ISSN 2186-7437

NII Shonan Meeting Report

No. 2016-16

Implicit and Explicit Semantics Integration in Proof Based Developments of Discrete Systems.

Yamine AIT-AMEUR
Shin NAKAJIMA
Dominique MERY

November 22–25, 2016

湘南会議
SHONAN MEETING

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

NII Shonan Meeting Report

Organizers:
Yamine AIT AMEUR (INPT-ENSEIHT/IRIT, Toulouse, France)
Shin NAKAJIMA (National Institute of Informatics, Tokyo, Japan)
Dominique MÉRY (LORIA, Université de Lorraine, Nancy, France)

Novemebr 22–25, 2016

Overview of the meeting

The Shonan meeting entitled ”Implicit and explicit semantics integration in proof based developments of discrete systems” was organised on Nov. 22- Nov. 25 2016. More than 30 participants attended the meeting. The participants belong to both industry and academia coming from different parts of the world.

Objectives

The objective of the meeting was to discuss mechanisms for reducing model heterogeneity induced by the absence of explicit semantics expression in the formal techniques used to specify these models. More precisely, the meeting highlighted the advances in handling both implicit and explicit semantics in formal system developments. The following topics were addressed during the presentations and discussions.

- Making explicit the domain knowledge in formal models in order to handle *hidden* relevant properties (because they are not explicitly modelled in classical formal modelling languages)
- Defining different knowledge models to handle domain knowledge: ontologies have been identified as a candidate model.
- Identifying the mechanisms allowing system developers to refer to domain knowledge models
- Studying composition mechanisms to handle domain knowledge in formal modelling techniques
- Discussing reasoning mechanisms in presence/absence of explicit domain knowledge in system design models
- Studying heterogeneous model alignment to reduce semantic mismatch
- Identifying several applications where making explicit domain knowledge is relevant like assurance cases, security, requirements engineering, e-voting systems, etc.

Organisation of the meeting

The organisation of the meeting followed the steps described below.

- An opening talk was given by the meeting organisers in the first session of the first day
- Each participant gave a 5-minutes talk in the first session of the first day in order to introduce her/himself and provide a quick overview of her/his talk
- Participants made presentations on the above described topics. Sessions of three presentations were defined. A break was set up between two sessions.
- A debriefing session was set up after the end of each day. Three subgroups were defined and a subgroup animator was identified. This debriefing session consists in two steps 1) first, free discussions among the subgroup on the relevant seminar topics and 2) second a summary of the discussions of each subgroup in a plenary session. Three such sessions have been organised for 40 min each.
- A concluding session was chaired by the three meeting organisers at the end of the meeting.

Conclusions of the meeting

The meeting participants mentioned unanimously the interest of the seminar. The interaction between the participants led to interesting results among which we can cite

- The interest of bridging the gap between formal system models and knowledge domain models and the associated reasoning mechanisms
- Setting up a workshop on the topics addressed in the seminar in one of the major conference related to formal methods
- Publishing a book containing extended descriptions of the contributions of the participants to the seminar topics. The participants committed to submit their chapters. First contacts with the Shonan editor are already set up.

Overview of Talks

Introduction to the seminar and organisation issues

Yamine Ait Ameur (INPT-ENSEIHT/IRIT, Toulouse, France, Shin Nakajima (National Institute of Informatics NII, Tokyo, Japan, Dominique Méry (LORIA, Université de Lorraine, Nancy, France)

The organisers introduced the seminar, its main theme. They described the overall organisation and schedule of the seminar.

5 Minutes presentation for intruduction

All speakers

Each participant has presented the main topics of their talk. They have used a single slide shared among all the participants.

On Modelling Integration

J. R. Abrial, Consultant, Marseille, France.

I must admit that I had some difficulties understanding the exact meaning of the title of the seminar: "Implicit or explicit semantics integration in developments of proof-based discrete systems". I eventually reinterpret this title to: "Implicit or explicit modelling integration in developments of proof-based discrete systems". I choose this new title because I felt more comfortable with modelling integration than with semantics integration. But in preparing my presentation to the seminar, I figured out that modelling integration is certainly a well known concept in the development of programs [1][2] and more generally in that of complex systems [3][4]. Would it then be adequate to cover once again such a common idea? Nevertheless, under pressures of the seminar organisers, I decided to give a short presentation. In preparing further my presentation, I came across some recent facts showing that my previous assertion concerning the integration of modelling in developments was not at all, even these days, a common practice. These facts are the following: (1) the failure of the Schiaparelli rover landing attempt on the surface of the planet Mars, (2) the absence of any specifications in the university admission program at the end of secondary school for French students resulting in distributing to them a listing of the final program only for analysis, and (3) a computer science undergraduate curriculum. In all three examples there are no references to complex system modelling, refinement and proofs. My proposal for the seminar was then to study a precise definition of modelling without forgetting the important notion of initial requirements. Finally, I also recommended that we had some chat concerning the education of such matters to computer science students and engineers.

