of the institute, the Large Hadron Collider (LHC). This sequence of accelerators is referred to as the LHC injector complex.

The LHC Injectors Upgrade (LIU) project has been launched to consolidate the aging low energy machines and to meet the ever more stringent requirements in terms of particle beam quality imposed by the LHC. Particle beams with higher energy and intensity¹ will allow to further increase the luminosity² of the LHC.

At CERN, beam loss monitoring (BLM) systems deployed at the accelerators are at the core of the strategy for beam setup and machine protection. A continuous supervision of the entire BLM signal chain is therefore essential.

¹The number of particles in the beam or, equivalently, the beam current. ²The rate of collisions, more precisely, the ratio of the number of events detected in a certain time to the interaction cross-section.

II. BEAM LOSS MONITORING

Beam loss monitoring consists of measuring the shower particles originating from particles lost from the beams and escaping the beam lines. If necessary, the safe extraction of beams and an inhibition of injections is triggered in order to protect the machines against unintended energy deposition, and measurement data are provided for machine calibration and tuning. In particular, for a machine based on superconducting magnets such as the LHC, the fast response of the BLM system is critical for protection against short and intense particle losses, while for longer losses, the quench protection system (QPS) and the cryogenic system provide assistance [2].



Figure 1. Photograph of an LHC ionization chamber with the cylindrical insulating cover removed [2]. Notice the stack of aluminum electrodes and the ceramic insulation at both ends.

The beam loss monitoring system in operation at the LHC employs mainly ionization chambers, depicted in Fig. 1, as detectors. The particles crossing the chamber create ionization, that is, they liberate electrons from the molecules of the gas filling the volume. The resulting electrons and ions are then separated by a bias high voltage applied to the terminals of the chamber and collected on a stack of conductive electrodes, thereby creating a measurable charge proportional to the total energy deposited in the chamber by the ionizing radiation. The most advantageous locations for the placement of the discrete detectors are determined in advance by simulation.

The output signal of the detectors is connected to the front-end electronics via copper cables for measurement and digitization. The numerical values are then sent to the back-end processing and decision making card over a high-speed serial link. The back-end card computes several moving window integrals of different durations, referred to as running sums (RS), for each channel. These are then compared to their respective predefined abort thresholds in real-time and in case of excess, a beam abort and injection inhibition is triggered.

In order to provide the necessary fail-safety and achieve the expected availability, a large number of additional processes



Figure 2. Architecture of the BLM acquisition system for the injector complex at CERN [3].

monitoring the status of the system in real-time have been implemented [4].

The LIU project mandates the development and commissioning of a new BLM system for the injectors. The new system features a higher acquisition frequency and extended dynamic range, and is adapted for the connection of more varied types of detectors as input than its predecessors [3].

III. THE BLM SYSTEM FOR THE INJECTOR COMPLEX

The architecture of the new BLM acquisition system, shown in Fig. 2, has been chosen to make it generic, versatile and highly performant, also taking into consideration the high expectations in terms of reliability and availability. The system makes use of reprogrammable FPGA devices for flexibility and to allow targeting the various requirements of the different injectors. Since the electronics will be installed in protected areas, no radiation tolerance is required.

The acquisition stage has been designed to allow connecting several detector types. In the majority of cases, the use of ionization chambers similar to those employed for the LHC is foreseen, however, other detector types such as secondary emission monitors, diamonds and Cherenkov detectors will also be used in some locations to cover particular cases [3].

The system is in an advanced stage of development. A prototype installation has been set up at the Proton Synchrotron Booster (PSB) accelerator with pre-series front-end cards, in parallel to the previous system currently used for machine protection. The prototype system is used to evaluate performance and gather operational experience.

A. Front-end electronics

An acquisition crate (BLEAC) hosts a maximum of eight acquisition modules along with a control unit (BLECU).

