INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team AtlanMod

Modeling Technologies for Software Production, Operation, and Evolution

Rennes - Bretagne-Atlantique

Theme : Distributed Systems and Services

Activity Report

2009

Table of contents

1. Team

Faculty Member

Jean Bézivin [Team leader, Professor, HdR]
Frédéric Jouault [INRIA engineer until March (on EDONA), EMN faculty member after]

Technical Staff

Hugo Brunelière [Engineer, Modelplex]
Guillaume Doux [Engineer, IDM++, Université de Nantes]
Jean-Sébastien Sottet [Engineer, Lambda]
Cauê Clasen [Engineer from November 2009, internship student between February and July 2009]

PhD Student

Kelly Garcés [Collaboration with Ascola, ARMINES]

Post-Doctoral Fellow

Benoît Combemale [until June 2009]
Mathias Kleiner [until August 2009]
Massimo Tisi [from November 2009, engineer in January and February 2009]

Visiting Scientist

Andrés Vignaga [Ph.D. student from University of Chile (June 2009)]
Daniel Perovich [Ph.D. student from University of Chile (February 2009)]

Administrative Assistant

Diana Gaudin [Part-time]

Other

Kling Wolfgang [internship student, from August 2009]

2. Overall Objectives

2.1. Presentation

Model Driven Engineering (MDE) is an emerging research area that considers the main software artifacts as typed graphs. This comes from an industrial need to have a regular and homogeneous organization where different facets of a software system may be easily separated or combined. The basic assumption of MDE is that the classical programming code is often not at the right representation level for managing all these facets even if, at some point of the process, executable code will usually be generated from some abstract representation level.

2.2. Previous Achievements

The team has progressively developed a conceptual modeling framework [39] [9] that is now well accepted in the scientific community. This provides a high-level view of MDE described in several research papers (e.g., [4] [5] [41] [43] [3]). The iterative definition of this conceptual framework has been validated by the construction of several experimental concrete toolboxes. The first generation (sNets, based on semantic networks) was built in Smalltalk in the 90's [37]. The second one is the current AmmA (for AtlanMod Model Management Architecture) toolbox available in Eclipse. This is at the same time a research platform and an industrial toolbox. The AtlanMod team is working to deliver a new version of this toolbox more compact, more efficient, more usable, and more portable. The basic principles will be the same, but many simplifications and improvements will allow coping with more general situations.

One important result that was obtained in the previous period is the fact that MDE is a branch of language engineering. This was one of the outcomes of a Dagstuhl school that we co-organized in 2004 [38]. More precisely a metamodel is now usually considered as the definition of the abstract syntax of a language, usually a Domain Specific Language (DSL). This idea is central in the AmmA toolbox that may be viewed as a DSL building framework composed itself of a number of primitive DSLs. The four most known AmmA DSLs are KM3 (a DSL for metamodel specification), TCS (a DSL for textual syntaxes), ATL (a DSL for model transformation), and AMW (a DSL for representing model correspondences). All these mutually dependent tools are available under *Eclipse.org* (projects or components: M2M, ATL, TMF, MoDisco, AM3, AMW). They are currently in use in research, teaching, and industry and they have a broad user community.

2.3. Highlights of the Year

Year 2009 has seen the consolidification of the scientific and technical approaches proposed by the AtlanMod team. This is notably visible in the three following points:

- The choice to base most of the operations on models on rule-based and metamodel-based declarative expressions has become mainstream. We published the first paper proposing this in 1998 at the EDOC conference in San Diego [10]. Today this idea has gone mainstream and most of the competing teams in the domain are following similar graph-based approaches. The advantages of using rule-based declarative formalisms is accepted not only in academic research, but in industry as well. This approach, proposed in the original paper by R. Lemesle, has been shown to be the best one for facilitating the maintenance of modeling applications. The partnership with the OBEO company has allowed the long research process on transformation languages to converge to industrialization and broad dissemination. As a consequence, the AtlanMod team may now refocus on the ATL-Research initiative.
- A second option that was taken by the team is to consider a transformation language to be defined by a metamodel, at level M2 of the metamodeling stack. This is also well accepted and many advantages are becoming apparent. In particular it is now clear that this choice allows to easily define and apply Higher-Order Transformations (HOTs), i.e. transformations taking other transformations as input and/or producing other transformations as output. AtlanMod has pioneered the use of HOTs in model engineering and this practice is now well accepted [26].
- The success of the model transformation paradigm has allowed many successful applications, and in turn generated a new set of constraints on the tools. For example the size of models and metamodels was previously limited when the main application was generating code from human-generated models like UML models. Now that we are moving towards model-driven reverse engineering and model-driven interoperability, we face huge or at least voluminous models that we have to process. One important activity of AtlanMod in 2009 has been to start investigating how the model-driven engineering do scale up in face of new constraints in term of size, of needed performance, of variability, etc. In the Lambda project, the team has lead a study on the identification of the dimensions of complexity in MDE. Additionally, AtlanMod contributed a case study [33], as well as an accompanying solution [32], on model transformation scalability to the GraBaTs'2009 workshop.

