FIRST STEPS TOWARD A VERIFICATION AND VALIDATION ONTOLOGY

Mounira Kezadri and Marc Pantel
IRIT, Université de Toulouse, ENSEEIHT 2 rue C. Camichel, Toulouse, France

Keywords: Ontology, Modeling languages, Verification and validation technologies.

Abstract: This paper presents the key elements of an ontology that formalizes part of the knowledge about behavioural modeling and the associated verification and validation technologies. It summarizes the objects, concepts, and other entities that are assumed to exist in this area of interest and the relationships that hold among them. We propose a classification of different modeling formalisms and a representation of possible verification and validation methods. A system is represented using several views conforming to different modeling languages. Its properties can be assessed with verification and validation technologies.

1 INTRODUCTION

Verification and validation (V&V) is the process of checking that a system satisfies its requirements. Verification relates a system implementation and its specification whereas validation relates a system and its end users. If the requirements have been specified then the system is verified against this specification, and the specification is validated against its end users. This paper presents the preliminary elements of a verification and validation Ontology (VVO[1]), which represents a knowledge sharing (Neches et al., 1991) initiative for the V&V domain. The classification is mainly based on the state of the art of CESAR[2] and Ptolemy[3] projects and others web sources. The ontology (Guarino and Giaretta, 1995) defines the verification and validation methods that can be applied on a system whose behaviour has been modeled. VVO is being used as a communication language, as a foundation for other engineering ontologies, and later to constitute a global verification and validation platform for a set of V&V techniques. In this paper we describe the conceptualization of the ontology, its design and motivate the major representation choices. Our contributions are: 1) the classification of system's description formalisms, 2) the classification of Verification and Validation methods, 3) the definition of relations between V&V technologies, formalisms and properties description languages. Our intention is first to make the knowledge of the behaviour modeling and V&V domains shareable and reusable and then to use the VVO as a guideline for choosing and applying dynamically the adapted V&V technologies in order to ease the development of safety critical systems. The VVO contains more than 250 classes. Is was developed using the Protégé tool-kit. It is available for reuse, comments and extension proposals at http://www.irit.fr/∼Mounira.Kezadri/Ontologies/VVO.owl.

The rest of this paper is organized as follows: In Section 2 the general architecture of the VVO, the definition of global concepts and relations between them are given. In Section 3 we present the case study of verification for Petri Nets. Section 4 discusses related work. Conclusion and future work appear in Section 5.

2 THE GENERAL STRUCTURE OF THE VVO

In this section, we present the general architecture of the ontology. A number of ontology modeling methods have been proposed in the literature (Gomez-Perez et al., 1991). The Web Ontology Language (OWL) (McGuinness et al., 2004) under the Protégé [4]

[1]This work was funded by the European Union and the french DGCIS through the ARTEMIS Joint Undertaking inside the CESAR project

[2]https://cesarproject.eu/

[3]http://ptolemy.eecs.berkeley.edu/

[4]http://protege.stanford.edu/

Kezadri M. and Pantel M..
FIRST STEPS TOWARD A VERIFICATION AND VALIDATION ONTOLOGY.
DOI: 10.5220/0003102904400444
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 440-444
ISBN: 978-989-8425-29-4

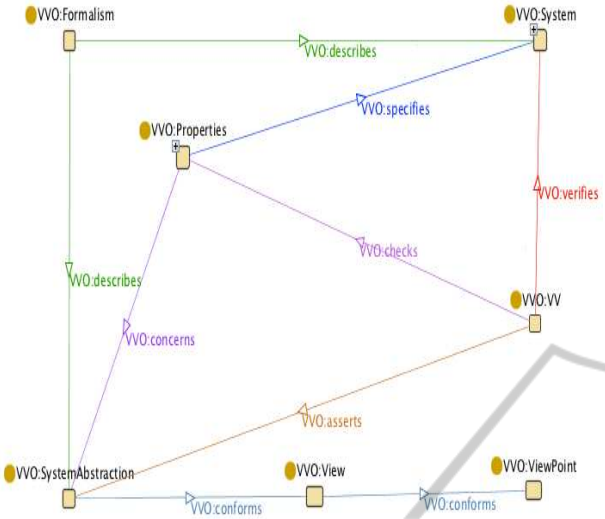

Figure 1: Global architecture of VVO.

