Integrating the Semantics of Events, Processes and Tasks across Requirements Engineering Layers

Maxime Bernaert (Supervisor: Geert Poels)

Department of Management Information Systems and Operations Management
Faculty of Economics and Business Administration,
Ghent University, Tweekerkenstraat 2, B-9000 Ghent, Belgium
Maxime.Bernaert@ugent.be

Abstract. Today, software should be more flexible, adaptable and more cost effective than ever before. There are indications that event-based architectures improve the flexibility, adaptability and cost effectiveness of software. Events are crucial concepts in event-based architectures, however, the concept of event has different interpretations in modeling techniques, which makes it difficult to integrate the use of different techniques during early and late requirements engineering. This paper outlines a PhD intended to develop an event-based requirements engineering methodology which supports the specification, development and verification of event-based systems. More specifically, this PhD strives to further develop the concept of event in requirements engineering and provide it with a formally defined semantics. The event concept is positioned with respect to existing concepts for modeling dynamic aspects of a system. A major goal is to keep the complexity of the modeling method at an acceptable level and enable a smooth transition of event-based architectures from requirements to implementation level. Finally, by performing an ontological analysis, using the BWW ontology and UFO, a set of orthogonal dimensions of the concept of event could be found.

Keywords: Requirements Engineering, Enterprise Ontology, Events, UFO, BWW

1 Introduction

Today, software should be more flexible, adaptable and more cost effective than ever before. Traditionally, interaction between components of a computer system is based on a mechanism of request/reply. Although the limitations of this interaction mechanism have been known for long, even the most recent technology platforms are still using request/reply as the basic principle for communication between a service provider and service consumer. Flexibility and adaptability of software can be significantly improved if an event-based architecture is applied [1] [2].

The power of event-based systems lies in the decoupling of sender and recipient which results in highly-flexible, adaptable and loosely-coupled systems. The majority of the applications of event-based systems is realized by using events at the implementation level, for example by means of event-based middleware technology. Although the basic components of an event-based system (publishers, subscribers, events, subscriptions, consumers and producers) are known in middleware technologies under different names (notifications, dispatchers, publications and broadcasters) with different semantics, existing taxonomies [3] [4] give a quite clear picture of existing definitions and interpretations of events at implementation level.

In the software development process, the implementation phase is generally preceded by a phase of requirements engineering and design. In requirements engineering two phases can be distinguished: an early requirements phase, in which the required functionality of the system is explored with respect to the objectives, and a late requirements phase, in which the specifications of a system are modeled in detail. Although these phases are crucial in developing the core requirements of the system, they are lacking a full integration of event-based systems. As an example, the definition of events and actions in UML 2.0 remains strongly implementation oriented and UML does not provide clear definitions of events across abstraction levels and requirement engineering layers.

The absence of the event-based architectural style in requirements engineering significantly hampers the implementation of these systems. In addition it becomes difficult to examine the correctness of the developed systems. Often it will be almost impossible to make the automatic translation of event-agnostic platform independent models to event-based platform specific models [5].

The aim of this PhD is the development of an event-based requirements engineering methodology which supports the specification, development and verification of event-based systems. More specifically, this PhD strives to further develop the concept of event in requirements engineering and provide it with a formally defined semantics. The event concept is hereby clearly positioned with respect to and is related to existing concepts for modeling dynamic aspects of a system. A major goal is to keep the complexity of the modeling method at an acceptable level [6] and enable a smooth transition of event-based architectures from requirements to implementation level.

Today, a smooth transition between a conceptual data model, towards the logical data model and the physical data models has been established. This PhD aims at developing a similar framework for the dynamic aspects of a system. The emphasis lies on early and late requirements and the transition between the phases. Verifying the feasibility of the transition to an implementation architecture will serve as a validation of the research results.

1. A first objective is to make an ontological analysis (using upper level ontologies, in particular UFO and BWW) of the concept of event and related concepts like activities, tasks and (sub-) processes in the different requirements engineering phases. The aim is to distinguish different dimensions regarding the concept of event, in which existing modeling techniques can be positioned with respect to their use of events. A systematic review will be performed, according to the procedures outlined in [7].

2. In a second part of this PhD, a framework and meta-model are developed for modeling dynamic and interaction aspects in an event-based matter. In this meta-model the concept of event is defined and its relationship with other concepts is

clarified. Existing modeling standards (such as UML and BPMN) are evaluated for their use of the concept of event. Particular attention goes to the support of the meta-model for quality aspects of event-based requirements. The meta-model must offer support for both a preliminary model (in which not all consistencies and quality requirements must be satisfied) and for a final model (which must satisfy all quality requirements).