References

[1] R. Floyd *"Assigning Meanings to Programs."* Mathematical Aspects of Computer Science Proceedings of Symposium on Applied Mathematics. 19. American Mathematical Society, 1967

[2] N. Wirth *"Program Development by stepwise Refinement."* CACM Vol 14 (4), 1971

[3] J.R. Abrial *"The B-Book."* Cambridge University Press, 1996

[4] J.R. Abrial *"Modeling in Event-B."* Cambridge University Press, 2010

Integrating Event-B Modelling and Discrete-Event Simulation to Analyse Resilience of Data Stores in the Cloud

E. Troubytsina, Abo Akademi University, Turku, Finland

Development of complex distributed systems that meet the desired functional and performance requirements is a challenging engineering task. While creating a specification of the system functional behaviour, we implicitly introduce some constraints on the behaviour of the system components that might be undermined after the performance optimisation. Often an increase in system performance is achieved such modifications of the system architecture that result in changing the mechanisms of processing service requests or communication between the components. To make the implicit assumptions about system constraints explicit, we need to support multi-view modelling and analysis. In my talk, I discuss an approach that we have proposed integrates Event-B modelling and discrete event simulation. The approach aims at facilitating development and ensuring resilience of complex distributed systems-cloud data stores.

Ensuring resilience of large data stores in the cloud is a challenging engineering issue. It requires the development techniques that allow the designers to predict the main resilience characteristics – fault tolerance and performance – at the early design stages. We experiment with integrating Event-B modelling with discrete event simulation. Event-B allows us to reason about correctness and data integrity properties of data stores, while discrete-event simulation in SimPy enables quantitative assessment of performance and reliability. Since testing in a real cloud environment is expensive and time-consuming, the proposed approach offers several benefits in the industrial settings.

Making the Argument in Assurance Cases Explicit, Precise and Well Founded

V. Cassano, T. Maibaum, S. Grigorova, McMaster University, Canada

The introduction of safety cases has proved to be a step in the right direction in regards to safety assurance. As presently practiced, safety cases aim at making a serious attempt to explicate, and to provide some structure for, the reasoning involved in assuring that a system is safe, generally in terms of so-called structured arguments. However, the fact current notations for expressing these structured arguments have no formal semantics and, at best, are loosely linked to goal structuring ideas and to Toulmin's notion of an argument pattern, is a crucial issue to be addressed. History clearly demonstrates that languages that have no formal semantics are deficient in relation to the requirements of a serious approach to engineering. In other words, one can only go so far with intuition, and certainly not far enough to justify the safety of complex systems, such as Cyber Physical Systems or autonomous cars. Making explicit the logical semantics of safety arguments, as expressed, for example, in a notation such as GSN, is n important long term goal in safety practice. By rehearsing Gentzen's program for formalising mathematical reasoning, his famous Calculus of Natural Deduction, we show how we can begin a program of formalising safety reasoning by developing a working definition of a structured argument in a safety case and a calculus for safety reasoning.

Explicit Semantics in Formal Validation of Railway Data

L. Voisin, Systerel, Aix-En-Provence, France

In the railway domain, configuration data are a first-class citizen and need to be validated independently of the generic software. This validation can be performed formally by first writing B predicates on the data which express the expected properties, and then evaluate that the actual data satisfy these predicates using a dedicated tool such as Ovado.

But when formalising properties, a lot of implicit knowledge gets lost in translation. For instance, measures of different nature can be freely mixed (they are just integers in the formal language) even when this does not make sense physically (e.g., adding a length and a duration). Moreover, objects on the network are usually located using two frames of reference: a physical frame based on large linear portions of the networks, and a logical frame based on oriented blocks which are much smaller. Again, measures in these two frames can be freely mixed in properties which can be meaningless.

In the ANR research project IMPEX, Systerel and IRIT are currently working at formalising such implicit domain knowledge in an ontology in order to reinforce the static checks that can be performed on the B predicates, avoiding the pitfalls exposed above.

Marrying Processes and Data: Modelling and Verification

D. Calvanese, Free University of Bozen-Bolzano, Italy

Data and processes are just two sides of the same coin, and for several activities related to the analysis and design of systems it is essential to capture both static and dynamic aspects in a uniform way. In recent years, we have seen various proposals that aim at marrying these two aspects, and that consider both the process controlling the dynamics and the manipulation of data as equally central. We present Data-centric dynamic systems (DCDSs), which are a pristine model that abstracts from specific features of concrete formalisms proposed in the literature. We discuss recent results on decidability of verification of expressive (first-order) temporal properties over such systems. We also present some variations and extensions of the model that make it attractive both as a theoretical tool and for concrete realisations.

Exploring explicit semantics for maintainability and reusability in formal refinement (of Event-B)

Fuyuki Ishikawa, National Institute of Informatics, Japan

One of the key challenges for system dependability is how to deal with the increasing complexity in system modelling and verification. The Event-B method tackles this point with its flexible refinement mechanism. It is possible to gradually introduce and verify concepts and constraints in the system while moving from abstract, prescriptive representations into concrete, realisable representations. Due to its flexibility, the refinement mechanism requires design of the refinement steps. In other words, we need to examine how symbols, predicates, and their proofs in the whole specification are decomposed and modularised into the refinement steps.

How does the refinement design affect use of an Event-B model? Obviously, it affects difficulty in construction and verification of the model, which is the original motivation of the refinement mechanism in Event-B. The design also affects comprehensibility of the model and that of its proofs. These points can be contained in the broad term of maintainability (*easy to understand, verify, and validate?*). In addition, the refinement design can affect reusability (*easy to make changes in certain aspects?*). Changes in an early abstract step may affect the succeeding concrete steps, specifically their correctness with respect to "proper inheritance" of the abstract step. It is thus easier to make changes on concrete steps than on abstract steps. In other words, abstract steps can be reusable by just modifying or swapping the concrete steps.