3. Scientific Foundations

3.1. Genesis

In the 80's the dominant scheme was object technology leading to class-based languages. The main concepts were classes and instances with the associated instantiation and inheritance relations. This produced an efficient implementation technology that has become mainstream today with such languages as Java and C#. The main associated paradigm is object composition that has progressively replaced procedural refinement used in previous technologies. Several extensions to object technology have proposed to use various component-based architecture schemes.

As an alternative to these proposals, MDE offers multiple schemes, not only object-based but also relation-based, event-based, rule-based, function-based, etc. Each of the schemes may be represented by a given metamodel. Correspondences between them may be explicitly stated, for example by a class to relational transformation. MDE can thus be seen as a generalization and abstraction of object technology allowing to map more abstract organizations on class-based implementations. This is the essence of the EMF (Eclipse Modeling Framework). The AmmA platform generalizes the EMF principles by proposing a more regular model-based organization. Similarly to EMF, AmmA may be efficiently implemented on top of Java or C#-based platforms.

In MDE, models are considered as the unifying concept [39]. Traditionally, models have often been used as initial design sketches mainly aimed for communicating ideas among developers. On the contrary MDE promotes models to primary and precise artifacts that drive the whole development process. The notion of model goes beyond the narrow view of semi-formal diagram thus requiring much more precise definitions and implementations that allow partial or full automation.

The MDE community has been using the concepts of terminal model, metamodel, and metametamodel for quite some time [12]. A terminal model is a representation of a system. It captures some characteristics of the system and provides knowledge about it. In MDE we are interested in terminal models expressed in precise languages. The abstract syntax of a modeling language, when expressed as a model, is called a metamodel. A language definition is given by an abstract syntax (a metamodel), one or more concrete syntaxes, a definition of its semantics, etc. The relation between a model expressed in a language and the metamodel of this language is the conformance relation (*conformsTo*). This should not be confused with the representation relation (*repOf*) holding between a terminal model and the system it represents. Metamodels are in turn expressed in a modeling language called metamodeling language. Its conceptual foundation is captured in a model called metametamodel. Terminal models, metamodels, and metametamodel form a three-level architecture with levels respectively named M1, M2, and M3. A formal definition of these concepts is provided in [4] and [40]. MDE promotes unification by models like object technology proposed in the eighties unification by objects [36]. The principles of MDE may be implemented in several standards. For example, OMG proposes a standard metametamodel called Meta Object Facility (MOF). An example of a metamodel in the context of OMG standards is the UML metamodel.

In our view the main way to automate MDE is by providing transformation facilities. One of the first papers to discuss metamodel-based transformation was [46]. The production of model Mb from model Ma by a transformation Mt is called a model transformation. When the source and target metamodels are identical ($MMa = MMb$), we say that the transformation is endogenous. When this is not the case ($MMa \neq MMb$) we say the transformation is exogenous. An example of an endogenous transformation is a UML refactoring that transforms public class attributes into private attributes while adding accessors for each such transformed attribute. Another important idea is the fact that a model transformation is itself a model [1]. This means that the transformation program Mt may be considered as a model and as such conforms to a metamodel MMt. The consequences are quite important since this allows for example uniformly storing and retrieving different kinds of terminal models including transformations. Besides storage and retrieval, many other common operations may also be applied to such different kinds of models. The property of a transformation being a model allows also to deal easily with Higher-Order Transformations (HOTs), i.e. transformations taking other transformations as input or/and producing other transformations as output.