4.1 Alpha is used to build the VVO and the Jambalaya tab [5] from the version 3.4.3 to obtain Figure1.

The most typical kind of ontology has a taxonomy defining the classes, objects and relations among them and a set of inference rules powering reasoning functions (Lee et al., 2001). The first phase for the ontology definition is the enumeration of important terms and relationships of the domain (in our case the domain of V&V). The figure 1 shows the global structure of our ontology, it represents its key concepts and the relations between them. One system can be described by different abstractions, every abstraction is expressed using a view conforming to a modeling language. We can express properties that specifies parts of the system. The system with, or without, associated properties can then be assessed with some V&V technique.

In the following subsections, we present the definition and the hierarchy of the principles concepts of the VVO.

2.1 An Ontology for Formalisms Description

System Concept. A system is a collection of components organized to accomplish a specific function or set of functions. One system can be composed from one or several components and described using several views expressed in modeling formalism.

Component Concept. The characteristic properties of a component are that it is both a unit of independent deployment, and of third-party composition; and it has no (externally) observable state (Clemens et al., 1998). A component can be described using views expressed in different formalisms, specified by properties and assessed with V&V technologies.

Formalism Concept. We consider this part of the ontology as a basis for structuring and constructing a domain-specific modeling tools. The Formalism's ontology collects a large number of widely-used formalisms for system's behaviour modeling. As the number of formalisms is quite important, we propose a classification of these formalisms shown in Figure 2.

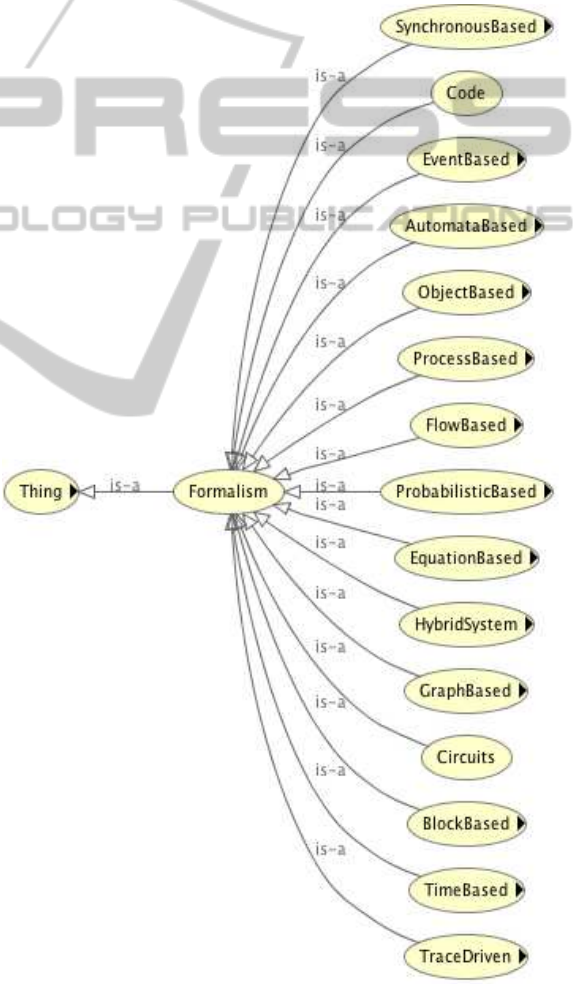

Figure 2: The first level of formalisms hierarchy.

We present the example of the hierarchy of the automata formalism which is a sub-class of automata-based formalisms, this hierarchy is composed of: Büchi automata, cellular automata (Cervelle et al., 2010), finite automata (Lawson,

[5] http://protegewiki.stanford.edu/wiki/Jambalaya_2.6.0

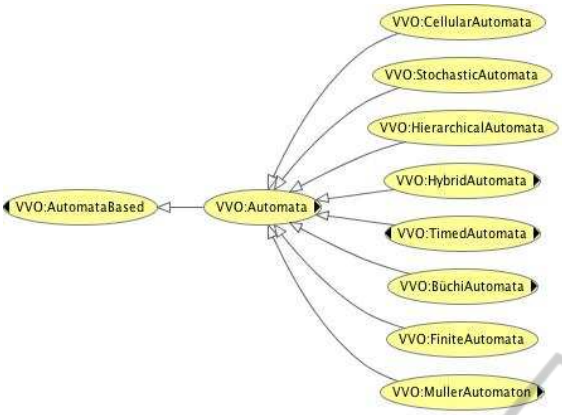

Figure 3: Automata based hierarchy.

2005), hierarchical automata (Mikk et al., 1997), Muller automaton (Perrin, 2004), stochastic automata (D'Argenio and Katoen, 2005), timed automata (Bengtsson and Yi, 2003), Ω automata (Krishnan et al., 1994) and hybrid automata (Alur et al., 1993). The hierarchy is illustrated in Figure 3. Each kind of this formalisms has special characteristics and have his own hierarchy.