3. In the final part of this PhD, the meta-model is extended with guidelines and methods to ease the use of this meta-model.

This paper is structured as follows. Section 2 starts with explaining the methodology, followed by section 3, briefly describing the scientific contribution and innovative aspects. Section 4 introduces ontological research, section 5 and 6 give a short description of UFO and BWW respectively, after which section 7 provides us with an overview of the results achieved so far. Section 8 stipulates the future research directions and the paper ends with a conclusion in section 9.

2 Methodology

2.1 Ontological Analysis

The first objective is a further in-depth analysis of how the concept of event is used in the different requirement engineering layers and how its semantics have (not) been defined. Through previous research, a good knowledge of the use and semantics of the concept of event in industry standards such as UML and BPMN has been developed. However, an additional study of philosophically-based ontologies for information and knowledge systems and their representations (e.g. Bunge-Wand-Weber [8] and UFO [9]) can provide additional insights.

2.2 Developing the Meta-model

A meta-model provides a definition of the concepts and their relationships which enables traceability between the different layers and models. Traceability is an important feature of the meta-model as aggregation and causal relationships are cornerstone concepts when modeling behavioral aspects. Activities at a higher level are realized by means of a number of activities at a lower level. In this respect, one can speak of aggregation (bottom-up) or vertical causality (top-down): an action at a higher level causes activities at the lower level. The other way round: a pattern of events at the lower level can lead to the conclusion that an event has happened at a higher layer.

Furthermore, the problem of vertical causality and aggregation is strongly related to the issue of granularity of software services, an issue for which there is no satisfying answer yet.

2.3 Making Models Executable

Modeling languages that are executable or can easily be translated to an executable format (such as the translation of BPMN to BPEL) have significant advantages over non-executable modeling languages. The meta-model will therefore be enriched with executable semantics by means of process algebra.

3 Scientific Contribution and Innovative Aspects

Powerful modeling languages with precise semantics are a basic requirement to leverage the promises of Model Driven Engineering. This requires however that requirement modeling languages build on the best practices of implementation technologies and architectures. The innovative contribution of the proposed research is at the level of requirements engineering by enriching existing modeling languages with the power of event-based software development.

The outcome of this research is a meta-model that defines and relates the semantics of events, actions, processes and other related concepts in an unambiguous way and a method to use this meta-model. In addition, a mapping to existing modeling languages and standards such as UML and BPMN is defined.

4 Ontological Analysis

One of the most-cited definitions of ontologies is: "An *ontology* is an explicit specification of a conceptualization" (199, [10]). Ontologies should be regarded as descriptive models [5], representing reality by a set of concepts, their interrelations, and constraints under open-world assumption [11], which states that anything not explicitly expressed by an ontology is unknown. Upper level ontologies, like UFO and BWW, provide basic concepts for classification and description. By looking at ontologies to make an analysis of the concept of event, we go back to the roots in principled philosophical theories about what kinds of things exist and what are their basic relationships with each other. We adopt the position that concepts should be founded on an upper level ontology referring to reality in a philosophically justified way. We assess goodness in terms of how well information systems embody the meaning of the real-world system they are intended to model [8], that is why it is crucial to look at the meaning of the concept of event in ontologies.

5 Unified Foundational Ontology (UFO)

In [9], a foundational ontology named UFO (Unified Foundational Ontology) has been developed, which can be used as a theoretically sound basis for evaluating and redesigning conceptual modeling languages in general, and ontology representation languages in particular. UFO is derived from a synthesis of two other foundational ontologies, GF0/GOL and OntoClean/DOLCE. The main purpose of UFO is to provide a foundation for conceptual modeling, including business modeling.

UFO addresses issues such as: the general notions of types and their instances; objects, their intrinsic properties and property-value spaces; the relation between identity and classification; distinctions among sorts of types and their admissible relations; distinctions among sorts of relational properties; part-whole relations.

UFO is divided into three incrementally layered compliance sets: UFO-A defines the core of UFO, excluding terms related to perdurants and terms related to the spheres of intentional and social things; UFO-B defines, as an increment to UFO-A, terms related to perdurants; UFO-C defines, as an increment to UFO-B, terms related to the spheres of intentional and social things, including linguistic things. In particular UFO-B discusses the meaning of the concept of event. UFO-C talks about intentional and social concepts.

