

PH. D. THESIS ABSTRACT

Animation of OASIS Specifications by Means of Object Oriented Petri Nets

Graduated: Pedro Sánchez Palma

Universidad Politécnica de Cartagena

Campus Muralla del Mar s/n 30202

Cartagena, España

e-mail: Pedro.Sanchez@upct.es

Advisor: Isidro Ramos Salavert

Universidad Politécnica de Valencia, España

e-mail: iramos@dsic.upv.es

Abstract

The validation by means of animation permits the systematic development of clear, concise, precise, and unambiguous software specifications, present aspects in the formal methods of software engineering but absent in traditional prototyping. In addition, the quality of the conceptual models depends to a great extent on the integration degree of the users in the development process. This paper describes the objectives and goals obtained in this software engineering line of work and as a result a doctoral thesis work.

Keywords: Requirements Engineering, Validation by Animation, Formal Methods.

1 Introduction

Hardware and software systems change continuously in an inevitable manner, in size as well as in functionality. Due to the increase in complexity, the possibility of committing errors also increases. A great part of these errors assume an economic cost in time, and even in human lives.

One of the principal objectives of Software Engineering is the power to facilitate to the developers the construction of systems that operate with a certain degree of trust, independently of the associated complexity. A form of reaching

this objective is using formal methods in language form, techniques and tools with mathematical foundations that allow the specification as well as the verification of such systems. A priori, the use of formal methods does not guarantee the correction although it can increase the system knowledge a considerable amount when detecting inconsistencies, incompleteness, ambiguity, etc.

Engineering Requirements (Wieringa, 1996) have the objective of obtaining a specification of the system requirements desired to be built at a certain abstraction level and fulfilling certain properties. Thus, the conceptual model represents a connection between the problem space and the solution space to express the functional requirements of the system, making use of a specification language. In this manner, the conceptual model constitutes the initial base for the tasks that follow in the development of a system.

Traditionally, the requirement specifications have been expressed in semiformal notations, although, more often they are written using formal specification languages. The formal specifications possess certain additional advantages as a result of having a well defined syntax and semantics. A formal specification allows to specify all the details minimizing or eliminating the omissions, ambiguities or inconsistencies in the specification. In addition, it is possible to reason about the properties of a formal specification, in particular, to verify it and to validate it with respect to the initial requirements.

If in addition the specification language is feasible then it obtains an additional advantage: the specification not only represents the conceptual model but also the software execution model that will be implemented.

The traditional development of software presents the problem of which the final implementation system is the first precise version of the same that can be executed. In this manner, it is only possible to validate the system when the design decisions have already been taken, which implies a great amount of implementation aspects. As a consequence, when errors are detected, the design needs to be remade.

If the specifications are feasible then the validation in a high level of abstraction is possible, increasing the correction and trust in the software and considerably reducing the development times and costs. With the validation in the first phases of the software development life cycle, not only is the detection of errors obtained at an early stage but those requirements that are not clear can be made specific and completed from the interaction with the specification execution.

The construction of the conceptual model is a discovery process for the analyst as well as for the user. The idea is to construct the conceptual model in an iterative form, with successive refinements and the participation of the user. In each iteration, the analyst verifies the model and validates it with the user. Once the agreement between the parties involved in the process has been reached, the conceptual model is used as a contract and entrance to the phases that follow in the development. In this manner, the associated cost is reduced to later modifications or to partially specified requirements.

When the specification language used is not directly executable, the model validation requires an additional effort. Therefore, the validation of the requirement specifications by animation of these and using prototyping techniques is the departure point and the motivation of this paper.

2 Work Frame

This thesis paper (Sánchez Palma, 2000) was generated in the "Object Oriented Conceptual Modeling" research group, pertaining to the Information and Computing Systems Department of the Polytechnic University of Valencia. One of the principal activities carried out in the last years by this group has been the development of OASIS (Open and Active Specification of Information Systems) (Pastor, 1995, Letelier, 1998), a formal model for the construction of conceptual models that follows the object oriented paradigm. From the OASIS perspective, an information system is a society of autonomous

and concurrent objects that interact between each other. Within this line of thinking, the intention of this thesis is to analyze and to make the automatic generation of code to animate OASIS specifications. Petri nets have been utilized as formalism destiny in this generation, given the good properties of the same as far as the simulation possibility, the property demonstration and the tool availability.