Maintainability and reusability come from the characteristics of the predicates (requirements and domain assumptions) about the system, e.g., how stable or fragile each predicate is expected to be through time passage and how common or specific it is among variants in a product family. Such characteristics do not appear explicitly in the formal model of Event-B but should be taken into consideration upon the refinement design.

We have been investigating engineering methods for refinement to deal with aspects including the ones discussed above. Our recent work showed how refactoring of refinement can be realized and how it helps improvement in maintainability and reusability. Our experience with refinement refactoring let us be aware of many essential aspects that are not explicitly specified in the formal model.

For example, our refactoring method includes slicing of a refinement step to divide it into two steps and we need to be careful not to break the existing proofs (or need to fix if we break). In the formal model, it is implicit which predicates essentially contribute to the proof of each predicate, as proof obligations just require that all of the available predicates imply the necessary predicate. Refinement refactoring becomes very easy once this dependency between predicates is made explicit.

We will continue to investigate what information should be made explicit to support engineering activities on refinement, or on general proof-based development methods.

Entity Resolution Meets Formal Methods

Q. Wang, Australian National University, Australia

Entity resolution (ER) is concerned with deciding whether two representations of entities refer to the same real-world object. It is one of the major impediments affecting data quality provided by information systems. The difficulty of this problem has been widely acknowledged by various research communities and industry practitioners. State-of-the-art approaches to entity resolution mostly favor similarity-based methods. In this talk, I will present a simple yet expressive framework that can support knowledge-based entity resolution. Knowledge patterns, as the building blocks of the framework, have the capability of capturing knowledge about different entities at an arbitrary level of abstraction. From a logical point of view, the expressive power of the framework is equivalent to a fragment of first-order logic including conjunction, disjunction and a certain form of negation. Nonetheless, given an ER task, how can we find out the meaningful knowledge patterns? Furthermore, given a knowledge model that consists of knowledge patterns, how can we guarantee the reliability and correctness of the knowledge model? These questions lead to an interesting area of research linking entity resolution and formal methods.

Houston, we have a problem: Implicit semantics in Engineering models

M. Lawford, McMaster University, Canada

In this talk I describe some well known aerospace system failures an interpret them from the point of view of trying to understand how implicit semantics of the systems led to the failures. It is not clear that formal methods in their current form would have prevented these failures. Then I discuss two different engineering models

1. Matlab/Simlink/Stateflow
2. IEC 61131-3 Function Blocks

describing some of their implicit semantics and asking the following questions.

- What issues have there been with implicit semantics of models created using these two modelling languages?
- How have the IEC 61131 standard and Matlab/Simulink changed to address these issues?
- How should they be changed?
- What should remain implicit in the interest of modelling efficiency?

I try to make the case that care must be taken in making the implicit explicit or the results may be a host of new problems in place of the old.

A formalization in Coq of abstract machines and refinements

P. Castéran, Labri, Univ. Bordeaux, Bordeaux, France.

The goal of our work is to derive explicit information from the structure of event-B developments. Such a development is mainly composed of a set of contexts and abstract machines, decorated with invariants, theorems and refinement declarations. Given such a development, Rodin generates a set of proof obligations, which are theorems statements that must be proved, automatically or interactively.

In order to make explicit the semantics of event-B developments, we are building a theory that considers abstract machines and their behaviours, simple or gluing invariants, proof obligations, as first-class objects. This is made possible thanks to the great expressive power of the Coq proof assistant's underlying logic.

In this talk, we show how to express in Coq the basic notions of event-B: events, invariants, machines and refinements, and how to prove generic theorems that can be applied to concrete simple cases.

We hope this semantics will be used to make explicit the meaning of each proof obligation of the Rodin tool, for teaching reasons, and also serve as an interface between Rodin and the usual theories of reactive systems and their semantics.

Where do the proofs belong in an assurance case?

A. Wassyng, McMaster University, Canada

Despite the original promise of safety and assurance cases, current practice is to leave the argument structure implicit. We make the case that the reasoning in the argument structure should be explicit. We also examine the different kinds of "proofs" in assurance cases:

i) the argument structure of the assurance case; and

ii) the proofs in the evidence nodes that support terminal claims in the assurance case.

Finally, we present reasons why even safety cases should not be structured on hazard analyses, and that completeness arguments must be made more explicit

Domain-specific development with Rodin Theories

Thai Son Hoang, ECS, University of Southampton, United Kingdom

The Theory plug-in for the Rodin Platform (Rodin) enables modellers to extend the mathematical modelling notation for Event-B, with accompanying support for reasoning about the extended language. We consider in this presentation using Rodin theories to capture domain-specific abstract datatypes (ADTs) and build dynamic systems using the developed structures. In particular, we proposed the notion of theory instantiation to incorporate more concrete representation of the ADTs. At the same time, the dynamic systems is refined further with respected to the changes of the underlying ADTs. We illustrate our approach with an industrial example of developing a CBTC train control system. We anticipate that by theory instantiation is a promising direction for reusing theries via abstraction.

Keywords ADTs; Rodin; Event-B; Theory instantiation; CBTC.

Assurance Cases and their ~~Arguments~~ Evidence

J. Rushby, SRI International, USA

An assurance case uses a structured argument to justify claims about a system, based on evidence about its context, design, and construction. The notions of "argument," "justification," and "evidence" raise questions in epistemology and logic as old as philosophy itself, but in an interesting new context. I will discuss these, focusing on a suitable notion of argument.