Transformations may thus be considered as models. But many other operations may be considered as transformations as well. For example verifications or measurements on a model can be expressed as transformations [41]. One can see then why large libraries of reusable modeling artifacts (mainly metamodels and transformations) will be needed.

As MDE developed, it became apparent that this was a branch of language engineering [38] dealing more specifically with DSLs. DSLs are programming or modeling languages that are tailored to solving specific kinds of problem by opposition to General Purpose Languages (GPLs) that aim to handle any kind of problem. Java is a programming GPL and UML is a modeling GPL. DSLs are already widely used for certain kinds of programming; probably the best-known of these is SQL, the language for manipulating relational data in databases. DSLs allow everybody to write programs using the concepts that actually make sense to their domain or to the problem they are trying to solve (for instance Matlab has matrices and lets the user express operations on them, Excel has cells, relations between cells, and formulas and allows to express simple computation in a visual declarative style, etc.). As well as making domain code programmers more productive, DSLs also tend to offer greater optimization opportunities. Programs written with these DSLs may be independent of the specific hardware they will eventually run on. MDE offers an improved way to the development of DSLs by allowing easier expression of abstract syntaxes (with KM3 for example [4]) and easier transformation on domain code – the equivalent of program or terminal models – (with ATL for example [3]).

3.2. Models and DSLs

In MDE the important concept is that of metamodel that is similar to ontology. A metamodel expresses the abstract syntax of a DSL. There are two ways of expressing such a DSL. Either one may reduce a general purpose modelling language like UML with mechanisms like profiles. Or alternatively one may directly define small metamodels that may be combined. The second alternative has been chosen and the AmmA platform may be seen as a DSL building framework. Basic building blocks are the kernel languages (KM3, ATL, AMW, TCS, etc.) on top of which the user may define new DSLs for domains like telephony, transportation, automotive, aeronautics, or any other.

When following the previously described principles, one may take advantage of the uniformity of the MDE organization. Considering similarly models of the static architecture and models of the dynamic behaviour of a system allows at the same time economy of concepts and economy of implementation. Considering models of products (e.g., software artifacts like UML) and models of processes (e.g., software processes like SPEM) may lead to a dual process/product organization. Considering transformation models, weaving models, and traceability models as special cases of correspondence models may also lead to simplicity and efficiency of implementations. These are a few of the use cases that are being explored in the team. A standard way consists to revisit an object-oriented solution to a problem by a new model-oriented solution and to compare them.

4. Application Domains

4.1. Overview

It is difficult to define a precise applicability domain for MDE because, by definition, the scope is the largest. Generic tools developed by the AtlanMod team may apply as well to information systems and to embedded systems. MDE is not even restricted to software engineering, but also applies to data engineering [42] and to system engineering [35]. There are a lot of problems in these application domains that may be addressed by model transformation techniques.

The AmmA tools are being currently used in projects related to embedded systems, like Lambda (see 7.7), Edona (see 7.4), OpenEmbeDD (see 7.1), etc. The domains addressed by these projects are mainly automotive, aeronautics, and transportation. Since our tools are offered as open-source, we may notice that several groups have used them in quite different areas like: Critical software, Real time, Formal methods, Web engineering, Ontology engineering, Web semantics, etc. However, some specific application areas have demonstrated particular suitability for MDE techniques, as detailed in the following sections.

4.2. Business Rule Engineering

One particularly successful area of application has been that of Business Rule Management Systems (BRMS). The ILOG Company has shown how to efficiently build MDE bridges between a set of languages (DSLs), some of them close to the platform (like IRL), some of them closer to the customers needs (like PRR). Operational conversions between these languages have been built with ATL and TCS in quite an efficient way.

