

## WAFOCA 2006 Abstracts

Title On the Use of QVT for Modeling Viewpoint Correspondences

Authors José R. Romero, Nathalie Moreno, Francisco Durán, and Antonio Vallecillo

Abstract Viewpoint modeling is currently seen as an effective technique for specifying complex software systems. However, having a set of independent viewpoints on a system is not enough. These viewpoints should be related, and these relationships made explicit in order to get the complete and consistent specifications of the system. In this paper we explore the use of MOF QVT for specifying viewpoint correspondences, in the particular case of RM-ODP. RM-ODP is an architectural framework that defines five complementary viewpoints for the specification of open distributed systems, and establishes correspondences between viewpoint elements. In addition to presenting how ODP correspondences can be represented in QVT, the paper focuses on the issues that we have found when modeling these correspondences with OCL and QVT relations. Among other things, these issues show the need to count with a formal foundations for OCL and QVT in order to be able to reason about the correspondences. On the Use of QVT for Modeling Viewpoint Correspondences return to WAFOCA06

Title Extending OCL for Security Requirements and Traceability

Authors Fernando Molina Molina, Francisco Javier Lucas Martínez, Eduardo Fernández-Medina, Mario Piattini<sup>2</sup> and Ambrosio Toval Álvarez

Abstract The Model Driven Architecture (MDA) proposal, promoted by OMG, has made an important impact in recent years. MDA proposes that the models, in their different abstraction levels, be used as a mechanism to carry out the whole software development process, from the construction of the computation and platform independent models (CIM and PIM) through to platform specific models (PSM) and automatic code generation from the same. One important feature that must be taken into account in the development of systems is related to security features. Information security is a serious requirement which must be carefully considered, not as an isolated aspect, but as an element present in all stages of the development lifecycle. This position paper proposes to use a variation of OCL as language that permit to add security constraints to all the stages of the MDA development cycle, from the requirements of the system to \_nal code. With this purpose, security constraints will be included in CIM models and will be automatically transformed in OCL expressions that will complete the di\_erent models used in the development cycle. return to WAFOCA06

Title OCL2: Using OCL in the Formal Definition of OCL Expression Measures

Authors Luis Reynoso, Marcela Genero and Mario Piattini

Abstract Within the Object Oriented software measurement a lot of measures have proliferated during the last decades. However, if the measure definition is not precise enough, for instance when natural language is used, misinterpretations and misunderstanding of their intent can be introduced. Therefore, this situation may flaw the interpretation of experimental findings or even can difficult in building adequate measures extraction tools. This paper carefully describes how a set of measures for OCL expressions were precisely defined upon the Object Constraint Language (OCL) Metamodel. Moreover, a precise definition was reached avoiding underspecification, and this was obtained due to the fact we use in the measure definition the OCL Language. So, we used OCL twice (OCL2): as a language for defining measures and as a target to capture its core concepts through measures. return to WAFOCA06

Title Definition of OCL 2.0 Operational Semantics by means of a Parameterized Algebraic Specification

Authors Artur Boronat, Isidro Ramos, José Á. Carsí

Abstract In Model-Driven Engineering, software artefacts are represented by means of models that can be manipulated. Such manipulations can be performed by means of transformations and queries. The forthcoming standard Query/Views/Transformations and the standard language OCL are becoming suitable languages for these purposes. This paper presents an algebraic specification of the operational semantics of part of the OCL 2.0 standard, focusing on queries. This specification is used in a model management tool that provides support for model-driven tasks within the

Eclipse platform. This algebraic specification of OCL can be used in the Eclipse Modeling Framework to represent models in an algebraic setting and to perform queries or transformations over software artefacts that can be represented as models: model instances, models, metamodels, etc. An interesting feature of the specification is the simulation of higher-order functions for the sake of the reuse of collection operators. return to WAFOCA06

Title Using reflection to implement a rewriting-based validation tool for UML+OCL class diagrams in Maude

Authors Manuel Clavel and Marina Egea

Abstract In this paper we explain the reflective design of the ITP/OCL tool, a rewriting-based tool that supports automatic validation of UML class diagrams with respect to OCL constraints. From a conceptual point of view, the ITP/OCL tool is directly based on our algebraic semantics for UML+OCL class diagrams according to which: i) class and object diagrams are specified as membership equational theories; ii) invariants are represented as Boolean terms over extensions of those theories; and iii) validating object diagrams with respect to invariants is reduced to checking whether the corresponding Boolean terms rewrite to true or false. From an implementation point of view, the ITP/OCL tool is written entirely in Maude, making extensive use of its reflective capabilities to implement the user interface, thanks to which the tool's underlying algebraic semantics remains hidden to the user who only must be familiar with the standard notions of UML diagrams and OCL invariants. return to WAFOCA06

Title Graph Transformation vs. OCL for View Definition

Authors Esther Guerra and Juan de Lara

Abstract Multi-View Visual Languages [3] (MVVLs) are made of several diagram types that can be used for the specification of different aspects of a system. A prominent example is the UML, which defines several diagram types for the description of the static and dynamic aspects of a software application. In our approach, we define the MVVL syntax by means of meta-modelling. Thus, we define a meta-model for the whole MVVL, and additional ones for each diagram type, which are restrictions of the whole MVVL meta-model. From this highlevel description, the AToM3 tool [5] is able to generate a customized modelling environment for the defined MVVL. In this environment, the user is able to build models, which are instances of the different diagram types. We call such models push-views. The push-views are glued together in a single model: the repository. The gluing is performed by automatically generated triple graph grammar rules, and is a means to perform consistency checkings between the push-views. The generated triple graph grammar rules are activated in a similar way as the model-view-controller pattern. In this way, when the user modifies one view, the changes are propagated by the grammar to the repository, and from there to the rest of the views. In addition, it is possible to define pull-views (or derived views), which extract information from a base model (another view or the repository). We propose defining such pull-views by means of a special kind of graph patterns that we call reverse patterns, as the morphism is sought from the model to the pattern. From these reverse patterns a number of triple graph grammar rules can be derived. The execution of the rules in the base model generates the derived view, together with a mapping that allows a bidirectional update of the view and the base model. Finally, as a pull-view can be thought of being generated by a special kind of query, we compare OCL and reverse patterns for expressing such queries. return to WAFOCA06