See: `http://www.csl.sri.com/~rushby/abstracts/aaa15` and
`http://www.csl.sri.com/~rushby/abstracts/assurance-cases15`

Model based software management (and formal semantics)

F. Khendek, Univ. Of Concordia, Montreal. Canada

Software management is the field of managing software artefacts after they have been developed. The main activities are software configuration and upgrade. In this presentation we will discuss the challenges of software configuration and upgrade in the context of high-availability. We will see how Model Driven Engineering (MDE) can help in alleviating the challenges. We will also discuss some semantic issues in relation to topic of the workshop

A Formal Ontological Analysis in Medical Domain

N. Singh, INPT-ENSEEIHT/IRIT, Toulouse, France

Medical domain is one of the challenging area that mainly offers the maintenance of health, prevention and treatment of disease through covering the range of sub domains, such as anatomy, physiology, pathology, pharmacology and neuroscience. For instance, clinical guidelines systematically assist practitioners to provide appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. The prime motivation of this work is to find anomalies and to improve the quality of medical protocols using mathematical formal reasoning. This work proposes a stepwise formal development for modelling the medical domain knowledge using ontologies and then the developed domain model is used for developing and verifying the medical protocols or guidelines. The progressive development allows to enrich domain model and medical protocol model, in which the developed ontological model helps to verify domain-related properties.

In this work, we use the Event B language for modelling domain model using ontologies and to capture the functional behaviour of the medical protocols or guidelines for their validation. Our main contributions are: to apply mathematical formal techniques to evaluate real-life medical protocols for quality improvement; to derive verification proofs for the protocol and properties according to medical experts; and to publicise the potential of this approach. An assessment of the proposed approach is given through a case study, relative to a real-life reference protocol (ECG interpretation), which covers a wide variety of protocol characteristics related to several heart diseases and it is the most applied test for mapping the heart activity. In this work, we present an ontology of the electrocardiogram (ECG). The ontology purpose is to bring in a theory of the electrocardiogram (ECG). This developed ontology of ECG is used to describe ECG medical protocol formally in progressive manner to detect abnormalities and malformations. Moreover, this developed ECG ontology can be reused in some other applications related to ECG.

An Algebra of Lightweight Ontologies -Implementation and Applications

M. Casanova, PUC, Rio De Janeiro, Brazil.

We argued elsewhere that certain familiar ontology design problems are profitably addressed by treating ontologies as theories and by defining a set of operations on ontologies [2,3]. Briefly, we define an ontology as a pair $O = (V, \Sigma)$ such that V is a vocabulary and Σ is a set of constraints in V. The theory of Σ is the set of all constraints that are logical consequences of Σ. The theory of Σ identifies the constraints that are implicitly defined, but which must be considered when using the ontology. The operations we propose create new ontologies, including their constraints, out of other ontologies. Consider first the problem of designing an ontology to publish data on the Web. If the designer follows the Linked Data principles, he must select known ontologies, as much as possible, to organise the data so that applications ”can dereference the URIs that identify vocabulary terms in order to find their definition”. We argue that the designer should go further and analyse the constraints of the ontologies from which he is drawing the terms to construct his vocabulary. Furthermore, he should publish the data so that the original semantics of the terms is preserved. To facilitate ontology design from this perspective, we introduce three operations on ontologies, called projection, union and deprecation. The projection operation is akin to the familiar modularisation operation. Consider now the problem of comparing the expressive power of two ontologies, $O1 = (V1, \Sigma1)$ and $O2 = (V2, \Sigma2)$. If the designer wants to know what they have in common, he should create a mapping between their vocabularies and detect which constraints hold in both ontologies, after the terms are appropriately mapped. The intersection operation answers this question. We argued elsewhere [1] that intersection is also useful to address the design of mediated schemas that combine export schemas in a way that the data exposed by the mediator is always consistent. On the other hand, if the designer wants to know what holds in $O1 = (V1, \Sigma1)$, but not in $O2 = (V2, \Sigma2)$, he should again create a mapping between their vocabularies and detect which constraints hold in the theory of $\Sigma1$, but not in the theory of $\Sigma2$, after the terms are appropriately mapped. The difference operation answers this question. Likewise, if the user wants to analyse what changed from one version of an ontology to the other, he should also use the difference operation. We developed a tool, called OntologyManagerTab [4], that implements the operations over lightweight ontologies, whose constraints correspond to DL-Lite core with number restrictions. The implementation depends on the notion of constraint graphs, introduced in [1]. OntologyManagerTab is a tab plug-in over Protg 3.4.8, but it works in a completely independent manner from the main framework, using Protg only as a Graphical User Interface (GUI) enclosure.

References

[1] Casanova, M.A., Lauschner, T., Leme, L.A.P.P., Breitman, K.K., Furtado, A.L., Vidal, V.M.P., 2010. *”Revising the Constraints of Lightweight Mediated Schemas”*. Data & Knowledge Engineering 69(12), 1274-1301.

[2] Casanova, M.A., Breitman, K.K., Furtado, A.L., Vidal, V.M.P., Macedo, J.A.F., 2011. *”The Role of Constraints in Linked Data”*. Proceedings of the Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, Part II. Lecture Notes in Computer Science v. 7045. Springer, 781-799.