The output current of the detectors is digitized with a new mixed measurement technique based on a fully differential integrator [5], implemented on the new BLEDP acquisition module. It allows the acquisition of input currents ranging from 10 pA to 200 mA, which corresponds to a dynamic range of $2 \cdot 10^{10}$. The measurement range is split into two overlapping ranges covered by two separate circuits: the Fully Differential Frequency Converter (FDFC) operates in the range 10 pA - 10 mA, and the Direct ADC (DADC) in the range $100 \mu \text{A} - 200 \text{ mA}$. The front-end automatically switches between the two modes depending on the magnitude of the input

current. The switching thresholds are set by the FPGA of the acquisition module, which is also in charge of collecting and transmitting the digitized data for further processing [6].

Each BLEDP card provides eight input channels. A small, constant offset current is injected into every channel in parallel to the input signal in order to stabilize the circuit. The use of this mechanism to detect whether the channel in question is operational is also foreseen in the future.

B. Back-end electronics

Up to eight processing modules, a Linux-based front-end computer (FEC), accelerator timing receiver cards and a Combiner and Survey module (BLECS) are hosted in a VME64x processing crate.

The processing and triggering (BLEPT) modules feature an active mezzanine card hosting an FPGA. The FPGA is responsible for communicating with the BLEDP cards over an optical link, receiving the acquired data, computing the running sums and comparing them to their respective beam abort thresholds. The information is transmitted to the FPGA on the carrier board, which handles communication with the FEC and the beam interlock lines. This involves publishing processed data and status information on the VME bus, and directing the decision regarding beam injection and circulation.

The BLECS module distributes accelerator timing and status signals to the BLEPT cards and forwards the beam abort requests generated by the BLEPT modules to the Beam Interlock System. It is also responsible for initiating the system sanity check procedures and providing confirmation of their successful execution at regular intervals [3].

IV. CONNECTIVITY CHECKS

At present, no particle accelerator features a procedure providing a continuous and comprehensive supervision of the functionality of its beam loss monitoring system, with particular emphasis on the connectivity and functionality of the detectors. The current best solution operates in the LHC, where among the sanity check procedures mentioned earlier, a connectivity check of each detector channel is enforced every 24 hours. This check can only be executed while the accelerator is not operational.

The primary purpose of this test is to ensure the correct cabling connection of each beam loss monitor. By adding a

harmonic modulation to the high voltage supply of the detectors, a corresponding modulation is induced in their output current, which can be detected with the normal signal acquisition chain. If the cable to a detector is missing, disconnected or discontinued for any reason, the harmonic modulation won't be present in the output current. The modulation frequency used in the LHC is on the order of 10 mHz.

This procedure also allows surveying the integrity of the components. Variations in the amplitude and phase of the output current have been found to correspond to various kinds of deterioration in the components of the acquisition chain in the LHC implementation [7].

The present PhD thesis work involves the design, construction, testing and integration of an improved procedure in the injector BLM system. The process should provide a continuous supervision of the entire signal chain from the detectors to the processing electronics as well as the measurement ability of the detectors. However, since the BLM system is mission critical, the processing of the measurement signal should not be altered for the purposes of this test, to avoid introducing new failure cases into the system. Therefore, the error induced by the connectivity checks must either remain within the error margin of the system or the measurement must be done during periods with no beam in the machine, in order to avoid compromising machine protection.

A. Hardware features of the injector BLMs

In the injector BLM system, a maximum of 64 ionization chambers are powered in parallel by a pair of high voltage power supplies through a custom-made high voltage distribution box. The two power supplies are connected in parallel through protection diodes, with the secondary set to a voltage about 50 V lower. This way, in normal operation, the secondary is idle, but if the primary fails, it can take over powering the ionization chambers.

As in the case of the LHC system, each ionization chamber has a low-pass filter on its input, which can temporarily take over supplying charges to the chamber in case of high losses and the resulting high current draw. The cutoff frequency of these filters is approximately 0.03 Hz.

The power supplies are from the Heinzinger NCE series, capable of supplying an adjustable DC output voltage of 0-3000 V with a current limited to 20 mA. The output voltage is specified to be proportional to the voltage of 0-10 V applied to the analog setpoint input. The manufacturer provides no additional details about the implementation of the power supply.