4.3. Model-driven Reverse Engineering

One important and original domain that is being investigated by the AtlanMod team is the reverse engineering of legacy code. Here again this spans through a spectrum of application domains, the legacy being coded in such languages as ADA, Java, COBOL, C, C++ or even FORTRAN. We have shown how the reverse engineering practices may be advantageously revisited with the help of MDE and open-source tooling. The team has set up the MoDisco (see 5.5) project under *Eclipse.org* to investigate this and to federate the international research efforts in this domain. The main idea is that a metamodel may precisely express what we want to extract from a low-level legacy code. For example we can define a rational process to extract business rules from COBOL legacy programs. The rise in abstraction allowed by MDE may bring new hopes that reverse engineering can now move beyond ad-hoc practices. The MoDisco Eclipse project is being referenced by the OMG ADM (Architecture-Driven Modernization) normalization initative. The AtlanMod team has proposed to bridge the open-source MoDisco project and the ADM initiative at OMG [34]. In 2009, AtlanMod has notably worked on the coupling of static and dynamic models [23], the measure of reverse-engineered models [24], and model-based platform cartography [22].

4.4. Model-driven System Interoperability

Historically the first application area of MDE was code generation from abstract models. A typical example is generation of Java code from UML models or of platform-dependent from platform-independent software artifacts. As discussed above, the subject of reverse engineering was the second major historical application field for MDE, with the discovery of structured precise models from heterogeneous and often unstructured systems.

The two first applications of MDE were thus mainly forward and reverse engineering of software systems. In these contexts, transformations were executed at software production or at software maintenance time. In the recent period however, an important evolution occurred towards the application area of system interoperability. In this new context, transformations are performed while systems are in execution. For example, a set of transformations may keep two different tools synchronized, by exchange of structured data or even by interpretation of complex events. This approach revisits tool interoperability by considering the associated metamodels. Our proposal to the CESAR (see 7.9) reference technology platform uses these ideas for defining the notion of virtual tool [19]. The establishment of correspondences between technical spaces (e.g., Eclipse Modeling and Microsoft DSL tools or OSLO) follows similar schemas [30]. Another example is the interoperability of graphical tools [15].

5. Software

5.1. Introduction

The scheme followed until now by the team has always been the same. It has been illustrated by the work on model transformation for example. First we work on an innovative research idea to get it accepted by the community. Then we promote it towards partial normalization and we implement a proof of concept prototype. Following this, we publish an improved version of the prototype as an open-source contribution. Until now we have used the Eclipse foundation to this purpose. When this is established, the improved prototype has to find and gather a significant user community to show its adequacy to user needs. In case the approach is successful, it needs to be stopped at this point because a research team is not formatted to support a large user community. The solution experimented at this stage consists in making arrangements with industrial partners so that they may help in supporting the user community while the research resources of the team may be used to start an additional software prototype cycle.

The most ancient tool developed in the team is the ATL model transformation language. In this case the transfer state has been recently reached and we are presently working on the last phase of the process. The idea is to allow at the same time stabilization of the tool and ongoing research. Since these two objectives are in contradiction we have found a feasible solution. As the usability of the current version will be improved by an industrial partner in the *ATL* initiative, other new research branches will be opened in the *ATL-Research* initiative. Needless to say, both initiatives will continue to be based on open-source practices.

The other prototypes developed by the team have not yet reached the same recognition level, but they are following a similar path. We name AmmA (see 5.7) the set of software tools developed by the AtlanMod team.

5.2. The ATL Model Transformation Language

Participants: Frédéric Jouault [contact], Jean Bézivin.

URL: http://www.eclipse.org/m2m/atl/

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [8].

Due to the previous iterations, the architecture of the ATL model transformation language is highly modular and based on a generic model engineering virtual machine and on a bootstrapped compiler itself running on this virtual machine. The central idea in Frédéric Jouault's Ph.D. thesis [45] of using MDE tools to build other MDE tools (or using DSLs to build other DSLs) has been quite productive. Seeking conceptual simplicity also led to implementation efficiency since ATL currently provides one of the most efficient solutions for model transformation.

The decision was taken early to develop the new framework as an open-source solution. The Eclipse community provided a rich environment for this. In addition to being based on Eclipse, the framework was made fully available on *Eclipse.org*, which allowed to build a strong user community and to get contributor help. In 2004 IBM gave an Eclipse innovation award to the ATL project. The CARROLL consortium (Thales, CEA, and INRIA) and the ModelWare European Integrated project funded the development of ATL. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see *Eclipse.org/m2m*). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

5.3. AMW (AtlanMod Model Weaver)

Participants: Jean Bézivin [contact], Frédéric Jouault.