[3] Casanova, M.A., Macedo, J.A.F., Sacramento, E., Pinheiro, A.M.A., Vidal, V.M.P., Breitman, K.K., Furtado, A.L., 2012b. *”Operations over*

Lightweight Ontologies". Proc. 11th International Conference on Ontologies, DataBases, and Applications of Semantics - ODBASE 2012 (Sept. 11-12, 2012), Rome. LNCS 7566, 646-663.

[**4**] Magalhaes, R.C. *"Operations over Lightweight Ontologies"*. M.Sc. Dissertation, Department of Informatics, PUC-Rio, Rio de Janeiro, Brazil (2015). Available at: `http://www.inf.puc-rio.br/~casanova/Publications/Dissertations-Theses/2015-Romulo.pdf`.

A case study of proof-based engineering on some practical software system

H. Yatsu, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan

Medical domain is one of the challenging area that mainly offers the maintenance of health, prevention and treatment of disease through covering the range of sub domains, such as anatomy, physiology, pathology, pharmacology and neuroscience. For instance, clinical guidelines systematically assist practitioners to provide appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. The prime motivation of this work is to find anomalies and to improve the quality of medical protocols using mathematical formal reasoning. This work proposes a stepwise formal development for modelling the medical domain knowledge using ontologies and then the developed domain model is used for developing and verifying the medical protocols or guidelines. The progressive development allows to enrich domain model and medical protocol model, in which the developed ontological model helps to verify domain-related properties.

In this work, we use the Event B language for modelling domain model using ontologies and to capture the functional behaviour of the medical protocols or guidelines for their validation. Our main contributions are: to apply mathematical formal techniques to evaluate real-life medical protocols for quality improvement; to derive verification proofs for the protocol and properties according to medical experts; and to publicise the potential of this approach. An assessment of the proposed approach is given through a case study, relative to a real-life reference protocol (ECG interpretation), which covers a wide variety of protocol characteristics related to several heart diseases and it is the most applied test for mapping the heart activity. In this work, we present an ontology of the electrocardiogram (ECG). The ontology purpose is to bring in a theory of the electrocardiogram (ECG). This developed ontology of ECG is used to describe ECG medical protocol formally in progressive manner to detect abnormalities and malformations. Moreover, this developed ECG ontology can be reused in some other applications related to ECG.

Integrating domain knowledge in formal requirements engineering

R. Laleau, LACL-UPEC, Créteil, France, A. Mammar, SAMOVAR, Telecom SudParis CNRS, Université Paris-Saclay, France

The framework of the work presented during the Shonan meeting is the French project founded by the ANR (National Research Agency. Project ANR-14-CE28-0009) and called FORMOSE (`http://formose.lacl.fr`). The aim of the project is to provide a formally-grounded, model-based requirements engineering (RE) method, for critical complex systems, supported by an open-source environment. A RE method includes the elaboration of a requirements model and a domain model to specify information about the application domain. To build a requirements model, we need a RE language. It must be multi-views. Indeed, requirements have to be expressed in natural language and also with graphical notations to be validated by the different stakeholders who participate to the construction of the RE model. Finally, formal notations are necessary in order to formally verify requirements because we are dealing with critical systems. In our project we have chosen to use a goal- based language, called SysML/KAOS [1], to express functional and non-functional requirements and a combination of existing formal methods, namely EventB [2] and UPPAAL [3]. In a previous work, we have defined a set of rules to derive a partial Event-B specification from a SySML/KAOS goal model [4]. However, the semantics of goal models is different from the usual Event-B semantics given by the proof obligations defined by J.R. Abrial [2]. So, we defined new proof obligations to explicitly express the semantics of goal refinement. A domain model is described by a domain ontology and a class diagram (and possibly object diagrams). Domain ontology is used to explicit and make clearer domain knowledge. By modelling shared knowledge, it fosters a common understanding of applications, particularly in complex systems design that involve various kinds of stakeholders. A class and object diagrams are specified to detail the components of a specific system. They must be consistent with the domain ontology. Finally, these elements are translated into Event-B to complete the formal specification obtained from the goal model [5].

References

[1] C. Gnaho, F. Semmak, R. Laleau.: *"Modeling the impact of non-functional requirements on functional requirements."* ER Workshops 2013, LNCS 8697, Springer, 2013.

[2] J. R. Abrial: *"Modeling in Event-B: System and Software Engineering."* Cambridge University Press, 2010.

[3] K. Guldstrand Larsen, P. Pettersson, and W. Yi.: *"UPPAAL in a nutshell."* STTT, 1(1-2):134-152, 1997.

[4] A. Matoussi, F. Gervais, R. Laleau: *"A goal-based approach to guide the design of an abstract Event-B specification."* 16th IEEE Int. Conf. on Engineering of Complex Computer Systems, 2011.

[5] A. Mammar, R. Laleau: *"On the Use of Domain and System Knowledge Modelling in Goal-Based Event-B Specifications."* 7th International

Symposium on Leveraging Applications of Formal Methods, Verification and Validation. ISOLA 2016, LNCS 9952, Springer, 2016.