2.2 An Ontology for Views

This part is mainly derived from the IEEE standard (Hilliard, 2000).

System Abstraction Concept. Most systems are generally very complex. Several abstractions are usually required in order to manage their description. An abstraction conforms to a view and it is modelled in some formalism.

View Concept. Is the representation of a whole system from the perspective of a related set of concerns. In our ontology a view must be conforming to at least one view point.

ViewPoint Concept. Is a specification of the conventions used for constructing and exploiting a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

2.3 An Ontology for Verification and Validation Techniques

V&V Concept. The most important and the largest part of this work is the classification of the V&V techniques. A wide variety of V&V strategies and techniques are available. A V&V technique (method or technology) can be applied to one or several abstractions of a system depending on the formalism used to describe the system and the property that must be assessed. We mean by verification techniques the tools to verify in some measure that a system satisfies some specification. We mean by validation technique the mean for the modeling language user to check that the model is a correct rendering of the idea he wanted to express. The goal is to increase the confidence that we have on the developed system, this can be done with different approaches. Figure 4 shows the hierarchy that we propose for the V&V techniques.

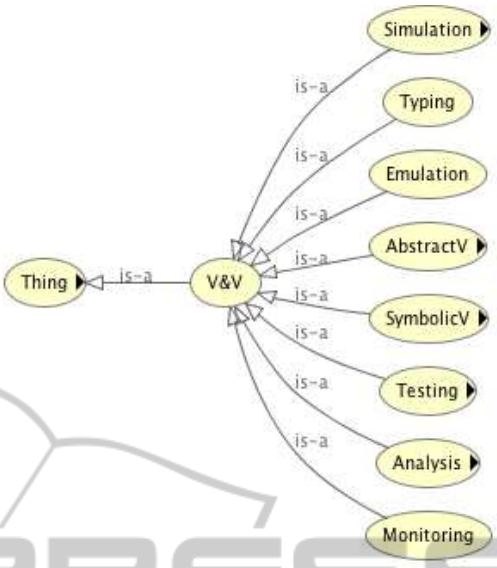

Figure 4: V&V techniques Hierarchy.

Property Concept. To describe a property, we can use several Property Description Languages (PDL). As an example we present the part of temporal logics that we use in the case study in Section 3. We can differentiate several temporal logics, the Linear Temporal Logic (LTL), Computational Tree Logic (CTL) and State Event-LTL (SE-LTL) are subclasses of temporal logic presented in Figure 5.

A SE-LTL formula (Chaki et al., 2004) can be associated to a Petri Nets illustrated in Figure 6 to express a verification property. This kind of property associated to Petri Nets can be verified using a model checking technique.

3 CASE STUDY

We instantiated the V&V concepts of the ontology with several V&V tools, like the TINA toolbox that

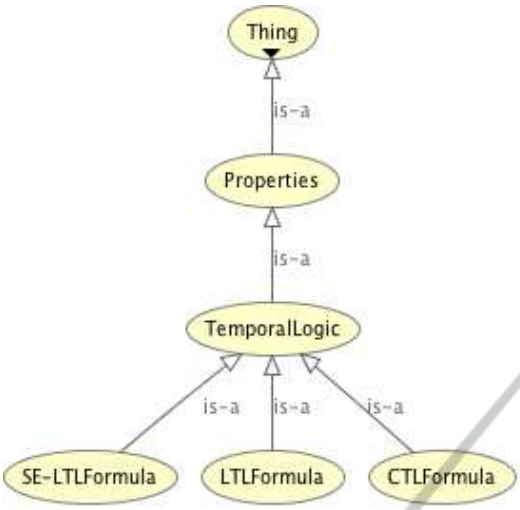

Figure 5: Temporal Logics Hierarchy.

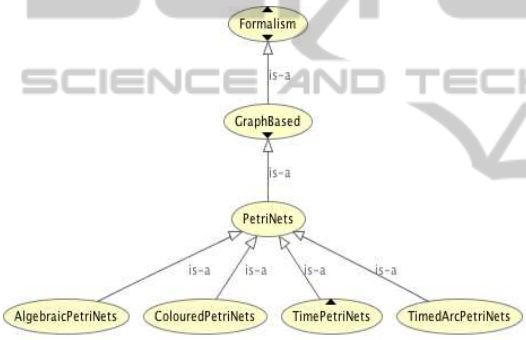

Figure 6: Petri Nets hierarchy.

we use in our case study. To elaborate this test, the DL Query[6] tab is used. We present the case for the verification of Petri Nets which are also known as place/transition nets. Petri Nets is one of the formalisms of our ontology, it is composed of several sub-classes: algebraic Petri nets, coloured Petri nets, time Petri nets, timed arc Petri nets and untimed Petri nets.