The voltage setpoint input of the power supplies is driven by the sum of the voltage output of two 16-bit DAC channels, one for the DC component, the other one for the modulation as required. Their output is subject to analog low-pass filtering, with a cutoff frequency of about 80 Hz. The voltage is capped at 6.8 V with a voltage regulator diode to avoid raising the bias voltage above the high end of the ionization chamber detection region, about 2000 V. In the current configuration, the highest possible modulation amplitude is approximately $250 V_{PP}$.

B. Measurement of the signal chain



Figure 3. Simplified view of the modulation signal chain.

In order to assess the capabilities of the modulation signal chain, described in Sec. IV-A and illustrated in Fig. 3, a series of measurements was executed.

Measured with an oscilloscope, the step response of the high voltage drive circuitry on the combiner card (BLECS) indeed corresponds to that of a first-order low-pass filter with a cutoff frequency of approximately 80 Hz.

Two measurement methods have been used to characterize the response of the high voltage power supply. First, the device features two analog feedback lines with voltages proportional to the output voltage and the output current, respectively. These are connected to an ADC and can be read by the FPGA of the combiner card. Since the operational firmware only acquires and publishes data at 1 Hz, a signal tap has been implemented in the firmware to access acquisitions at their native frequency of about 530 Hz. Second, a high voltage divider with a ratio of 1 : 1000 can be connected directly to the output of the power supply or to the high voltage distribution box, and its output can be measured with an oscilloscope. The results obtained with the two acquisition methods have proven to be consistent and have been used interchangeably.



Figure 4. Output spectrum of the high voltage power supply when driven with a 10 Hz modulation of approximately $250 V_{PP}$ added to a DC level of 1500 V. The waveforms were acquired from the feedback output of the supply. Notice the apparent nonlinear distortion.

With sinusoidal inputs of maximum amplitude and at frequencies up to 5 Hz, a clear sine wave is observed at the output of the high voltage power supply. However, at higher frequencies, distortions become apparent even in the time domain, and the spectrum becomes similar to that shown in Fig. 4, indicating nonlinear effects. Measurements with a square wave excitation suggested a slew rate limitation for voltage decrease, however, the analysis of the step response of the circuit revealed a different behavior. For increases in voltage, the high voltage power supply behaves as a firstorder low-pass filter, and for decreases, it exhibits a slow exponential decay with a time constant largely dependent on the loading impedance, typically on the order of 1 s. This behavior is consistent with the power supply actively driving output voltage increases and just letting its output capacitors discharge over the load and an internal parallel resistor for voltage drops. If this limitation is observed, the power supply behaves like a first-order low-pass filter with a cutoff frequency of about 30 Hz.

The high voltage distribution box contains no capacitors and appears to have no significant influence on the frequency response of the system.

C. Data acquisition

A rack hosting an acquisition crate and a processing crate (see Sec. III) along with a single high voltage power supply and a high voltage distribution box is available in the lab. A single ionization chamber was connected to the power supply through the distribution box and its current output was measured through the BLEDP card. The raw data acquired at 500 kHz were transmitted directly to a PC over Ethernet for processing [8]. For this measurement, the amplitude of the modulation needed to be reduced to ¹/₈ of the maximum in order to avoid exceeding the magnitude of the offset current, thereby clipping and distorting the signal, since the system can only acquire positive current.

The spectrum of the acquisitions was very flat, with peaks scattered throughout the spectrum due to the operation of the high voltage power supply. At modulation frequencies up to 20 Hz, the corresponding peaks were clearly detectable in the spectrum, at least 20 dB above the noise floor.

Similar acquisitions were made with the prototype system installed in the PSB. This prototype rack features both power supplies, driving a single high voltage distribution box. 32 ionization chambers, mounted onto the accelerator and acquiring actual loss signals whenever there is beam, are connected to the system. Due to the attenuation of the high voltage cables of 60 - 80 meters and the presence of multiple detectors, the modulation amplitude was set to the maximum in this case to obtain current amplitudes similar to those in the previous measurement, albeit with more noise.