6 Bunge-Wand-Weber (BWW)

The BWW ontology, a framework created by Wand and Weber [8] on the basis of the original metaphysical theory developed by Mario Bunge [12] [13], is developed to model information systems. They argue that they have relied on Bunge's ontology for three reasons [14]. First, they contend that Bunge's ontology is better developed and better formalized than any competing ontology they have encountered. Second, Bunge models the world as a world of systems. Bunge uses concepts that are fundamental to the computer science and information systems domains. Third, Wand and Weber argue they have been able to produce useful results using Bunge's model.

Having chosen Bunge's model as the basis for their work, Wand & Weber [8] argue the BWW model can be used to understand and predict the characteristics of good information systems grammars. Good is defined in a restricted way to indicate how well the scripts produced using the grammar convey the deep structure or meaning of the real-world system the information system is intended to represent.

By starting to look at ontologies to derive the meaning of the concept of event, we believe a good analysis can be made of existing meanings of the concept of event in requirements engineering. The position defended here is that, in order to model reality, the concepts used (like events) should be founded on upper level ontologies.

The purpose of the ontological model Wand and Weber have proposed is to define a set of constructs that are necessary and sufficient to describe the structure and behavior of the real world [8]. This set of constructs provides a benchmark to evaluate whether those grammars used to describe real-world systems are ontologically complete.

7 The Proposed Approach and the Results Achieved so far

A development process consists of different phases (conceptual modeling, design and implementation), which should fulfill different sets of requirements. This PhD is concerned with the meaning of the concept of event in conceptual modeling, in particular in the early and late phases in software engineering.

Conceptual modeling's main objective is concerned with identifying, analyzing and describing the essential concepts (for instance the concept of event) and constraints of a universe of discourse with the help of a modeling language that is based on a set of basic modeling concepts (forming a meta-model). Conceptual models therefore have to resemble the concepts of the real world in the most appropriate way, therefore our basis to evaluate different modeling languages will be the use of the concept of event in upper level ontologies, in particular UFO and BWW.

7.1 UFO

According to UFO, the concept of event can be classified according to three orthogonal dimensions: atomic event – complex event, instantaneous event – time-extended event, action event – non-action event.

7.1.1 Atomic Event – Complex Event

UFO-B makes a distinction between atomic events and complex events [15] [16].

An atomic event is an event that has no improper parts. Examples: an explosion, a message reception. A complex event is an event that is an aggregation of at least two events (that can themselves be atomic or complex). Examples: a parallel occurrence of two explosions, an absence of a message reception (without some time window).

A process is a complex event that is a sequence of two or more (possible parallel occurrences of) atomic events. Examples: a storm, a football game.

7.1.2 Instantaneous Event – Time-Extended Event

Does the event take place without duration or is there a time duration involved?

In [16], the authors state that it is important to emphasize that being atomic and being instantaneous are orthogonal notions in this framework, i.e., atomic events can be time-extended as well as an instantaneous event can be composed of multiple (instantaneous) events.

7.1.3 Action Event – Non-action Event

In UFO-C, a distinction is made between action events and non-action events [15].

An action event is an event that is created through the action of a physical agent. A non-action event is an event that is not created through an action of a physical agent.

A physical agent is a physical object that creates action events affecting other physical objects, that perceives events, possibly created by other physical agents, and to which we can ascribe a mental state. Examples: a dog, a human, a robot.

Action events are intentional events [16], i.e., events which instantiate a plan with the specific purpose of satisfying some intention.

7.2 BWW

BWW describes three dimensions of the concept of event: event – process, internal event – external event, well-defined event – poorly-defined event.

7.2.1 Event – Process

A BWW-event is “A change of state of a thing. It is effected via a transformation” [8].

A BWW-process is “An intrinsically ordered sequence of events on, or states of, a thing” [17]. “Processes are either chains or trees of events” [12].

7.2.2 Internal Event – External Event

A BWW-internal event is “An event that arises in a thing, subsystem or system by virtue of lawful transformations in the thing, subsystem or system. The before-state of an internal event is always unstable. The after-state may be stable or unstable” [8].

A BWW-external event is “An event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment of the thing, subsystem or system. The before-state of an external event is always stable. The after-state may be stable or unstable” [8].

A BWW-stable state is “A state in which a thing, subsystem or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event)” [8].