We can explore very easily our ontology, for example, if we want to verify a SE-LTL (State/Event LTL) formula on a time Petri nets system, our query and the result are expressed in Figure 7. The query states that the V&V technique must support the time Petri nets formalism and the SE-LTL formula and that a property on the time Petri nets formalism can be defined in the SE-LTL logic as presented in Figure 7. The result is TINA-Selt, it is the V&V tool specialised in the verification of SE-LTL formulas on time Petri nets systems.

[6]http://protegewiki.stanford.edu/wiki/DL_Query

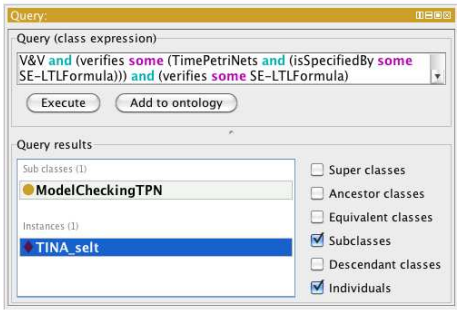

Figure 7: A query example.

4 RELATED WORK

By domain ontology we mean ontologies that exist in a specific domain of interest, it defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary. For instance, a number of domain ontologies are available on the Internet. But, to our knowledge, there is no ontology reported anywhere for the V&V domain. A Software Testing Ontology in UML for A Software Growth Environment of Web-Based Applications was proposed by Hong Zhu in (Huo et al., 2003). We have reported the terms in relation with the concept test of V&V techniques of the VVO presented in Figure 4.

5 CONCLUSIONS

We are working now on improving the VVO, the next step is to validate the ontology by linking existing with V&V tools and experimenting with various systems. We will submit the ontology to the system modeling and V&V community in order to gather as much feedbacks as possible. The ontology includes conceptual foundations for formalisms and V&V techniques, but is designed for the moment for knowledge sharing purposes.

REFERENCES

Alur, R., Courcoubetis, C., Henzinger, T., and Ho, P. (1993). Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. *Hybrid systems*, pages 209–229.

Bengtsson, J. and Yi, W. (2003). Timed automata: Semantics, algorithms and tools. *Lectures on Concurrency and Petri Nets*, pages 87–124.

Cervelle, J., Formenti, E., and Guillon, P. (2010). Ultimate Traces of Cellular Automata. *Arxiv preprint arXiv:1001.0251*.

Chaki, S., Clarke, E., Ouaknine, J., Sharygina, N., and Sinha, N. (2004). State/event-based software model checking. In *Integrated Formal Methods*, pages 128–147. Springer.

Clemens, S., Dominik, G., and Stephan, M. (1998). Component software: Beyond object-oriented programming.

D'Argenio, P. and Katoen, J. (2005). A theory of stochastic systems part I: Stochastic automata. *Information and computation*, 203(1):1–38.

Gomez-Perez, A., Fernández-López, M., and Corcho, O. (1991). Ontological engineering. *AI Magazine*, 36:56.

Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. *Towards very large knowledge bases: knowledge building and knowledge sharing*, 1(9):9.

Hilliard, R. (2000). Ieee-std-1471-2000 recommended practice for architectural description of software-intensive systems. *IEEE, http://standards. ieee. org*.

Huo, Q., Zhu, H., and Greenwood, S. (2003). A multi-agent software environment for testing Web-based applications. *COMPSAC-NEW YORK-*, pages 210–215.

Krishnan, S., Puri, A., and Brayton, R. (1994). Deterministic Ω automata vis-a-vis deterministic Buchi automata. *Algorithms and Computation*, pages 378–386.

Lawson, M. (2005). Finite automata. *Handbook of networked and embedded control systems*, pages 117–143.

Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific American*, 284(5):34–43.

McGuinness, D., Van Harmelen, F., et al. (2004). OWL web ontology language overview. *W3C recommendation*, 10:2004–03.

Mikk, E., Lakhnechi, Y., and Siegel, M. (1997). Hierarchical automata as model for statecharts. *Advances in Computing ScienceASIAN'97*, pages 181–196.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., et al. (1991). Enabling technology for knowledge sharing. *AI magazine*, 12(3):36.

Perrin, D. (2004). *Infinite words: automata, semigroups, logic and games*. Academic Press.