The following two projects, "Comisión Interministerial de Ciencia y Tecnología (CICYT): OASIS (1994-1997, TIC94-0557) and MENHIR (Participantes en el Proyecto Menhir, 1998) (1997-2000, TIC97-0593-C05-01), have made possible the attainment of this thesis.

3 Objectives

The objective of this work has been to establish the necessary theoretical and practical bases for the automatic validation of OASIS specifications making use of the Petri nets. In a more specific form, the following objectives have been obtained:

- * Extension and completion of the OASIS language. The associated work has been framed within the 3:0 version language development (Letelier, 1998). The departure point was the 2:2 version (Pastor, 1995) where diverse modifications and amplifications took place. Some of the obtained improvements are: the uniform use of the Dynamic Logic to express the properties of one OASIS class; the incorporation of the client perspective; the redefinition of the process concept (including operations and protocols) and the enrichment of the used mechanisms to define complex classes (adding and specialization).
- * Definition of an execution abstract model for the OASIS language that constitutes the operational semantic of the same.
- * Establishment of the necessary translation guidelines to pass from OASIS to Petri nets. A translation program that automatically generates an equivalent Petri net parting from an OASIS specification has been implemented.
- * Construction of an initial version of an environment for the automatic animation of OASIS specifications framed within the engineering process requirements that include the animation of conceptual models in an automatic form and consider the scenario use.

If we study the existing solutions we can observe the following deficiencies:

1. The functionality and integration of the developed modules for validation is scarce. In most of the cases it deals with preliminary versions.
2. The operational semantic (execution model) of different environments is not provided in any case.
3. The implementation is made in imperative languages (the C++ language is usually habitual in these cases) with which the proof that the implementation is faithful to the conceptual model semantics to begin with seems difficult.
4. Plans do not take place as a result of the animation with which the user must deduce the interactions had between the objects.
3. The interaction with the animator is made through scenery techniques which enables communication between user-analyst. In the previously mentioned works, the result of the animation presents for each object, which takes an additional effort at the sight of having to fix the global interaction between all the participating.
4. The Petri nets are a formalism that enable the equivalency demonstration between the specification and the corresponding implementation.
5. It is possible to partition a problem in two parts: one oriented to the conceptual modeling with OASIS and the other represented making direct use of Petri nets. In this manner, the concurrency aspects that are modeled more easily with Petri nets can be represented directly.

The work made on the animation of OASIS specifications situates it advantageously, with respect to the previous jobs, for the following reasons:

1. OASIS is associated with a declarative semantic based on Dynamic Logic (Meyer, 1998) and an operational semantic that constitutes the language execution model. For all the OASIS language concepts a translation procedure is presented to this logic (for the exception of synchronous transactions, adding and communication). Tasks like automatic verification, validation and generation of code are favored, in consequence, given the formal frame that integrates them.
2. The animation is given by the execution of a concurrent implementation (in Petri nets) of the OASIS model semantic, avoiding the necessity of artificial monitors in charge of distributing the system execution thread.
6. The implemented translator has as a departure point the set of formulas in Dynamic Logic of a specification. This makes possible the future evolution of OASIS (in its syntax and/or associated semantics) minimizing or eliminating the necessity of modifying the code of this translator. That is, the Dynamic Logic appears as a reference point for future possible implementations.

4 Work Summary

The work frame is summarized in the following points:

Model. OASIS is the approach considered to represent the conceptual model corresponding to the requirements of the user.