Safety, privacy and liveness of medical access control policies

M. Frappier, University of Sherbrooke, Sherbrooke, Canada, R. Laleau, LACL-UPEC, Créteil, France, A. Mammar, SAMOVAR, Telecom SudParis CNRS, Université Paris-Saclay, France

We investigate the verification of access control policies for SGAC, a new healthcare access-control model, using Alloy and ProB, two first-order logic model checkers based on distinct technologies. SGAC supports permission and prohibition, rule inheritance among subjects and resources ordered by acyclic graphs; conflicts are autonomously managed using rule precedence based on priority, specificity and modality. In order to protect patient privacy while ensuring effective caregiving in safety-critical situations, we check four types of properties: accessibility, availability, contextuality and rule effectivity. Our performance results show that ProB performs two orders of magnitude better than Alloy, thanks to its programmable approach to constraint solving. Results are promising enough to consider ProB for verifying patient policies in SGAC.

A Case-study of Abstract Model Instantiation in Event-B

H. Kuruma, Hitachi, Ltd. Japan

In large scale system developments, many assumptions relating to physical and social restrictions need to be considered. Also, a variety of system designs are produced to satisfy customers' demands. Usually, domain engineers clarify assumptions on environments and system designers write specifications of the system. The work of domain engineers and system designers define the abstract model and the concrete model, which is an instantiation of the abstract model, respectively. In this presentation, I show our case study of train monitoring system formalisation. The system gets state of equipment from the centralised train control system and displays it for the railway operators. Since the vocabulary for equipment state is shared by them, it is necessary to verify that the system design interprets the vocabulary correctly. We constructed an abstract model and instantiated it using refinement and generic instantiation and verified the correctness of interpretation using invariants in Event-B.

A Formal Framework for the Design of Software Components with the B method

David Déharbe (ClearSy Systems Engineering, France & UFRN, Brazil

We explicit the semantics for the B method in terms of labelled transition systems (LTSes), using the Isabelle/HOL proof assistant. In this model, a B component is identified with a labelled-transition system (LTS). The LTS structure is parameterised with a type for component states and events, which are left abstract in this work. We identify the internal and external behaviour of the B component with the runs and traces of the labelled transition system. The B method concept of refinement is then identified as a simulation relation between LTSes. We specialise this definition of simulation to take into account the fact that the refinement of a component is allowed to weaken the guard of events. Then several properties of the refinement relation have been stated and shown correct. This is a first step towards the definition of a formal framework that would allow to establish provably correct refactoring laws and a formal refinement calculus for B.

Distribution and temporal behaviour patterns

M. Filali, CNRS-IRIT, Toulouse, France.

Traditionally, formal methods have been mostly concerned by producing correct and certified code. Recently, the use of formal methods has shifted to requirements. Actually, formal methods are more and more used to formalise as well high level requirements as well as domain specific skills. In this talk, we discuss how patterns could be used in order to generate Event-B refinements automatically. We are interested in behavioural patterns formalised as Büchi automata. One of our major concern is to produce Event-B machines such that the user can refine them further. Our ultimate goal is to produce certified code for distributed platforms starting from high level requirements.

Light-weight formalisation and verification of safety requirements for ISO 26262

T. Aoki, JAIST Japan

In ISO 26262, safety requirements are constructed step by step. The construction is started to set safety goals to be achieved in a system up, then they are refined into hardware and software requirements which the system consists of. Such stepwise construction of the safety requirements provides traceability among them and allows us to confirm that the system surely realises the goals. The traceability also helps us to exhaustively extract requirements which are necessary to achieve safety. On the other hand, the quality of a document describing them is important to obtain those merits. If the document contains ambiguities, contradictions and many of requirements are missed, those lead to the unsafety of the system. In fact, we found many of missing implicit assumptions and ambiguous requirements by analysing a document which describes safety requirements. To solve this problem, we proposed a method to describe the safety requirements based on the goal tree of KAOS and its patterns. We confirmed the effectiveness of the method by applying it to an electronica power steering system as a case study. In this talk, we show the case study which is not trivial but a real system in addition to the proposed method.

Iterative language specification - Making the implicit explicit

M. Pantel, INPT-ENSEEIHT/IRIT, Toulouse, France

Models play a key prescriptive role in engineering as they make explicit the various aspects of the system and its development artefacts. They enable both the use of tools that automate parts of the development and formal methods that provide a higher confidence. The complexity of the currently developed systems such as Distributed or Cyber Physical Systems presents a problem to the language engineering community as many different aspects of the system must be made explicit and many different tools are required to improve their development. The current state of practice has shown that: a) general purpose languages and tools that aims at modelling everything and providing generic services first have a very high development cost; then usually miss their target as there always exists some elements that cannot be modelled in a simple and precise manner; and last only provide simple services that only gives a shallow perception of the system. This contribution first advocates the use of domain specific modelling languages and tools dedicated to some aspects of the system and specific development services; and then the iterative specification of these languages in order to only make explicit the language concepts required to make explicit the required elements in the system; and the aspects of these concepts needed to build tools that provide the appropriate services. Each time a new aspect of the system must be made explicit, or a new tool must be integrated in the development process, the modelling languages must be extended in order to provide the appropriate elements for building the models and the associated tools. This language development methodology both reduces the cost, eases the language and tool development and improves their quality. This talk relies on the Block Library Specification language case study that allowed to extend the specification of the Simulink language in order to make explicit additional aspects in order to integrate formal methods in the usual simulation based validation and verification activities.