Figure 5. Spectrum of the data acquired in the PSB with and without beam. The frequency axis spans 0-500 kHz. Notice the scattered peaks due to the operation of the power supply.

The spectra obtained with no beam are remarkably similar to those seen in the lab. However, as shown in Fig. 5, the presence of beam has a considerable effect on the spectrum. Along with an increase in the noise floor, an amplitude roll-off type effect can be observed at frequencies up to 100-150 kHz. The shape of the roll-off appears to vary from acquisition to acquisition. The low end of the spectrum contains a 0.833 Hz component along with its harmonics, which corresponds to the 1.2 s duration of the basic period³ of the PSB. Thus, such modulation frequencies should be chosen that the resulting peaks fall between the ones corresponding to the basic period. In this case, even though the magnitude of the peaks created by the modulation doesn't far exceed that of the basic period peaks, the modulation remains detectable up to about 15 Hz. Contrary to previous expectations, without beam, a 20.5 Hz peak is detectable with a margin of about 20 dB. It has to be noted that this signal is no longer detectable in the presence of beam.

V. PERSPECTIVES

Implementation of a prototype algorithm to acquire and identify the modulation is foreseen as the next step. The measurements suggest remarkable performance potential and exploiting a frequency range unused in the LHC implementation seems conceivable. However, the frequency response of the system will need to be studied further in order to implement an integrity survey similar to the LHC system.

Improvements to the simulation model of the signal chain, currently giving good estimates of the discharge time constants of the power supply, are foreseen to feature a more realistic series resistance in the power supply and better predict other parameters such as current consumption.

REFERENCES

- [1] The accelerator complex | CERN. http://home.web.cern.ch/about/accelerators.
- [2] B Dehning, E Effinger, J Emery, G Ferioli, G Guaglio, E B Holzer, Daniel Kramer, L Ponce, V Prieto, M Stockner, and C Zamantzas. The LHC Beam Loss Measurement System. (LHC-PROJECT-Report-1025. CERN-LHC-PROJECT-Report-1025):4 p, 2007.
- [3] C Zamantzas, M Alsdorf, B Dehning, S Jackson, M Kwiatkowski, and W Viganò. System Architecture for measuring and monitoring Beam Losses in the Injector Complex at CERN. Technical Report CERN-ACC-2013-0252, CERN, Geneva, Aug 2012.
- [4] C Zamantzas, B Dehning, E Effinger, J Emery, and S Jackson. Real-Time System Supervision for the LHC Beam Loss Monitoring System at CERN. (CERN-ACC-2013-0255):4 p, Feb 2013.
- [5] W Viganò, B Dehning, E Effinger, G G Venturini, and C Zamantzas. Comparison of three different concepts of high dynamic range and dependability optimised current measurement digitisers for beam loss systems. Technical Report CERN-ATS-2012-279, CERN, Geneva, Oct 2012.
- [6] W Viganò, M Alsdorf, B Dehning, M Kwiatkowski, G G Venturini, and C Zamantzas. 10 Orders of Magnitude Current Measurement Digitisers for the CERN Beam Loss Systems. J. Instrum., 9(CERN-ACC-2014-0001):C02011. 10 p, Sep 2013.
- [7] J Emery, B Dehning, E Effinger, G Ferioli, C Zamantzas, H Ikeda, and E Verhagen. LHC BLM Single Channel Connectivity Test using the Standard Installation. Technical Report CERN-BE-2009-027, CERN, Geneva, May 2009.
- [8] M Kwiatkowski, M Alsdorf, B Dehning, W Vigano, and C Zamantzas. A Gigabit Ethernet link for an FPGA based Beam Loss Measurement System. (CERN-ACC-2013-0296):4 p, Sep 2013.

³One basic period corresponds to an entire beam cycle: injection, acceleration, ejection. This periodicity is therefore characteristic of the entire system.