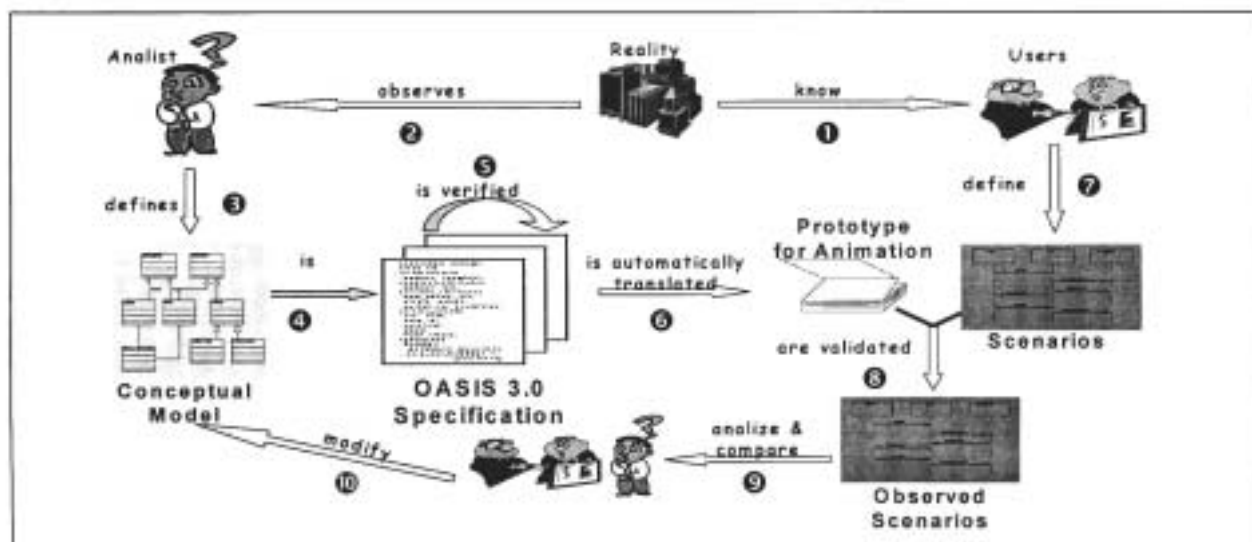


Figure 1. Conceptual Modeling Based on OASIS.

Animation. A prototype oriented to the validation of specified functional requirements using the OASIS language is obtained in an automatic form. The implementation is made using Petri nets and the animation environment chosen for these nets is the CODESIGN tool¹.

Scenarios. The use of scenarios is used to represent part of the user requirements and to show the obtained plans after the specification animation.

The quality of a conceptual model depends to a great extent on the degree of integration of the user in the process associated in the capture of the requirements. Figure 1 shows the process model considered for OASIS when the raised approach is followed. The departure point is the users who know the reality (the system) that is desired to model. The task of the analyst is to capture the requirements that allows to obtain the system conceptual model (probably with the user intervention). This conceptual model is represented in some graphical annotation (for example, UML for those aspects of the annotation that has a clear correspondence with the OASIS concepts). A translation process takes of the graphical annotation to the OASIS specification for the modeled system.

The different commented tasks of verification in the previous section are applicable in this point. When the OASIS specification is free of syntactic and/or semantic errors then it proceeds to obtain automatically the prototype for animation. Previously, the users define a set of scenarios (case studies) that they consider representative and that describe partially or totally the system behavior. The prototype execution (using the scenarios as input data) gives rise to results represented in the same annotation.

Users and analysts study the obtained differences which give rise to possible modifications in the departure conceptual model. The cycle continues until the conformity of the users is obtained, verifying and validating the conceptual model. The following considerations contribute a greater detail to the process presented:

- * To simplify the Figure, the scenarios that can also be an input point to obtain the conceptual model have not been represented.
- * The scenarios made by the users can present partial system information that later appears, as a result of the animation, in an interrelated form. This aspect justifies, partly, the presence of the users in the validation process. If the analyst only hopes to obtain the scenarios that the users have made then these users are dispensable in the process. In other words, there can be animation results presented as errors indicators when in fact they are particular situations that the users did not reflect in the scenarios or represented in partial form.

- * The scenarios must include some state representation that facilitates to the analyst the comparison process. To simplify, in each life passage of each object, the old and new value for each modified attribute can be demonstrated.
- * The validation can begin with a subset of the total model. Even more, it is possible to partition the model in such that prototypes for each one of the calculated partitions are executed. It is obvious that the only condition that prevails is the model consistency from which the prototype will be generated.
- * The prototype is constructed in a concurrent implementation environment which permits a more natural correspondence of the OASIS concepts with the prototype elements. When the environment is not concurrent, a monitor that serializes the inherent parallelism to an OASIS specification is necessary.

5 Conclusions

This work supports the argument of which the formal specifications give support to the process of the specification of requirements, allowing the gathered properties to be examined within a precise work frame based on validation by means of animation.

In this manner the strong points of the formal methods and prototyping are emphasized, at the same time that the classical problems are mitigated.