Verification of natural language requirements using ontologies

C. Dubois, ENSIIE, Evry, France

The objectives of the work presented in this talk is to verify user requirements written in natural language in the context of a case study about smart homes, in order to discover incompleteness or inconsistency and to provide technical elements for deployment. The user describes her requirements as behavioural rules using everyday life terms and not technical ones such as sensors, actuators or controllers. Our approach consists in first designing an OWL ontology that describes the domain knowledge, more precisely the generic behaviour of a smart home, introducing concepts such as sensors, actuators, behavioural rules and dedicated relations. It allow us to bridge the gap between informal and formal languages and to automate the transformation of NL rules into a Maude specification.

Modelling an e-voting domain for the formal development of a Software Product Line: when the implicit should be made explicit

J Paul Gibson and Jean-Luc Raffy Telecom Sud Paris, Evry, France

There has been much recent interest in the development of electronic voting (e-voting) systems, but there remain many outstanding research challenges for software and system engineers [1]. Software product line techniques offer many advantages for the practical development of reliable and trustworthy e-voting systems, but the composition of system features poses significant problems that can be addressed satisfactorily only through the use of formal methods [2]. When such systems are used in government elections then they are obliged to follow legal standards and/or recommendations written in natural language [3]. For the formal development of e-voting systems it is necessary to build a domain model which is consistent with the legal requirements [4]. We have already demonstrated that Event-B models can be used to verify critical requirements for e-voting system components [5,6]. However, the refinement-based approach needs to be applied to the engineering of a complete e-voting system. We report on our approach, using Event-B contexts to model an e-voting ontology, and its integration with an e-voting features model tree which formally specifies the SPL. During this work, we identified the importance of making the implicit explicit in 2 different ways - domain experts need to explicitly model implicit knowledge, and Event-B modellers need to explicitly communicate the semantics of the formal model constructs to the domain experts. If either of these tasks is not adequately carried out then this compromises validation of the requirements model (instance of the SPL).

References

[1] A Review of E-voting: the past, present and future, J Paul Gibson, Robert Krimmer, Vanessa Teague and Julia Polmares. Springer Annals of Telecommunications, volume=71, number=7, pages=279–286, July 2016

[2] Feature Interactions in a Software Product Line for E-voting, J. Paul Gibson, Eric Lallet, Jean-Luc Raffy. Feature Interactions in Software and Communication Systems X, Nakamura and Reiff-Marganiec (editors), pages 91 - 106, IOS Press, ISBN 9781607500148.

[3] Verification and Maintenance of e-voting systems and standards, J. Paul Gibson and Margaret McGaley. In proceedings ECEG 2008, the 8th European Conference on e-Government Ecole Polytechnique, Lausanne, Switzerland, 10-11 July 2008, pages 283-290, editor Dan Remenyi, published by Academic Publishing International, ISBN 978-1-906638-09-2

[4] Engineering a distributed e-voting system architecture: meeting critical requirements, J. Paul Gibson, Eric Lallet, Jean- Luc Raffy. Accepted at: the 1st International Symposium on Architecting Critical Systems (ISARCS10). Published in: Architecting Critical Systems (Springer LNCS 6150), editor Holger Giese, ISBN 978-3-642-13555-2, pages 89-108.

[5] Refinement: a constructive approach to formal software design for a secure e-voting interface, Dominique Cansell, J Paul Gibson, and Do-

minique Méry. Electronic Notes in Theoretical Computer Science, 183 (2007), pages 39-55, Elsevier, ISSN 1571-0661.

[6] Formal verification of tamper-evident storage for e-voting, Dominique Cansell, J Paul Gibson, and Dominique Mry, in Proceedings of 5th IEEE International Conference on Software Engineering and Formal Methods (SEFM07), London, 10-14 September 2007. Published by IEEE Computer Science Press, pages 329-338, ISBN = 0-7695-2884-8, editors Mike Hinchey and Tiziana Margaria.

Next steps and closing of the meeting

Y. Ait-Ameur, S. Nakajima, D. Méry.

Concluding remarks have been raised by the seminar organisers. The participants have discussed the next steps and how this fruitful meeting can be fertilised. The decision to edit a book summarising the contributions and discussions has been taken.

List of Participants

Organisers

- Prof. Yamine AIT-AMEUR INPT-ENSEEIHT/IRIT, France
- Prof. Shin NAKAJIMA National Institute of Informatics, Tokyo, Japan
- Prof. Dominique MÉRY LORIA, Universit de Lorraine, Nancy, France

Participants

- Jean-Raymond ABRIAL independent, Marseille, France
- Prof. Toshiaki AOKI JAIST Japan
- Prof. Mohand BOUGHANEM, IRIT-Université de Toulouse, France
- Prof. Diego CALVANESE Free University of Bozen-Bolzano, Italy
- Prof. Marco Antonio CASANOVA Pontifical Catholic University of Rio de Janeiro, Brasil
- Prof. Pierre CASTÉRAN University of Bordeaux Labri, Bordeaux, France
- Prof. David DEHARBE Clearsy France
- Prof. Catherine DUBOIS ENSIIE, Evry, France
- Prof. Marc FRAPPIER Université de Sherbrooke, Canada
- Prof. J. Paul GIBSON Telecom Sud Paris, France
- Dr. Thai Son Hoang, ECS, University of Southampton, United Kingdom
- Prof. Fuyuki ISHIKAW A National Institute of Informatics, Tokyo, Japan