URL: http://www.eclipse.org/gmt/amw/

AMW is a component-based platform for model weaving, i.e. establishing and managing abstract correspondences between models. The platform is based on the Eclipse contribution mechanism: components are defined in separate plugins. The plugins are further interconnected to create the model weaver workbench. Components for user interface, matching algorithms and serialization of models may be plugged as necessary. We extended the Eclipse EMF architecture for model manipulation to coordinate the weaving actions. We use the EMF reflective API to obtain a standard weaving editor which adapts its interface according to metamodels modifications. The ATL transformation engine is plugged as the standard transformation platform. AMW is released as open-source software under the Eclipse Public Licence and available as an Eclipse plugin. AMW is being used by more than 40 user sites, including research labs and major companies (NASA, BAE, Versata, Obeo, etc.).

5.4. TCS (Textual Concrete Syntax)

Participant: Frédéric Jouault [contact].

URL: http://www.eclipse.org/gmt/tcs/

It is often necessary to define concrete syntaxes for metamodels, which only define abstract syntaxes. TCS is a language in which context-free concrete syntaxes can be defined by specifying how each concept of a metamodel is represented textually. Once such a definition has been defined in the form of a TCS model, it is possible to: 1) parse programs into models, 2) pretty-print models into programs, and 3) edit programs with a rich text editor supporting syntax highlighting, code completion, outline, text hovers, hyperlinks, etc. Many textual languages have already been specified with TCS: 1) within AmmA (see 5.7) with ATL, KM3, TCS itself, etc. 2) in other contexts with SQL, LOTOS, EBNF, etc. TCS currently generates ANTLR v3 grammars that are LL(*).

5.5. MoDisco (Model Discovery)

Participants: Hugo Brunelière [contact], Jean Bézivin, Frédéric Jouault.

URL: http://www.eclipse.org/gmt/modisco/

MoDisco (for Model Discovery) is an Eclipse component that gathers contribution from several academic and industrial partners in the field of model driven reverse engineering. The goal of the project is to federate common efforts in the transformation of legacy systems into models. The extraction process is metamodel driven, i.e. (1) all extracted models conform to a given metamodel and (2) the discoverer itself is generated from the metamodel, usually in a semi-automatic way. In some cases the legacy system is structured which greatly facilitates model extraction. For example if the legacy is composed of code (e.g., ADA, COBOL, Java, Visual Basic, etc.), the grammar and the target metamodel may be jointly used in order to generate the discoverer. Once the model has been extracted from the legacy, it can be measured, understood or manipulated by way of model transformations in languages like ATL.

5.6. AM3 (AtlanMod MegaModel Management)

Participants: Hugo Brunelière [contact], Guillaume Doux, Jean Bézivin, Frédéric Jouault.

URL: http://www.eclipse.org/gmt/am3/

The AtlanMod Megamodel Management tool offers several functionalities for modeling in the large [2], i.e. for handling several related models. This component hosts the various libraries of terminal models, metamodels and transformation that can be reused. But the main tool available in AM3 is a generic megamodel manager allowing for example to browse a set of related models. This manager knows the semantic relations between all these models. These relations are often associated to a given weaving model allowing not only navigating the traces between models, but also the traces between model elements. Since the links are stored externally as weaving models, the participating models do not get polluted and may be used as they are. Furthermore it is possible to handle multiple traceability chains going through similar models. The generic tool for megamodel management has been used by different partners for several use cases like operationalization of chains of transformations.

5.7. The AmmA ToolBox

ATL, AMW, TCS, MoDisco, and AM3 are among the most important *Eclipse.org* components produced by the AtlanMod team. However there are also other components and a lot of functionalities, examples, and use cases made available and necessary to express solutions to many problems. The whole set of contributions composes the AmmA platform [11].

One difficulty of developing research prototypes intended also to become practical industrial tools is the necessity to stay partially aligned with the standards of the market at the time the research is developed. We have managed to keep the ATL prototype sufficiently synchronized with the OMG QVT evolving recommendation without prejudice to the conceptual rigor of the proposal. In the domain of metamodeling we have similarly kept aligned with the OMG MOF and with the Eclipse Java-based Ecore by defining an independent solution called KM3 with good semantic properties described in [4]. The price to pay to remain reasonably independent from standards and industrial practical applications is bearable since today we have one of the best self-contained modeling frameworks and this is mainly due to the simplicity of the KM3 solution.