A BWW-unstable state is “A state that will be changed into another state by virtue of the action of transformation in the system” [8].

7.2.3 Well-defined Event – Poorly-defined Event

A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known [8]. A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known [8].

8 Future Research

Starting from the different dimensions of the concept of event found in UFO and BWW as upper level ontologies, a first step is set to investigate modeling techniques, for example BPMN, BPDM, UML, and their use of the concept of event. By performing a systematic review with a clearly defined protocol, a list of modeling techniques will be distilled in which the concept of event is used. Our intention is to make a mapping of the different existing modeling techniques, according to their use of the concept of event, using the different dimensions found in UFO and BWW. This ontological mapping will give us a clear view of the most frequently used meaning of the concept of event and the least frequently used meaning of the concept of event.

This ontological analysis will be a good starting point to start developing an ontologically well-based meta-model, where the concept of event is used in all its possible meanings. The meta-model can be developed starting from existing meta-models, or can be developed from scratch, if no sufficient basis of meta-models is available.

This meta-model will be supported by a method, which provides us with guidelines on how to interpret the meta-model and develop models.

Complementary to an ontological analysis, a study starting from existing modeling techniques can distill the meaning of the concept of event as used in these modeling techniques. Comparing the ontological analysis with this research can provide us with interesting new insights.

9 Conclusion

The concept of event has several interpretations. By making an ontological analysis, based on UFO and BWW, two sets of each three orthogonal dimensions have been discovered on which existing modeling techniques can be mapped according to their use of the concept of event.

UFO makes a distinction between atomic and complex events, instantaneous and time-extended events and action events and non-action events.

BWW's dimension of event and process resembles the atomic-complex dimension of UFO. The two other dimensions, internal event – external event and well-defined event – poorly-defined event, are two other dimensions.

By projecting the used event semantics onto these ontological dimensions, it becomes easier to compare different modeling techniques. If we want to make a meta-model, we need to have a clear understanding of the concepts used in the existing models.

References

1. Mühl, G., Fiege, L., Pietzuch, P.: Distributed Event-Based Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
2. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Rule Representation, Interchange and Reasoning on the Web (2008) 3-3
3. Barrett, D.J., Clarke, L.A., Tarr, P.L., Wise, A.E.: A framework for event-based software integration ACM Transactions on Software Engineering and Methodology Vol. 5 (Oct. 1996) pp. 378-421
4. Meier, R., Cahill, V.: Taxonomy of Distributed Event-Based Programming Systems. The Computer Journal 48 (2005) 602-626
5. Aßmann, U., Zschaler, S., Wagner, G.: Ontologies, Meta-models, and the Model-Driven Paradigm. Ontologies for Software Engineering and Software Technology (2006) 249-273
6. Siau, K., Erickson, J., Lee, L.Y.: Theoretical vs. Practical Complexity: The Case of UML Journal of Database Management Vol. 16 (July-Sept. 2005) pp. 40-57
7. Kitchenham, B.: Procedures for performing systematic reviews. Technical Report TR/SE-0401 (2004)
8. Wand, Y., Weber, R.: Towards a theory of the deep structure of information systems. In DeGross, J., Alavi, M. & Oppelland, H. (eds), Proceedings of the Eleventh International Conference on Information Systems, Copenhagen (1990b) pp. 61-71

9. Guizzardi, G.: Ontological foundations for structural conceptual models. Enschede (2005) 416
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5 (1993) 199-220
11. Horrocks, I., Schneider, P., van Harmelen, F.: From SHIQ and {RDF} to {OWL}: The making of a web ontology language. Journal of Web Semantics 1 (2003) 7-26
12. Bunge, M.: Treatise on Basic Philosophy, Vol. 3, Ontology I: The Furniture of the World. Reidel, Boston (1977)
13. Bunge, M.: Treatise on Basic Philosophy: Volume 4: Ontology II: A World of Systems. Reidel, Boston (1979)
14. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information Systems Journal 3 (1993) 217-237
15. Guizzardi, G., Wagner, G.: Some applications of a Unified Foundational Ontology in Business Modeling. In: Green, P., Rosemann, M. (eds.): Business Systems Analysis with Ontologies. Idea Group Publishing (2005)
16. Guizzardi, G., Falbo, R.A., Guizzardi, S.S.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS2008), Recife (2008)
17. Green, P.F.: An Ontological Analysis of Information Systems Analysis and Design (ISAD) Grammars in Upper CASE tools. Department of Commerce. University of Queensland (1996)