We can summarize the contributions in the following items:

1. The OASIS declarative semantic has been developed, incorporating the passage notion as a consistent set of actions carried out concurrently that determine the state change.
2. An OASIS execution model (Letelier, 1998) that is the reference point for the animation of OASIS object societies has been established. This process model is concurrent differentiating itself from other industrial proposals with sequential and monitored direction.
3. A correspondence between the concepts of class OASIS and Dynamic Logic used has been established (Meyer, 1998). The evaluations, preconditions, firings, protocols, operations, derived attributes and integrity restrictions (static and dynamic) from OASIS have been considered. Thus doing this, it is possible to integrate in a homogeneous frame the different language aspects and to provide a way for the demonstration of properties.
4. The theoretical foundations have been established to incorporate in the Dynamic Logic translation process the specification relations between classes (static and dynamic partitions).

¹ <http://www.tik.ee.ethz.ch/~naedele/moses/Distribution250.html>

5. An architecture for the concept implementation included in an OASIS class has been defined. With this approach, an OASIS specification (seen as Dynamic Logic formulas) translator to Petri nets has been constructed, with which the execution of these specifications is obtained. The prototype obtained is oriented to validation for which those non functional requirements or better yet relative to the interface with the user are outside of the animation process.
6. An environment proposal for the capture and validation of requirements in incremental form has been presented. This validation is supported by the execution of the Petri nets corresponding to an OASIS specification. The interaction with the users is based on scenarios expressed as Sequence Diagrams. This is the mechanism that allows users to validate that the "correct product" is being constructed.

6 Future Work

The work that is a continuation of the present has double intention. From a theoretical point of view, the following are susceptible aspects for future consideration:

1. Extend the execution model (Letelier, 1999) to incorporate the synchronous communication.
2. Justify the equivalence between the OASIS model and the Petri nets implementation. This work is not trivial given the dimensions of the obtainable graph associated with the nets obtained in the translation and the conceptual distance that exists between the presented formalisms. In (Sánchez, 1997), a path is outlined to demonstrate this aspect (a Transformation Function that establishes equivalence between the state representation and the net framing is needed).

From the an applied point of view, the following tasks arise:

1. Integrate the translators implemented in a graphical specification environment for OASIS.
2. Construct a translator of the different repositories of the OASIS language (in each one of its versions) to Dynamic Logic. Thus, the animator can be useful for any version.
3. Construct a module of scenario comparisons that automates the detection of differences between the predicted thing and the resultant.

4. Modify the Dynamic Logic translator to Petri nets so that it incorporates the derivation formulas and integrity restrictions (static and dynamic).
5. Connect the CODESIGN application with some external simulator so that it can perform the validation in remote form and can collect statistics of firings, framings, etc.

References

- Letelier P., Ramos I., Sánchez P., Pastor O.** OASIS versión 3.0: Un Enfoque Formal para el Modelado Conceptual Orientado a Objeto. Servicio de Publicaciones de la Universidad Politécnica de Valencia, SPUPV-98.4011, 1998. www.dsic.upv.es/users/oom/oasis.html
- Letelier P., Sánchez P., Ramos I.** Un Modelo Abstracto de Ejecución para Animar Modelos Conceptuales. VI Congreso Internacional de Investigación en Ciencias Computacionales, páginas 80-91, Cancún México, Septiembre de 1999.
- Participantes del Proyecto Menhir.** MENHIR: Modelos, Entornos y Nuevas Herramientas para la Ingeniería de Requisitos. Actas de las III Jornadas de Investigación y Docencia en Bases de Datos, JIDBD'98, Valencia, 1998.
- Meyer J.-J.Ch.** A Different Approach to Deontic Logic: Deontic Logic viewed as a Variant of Dynamic Logic. In Notre Dame Journal of Formal Logic, vol. 29, pages 109-136, 1988.
- Pastor O., Ramos I.** OASIS versión 2 (2.2): A Class-Definition Language to Model Information Systems Using an Object-Oriented Approach. Servicio de Publicaciones Universidad Politécnica de Valencia, SPUPV-95.788, 1995.
- Sánchez Palma, Pedro.** Animación de Especificaciones OASIS mediante redes de Petri Orientadas a Objeto. Tesis Doctoral. Director: Isidro Ramos Salavert, Universidad Politécnica de Valencia. Publicación digital en CD con ISBN 84-699-2899-6. Año 2000. www.dsic.upv.es/users/oom/docs/tesis-ppalma.pdf
- Sánchez P., Letelier P., Ramos I.** Representación de Objetos en Teorías y su Traducción a un Modelo de Concurrencia. Revista Computación y Sistemas, vol. 1, No. 2, páginas 101-111, 1997.
- Wieringa R.J.** Requirements Engineering: Frameworks for understanding. John Wiley & Sons, 1996.