The success of the ATL language allowed building one of the first libraries of model transformations as part of the *Eclipse.org/m2m/atl* project. This library received contributions from many parties. It consists of a rich set of components, at various levels of maturity. One of the multiple conclusions drawn from this experience is that the number of modeling artifacts (terminal models, metamodels, transformations, etc.) involved in a given project is quite important. This creates an accidental complexity that may prevent MDE to scale up in practical industrial environments. In order to show that MDE is able to deal with complex situations beyond toy problems, new solutions need to be defined. The experimental construction of tools to deal with complexity need to be engaged as soon as possible. This is why we started the study of the megamodel management tool to handle complex situations when there is a high number of DSLs involved with a lot of semantic relations between these DSLs.

In retrospect, one of the main contributions of the second period of research in the ATLAS modeling group has been to show the power of the transformation paradigm over more classical direct execution schemes, based on Java or other general purpose languages for example. Through many examples we have demonstrated that many problems could be expressed as chains of information transformation. There is an obvious gain when we can express a problem as a chain of transformations in a declarative modeling language over a similar expression as a set of imperative statements in a general purpose language. However we do not consider ATL as the definitive model transformation language. On the contrary we view it only as a representative of a broad family of such languages yet to be precisely identified. Fortunately parts of the AmmA toolbox may be reused in the implementation of these new languages. Our team has also been the first one to show the deep practical impact of Higher-Order Transformations (HOTs) in practical modeling environments. This has been made possible by the fact that ATL is a true DSL, and that ATL programs may be considered as models conforming to a given metamodel [1].

6. New Results

6.1. Model Transformation

Participants: Frédéric Jouault [contact], Jean Bézivin.

As a legacy of our previous work there are a number of assets that we may use [47], [48], [49], [50], [51], [52] [10]. The ATL virtual machine and the ATL development environments are stabilized. There is a large user community and an initial library of transformations. Part of these assets are being industrialized with the help of a local company (Obeo) but kept in the open-source mode. The resources freed in this process are invested in pursuing research on model transformation.

In 2009, AtlanMod studied four specific model transformation issues:

- **Performance** of model transformation applied to large models. This work notably lead to the publication of a case study at the 5th International Workshop on Graph-Based Tools (GraBaTs'2009). This case study [33] focuses on the implementation of program comprehension tasks using model transformation tools. In this domain, large models obtained by parsing source code, and consisting of several millions of elements need to be processed. AtlanMod also submitted a solution [32] to this case study, as well as a solution to a synthesis case study around BPEL and BPMN [31] in the same workshop.

- **Complex model navigation** in model transformation relies in some cases on OCL expressions, which are more verbose than necessary (e.g., to express a JOIN, a transitive closure). This leads to complex code that is difficult to write, maintain, and optimize. In order to address this issue, we have worked on the HLN [25] (Higher-Level Navigation) language. This language defines explicit language constructs for some advanced navigation problems.
- **Higher-Order Transformations** (HOTs) enable complex transformation scenarios in which a transformation may for instance generate other transformations. It is possible to write such transformations in ATL because every ATL transformation is a model. After having used HOTs during several years, we have started a systematic study [26] of their applications.
- **Transformation chaining constraints** need to be well understood before composing transformations. Otherwise, invalid combinations may yield to errors. However, these constraints are not always obvious (e.g., they may depend on source and target metamodel subsets). In [14], we have studied the automatic discovery of some chaining constraints.

Additionally, AtlanMod studied the application of ATL in system validation chains [13].

6.2. Model Weaving and Model Matching

Participants: Frédéric Jouault [contact], Jean Bézivin, Kelly Garcés.

Our initial investigation in the area of model weaving has lead us to consider three problems: (a) the representation of correspondences between models, (b) the computation of correspondence sbetween models and (c) the utilization of correspondences between models. The Eclipse AMW prototype has allowed to experiment in these three situations and to propose a first spectrum of use cases [44]. Starting from this situation we are improving the AMW prototype to investigate in several areas for example:

Problem 1. Provide a development environment such that, given two metamodels, one may partially automate the creation of matching rules between them. One way to achieve this is to define a new DSL more adapted to this task than a simple transformation language. Also the expression of matching heuristics requires a specific DSL. In 2009, AtlanMod has defined the AML [29] (AtlanMod Matching Language) for this purpose.