- Prof. Ferhat KHENDEK Concordia University , Montreal, Canada
- Dr. Hironobu KURUMA Hitachi Co. Ltd. Japan
- Prof. Régine LALEAU Université Paris-Est Créteil, France
- Prof. Mark LAWFORD McMaster University, Canada
- Prof. Thomas MAIBAUM McMaster University, Canada
- Prof. Marc PANTEL IRIT/Université de Toulouse, France
- Prof. Jean-Luc RAFFY Tĺecom Sud-Paris, France
- Dr. John RUSHBY SRI International, USA
- Prof. Neeraj Kumar SINGH INPT-ENSEEIHT/IRIT, University of Toulouse France
- Prof. Elena TROUBITSYNA Abo Akademi University, Turku, FINLAND
- Dr. Laurent VOISIN Systerel , Aix-En-Provence, France
- Dr. Qing WANG , Australian National University, Australia
- Prof. Alan WASSYNG McMaster University, Canada
- Dr. Hirokazu YATSU, Kyusyu University, Japan
- Prof. Mamoun FILALI AMINE IRIT-CNRS, France

Meeting Schedule

Check-in Day: Monday Nov. 21

- Welcome Banquet

Day1: Tuesday Nov. 22nd. Room 208

09:00 - 09:30 **Y. Ait-Ameur, S. Nakajima, D. Méry** *"Introduction to the seminar and organisation issues"*

09:30 - 10:00 **All speakers** *" One slide presentation (5 Min)"*

10:00 - 10:30 **All speakers** *" One slide presentation (5 Min)"*

10:30 - 11:00 **Coffee break**

11:00 - 11:30 **J. R. Abrial** *"On Modelling Integration"*

11:30 - 12:00 **E. Troubytsina** *"Integrating Event-B Modelling and Discrete-Event Simulation to Analyse Resilience of Data Stores in the Cloud"*

12:00 - 13:30 Lunch

13:30 - 14:00 **Group Photo**

14:00 - 14:30 **T. Maibaum** *"Making the Argument in Assurance Cases Explicit, Precise and Well Founded"*

14:30 - 15:00 **L. Voisin** *"Explicit Semantics in Formal Validation of Railway Data"*

15:00 - 15:30 **D. Calvanese** *"Marrying Processes and Data: Modeling and Verification"*

15:30 - 16:00 **Coffee break**

16:00 - 16:30 **F. Ishikawa** *"Exploring explicit semantics for maintainability and reusability in formal refinement (of Event-B)"*

16:30 - 17:00 **Q. Wang** *"Entity Resolution Meets Formal Methods"*

17:00 - 17:30 **Debriefing**

- Talks and Discussions
- Group Photo Shooting

Day 2: Wednesday Nov. 23rd. Room 208.

09:00 - 09:30 **M. Lawford** *"Houston, we have a problem: Implicit semantics in Engineering models "*

09:30 - 10:00 **P. Casteran** *"A formalization in Coq of abstract machines and refinements"*

10:00 - 10:30 **A. Wassyng** *" Where do the proofs belong in an assurance case ?"*

10:30 - 11:00 **Coffee break**

11:00 - 11:30 **Thai Son Hoang** *"Domain-specific development with Rodin Theories"*

11:30 - 12:00 **J Rushby** *"Architectural Models For Self-Integrating Systems"*

12:00 - 13:30 Lunch

13:30 - 14:00 **F. Khendek** *"Model based software management and formal semantics"*

14:00 - 14:30 **N. Singh** *"A Formal Ontological Analysis in Medical Domain"*

14:30 - 15:00 **M. Casanova** *"An Algebra of Lightweight Ontologies -Implementation and Applications"*

15:00 - 15:30 **H. Yatsu** *"A case study of proof-based engineering on some practical software system"*

15:30 - 16:00 **Coffee break**

16:00 - 16:30 **R. Laleau** *"Integrating domain knowledge in formal requirements engineering"*

16:30 - 17:00 **M. Frappier** *"Safety, privacy and liveness of medical access control policies"*

17:00 - 17:30 **Debriefing**

- Talks and Discussions

Day3: Thursday Nov. 24th. Room 208

09:00 - 09:30 **H. Kuruma** *"A Case-study of Abstract Model Instantiation in Event-B"*

09:30 - 10:00 **D. Deharbe** *"A Formal Framework for the Design of Software Components with the B method"*

10:00 - 10:30 **M. Filali** *"Distribution and temporal behaviour patterns"*

10:30 - 11:00 **Coffee break**

11:00 - 11:30 **T. Aoki** *"Light-weight formalisation and verification of safety requirements for ISO 26262."*

11:30 - 12:00 **M. Pantel** *"Iterative language specification - Making the implicit explicit"*

12:00 - 13:30 Lunch

13:30 - 22:30 **Social programme**

- Talks and Discussions
- Excursion and Main Banquet

Day4: Friday Nov. 25th. Room 208.

09:00 - 09:30 **C. Dubois** *"Verification of natural language requirements using ontologies"*

09:30 - 10:00 **P. Gibson** *"Modelling an e-voting domain for the formal development of a Software Product Line: when the implicit should be made explicit "*

10:00 - 10:30 **Debriefing**

10:30 - 11:00 **Coffee break**

11:00 - 11:30 **Group animators** *"Feedback"*

11:30 - 12:00 **Group animators** *"Feedback"*

12:00 - 14:00 Lunch

14:00 - 15:00 **Y. Ait-Ameur, S. Nakajima, D. Méry.** *Next steps and closing of the meeting.*

- Talks and Discussions
- Wrap up