Problem 2. Provide a development environment such that, given two metamodels and matching rules between them, one may easily create transformation code. This is also related to the next problem (Model Driven Software Modernization).

Problem 3. Model weaving may also be applied to model driven software modernization. In a given organization, there are a lot of evolving software fragments. One typical problem is the change of versions of schemas, grammars, metamodels, APIs, etc. We are working on the partial automation of these operations. The idea is to compute weaving difference models between different versions by the way of a sequence of matching heuristics. From there we can apply higher order transformations to generate the tools that will perform the conversion [18].

6.3. Global Model Management

Participants: Hugo Brunelière [contact], Guillaume Doux, Jean Bézivin, Frédéric Jouault.

As noted above, MDE generates a lot of artifacts that need to be handled if we want to harness the accidental complexity and if we wish MDE to scale up in the context of collective usage by teams of engineers of several models in different versions. The problem we tackled here is part of the "tower of models" grand challenge proposed by Robin Milner.

Problem 1. In our recent work we have found that what we must have more than single-shot transformations. We also need long chains of transformations or even complex graphs of transformations. For example in the collaboration with ILOG more than 25 transformations are being built and the engineer needs to control the corresponding complexity. With the notion of a megamodel we propose solutions to this problem. A megamodel is a model such that some of its elements are themselves models. From there, one problem is to make sure that transformations are composed correctly. To address this issue, AtlanMod worked on a type system for megamodels [27].

Problem 2. As many companies realize that they may not rely anymore on one unique modeling language like UML, they face the problem of allowing their engineers to navigate between numbers of different models based on heterogeneous DSLs. This is an open problem that we are solving on a conceptual and practical basis. Until now all solutions to this problem have been specialized ad-hoc implementations [17], [20].

6.4. Business Rule Engineering

Participants: Mathias Kleiner [contact], Jean Bézivin, Frédéric Jouault.

The field of BRMS is characterized by a number of normative, open-source, or proprietary systems and languages (ILOG JRules, JBoss Drools, Fair-Isaac Blaze Advisor, etc.), allowing the expression of various solutions to business problems at a high abstraction level, but with heterogeneous sets of capabilities and languages. SBVR (Semantics of Business Vocabulary and Business Rules) is a new standard in this domain, for which tool support is in its infancy. During his postdoc, Mathias Kleiner worked on the parsing and analysis of SBVR textual statements, as well as on their transformation into a UML model [21] with a set of open-source MDE tools previously built at AtlanMod (the AmmA tool suite [9]). In addition to metamodels and model transformations, the creation of an SBVR syntax model from plain english text has been performed using constraint-based parsing techniques. In addition to the actual result, the chosen approach notably demonstrates that MDE tools can be coupled with other technologies in order to achieve more complex tasks (e.g., parsing SBVR-compliant english text). This work has been done in collaboration between the AtlanMod team in Nantes and the IBM/ILOG company, and is an extension of our joint work on BRMS interoperability discussed in [16], [28].

7. Contracts and Grants with Industry

7.1. RNTL OpenEmbeDD (2006-2009)

Participants: Jean Bézivin, Frédéric Jouault.

The project involves Airbus, Anyware, CEA, CS, FT R&D, LAAS, Thales DAE, Thales TRT and Verimag. OpenEmbeDD is an Eclipse open-source platform based on the model engineering principles for the software engineering of embedded systems. In this project, AtlanMod provides a model transformation virtual machine, and components for global model management. Furthermore, we address the interoperability between the AmmA and GME platforms.

7.2. IP Modelplex (2006-2010)

Participants: Hugo Brunelière, Jean Bézivin, Frédéric Jouault.

The project (with Thales, IBM, Sodifrance, SAP, etc.) aims at defining a coherent infrastructure for the development of complex systems, where complexity corresponds to several factors like size, heterogeneity, dynamic evolution, distribution and subsystem autonomy. Examples of highly heterogeneous systems are legacy systems that have been built and adapted on long period of time, using different technologies. Model driven reverse engineering is also an important problem which we address.

7.3. ANR non thématique FLFS (2006-2009)

Participants: Jean Bézivin, Frédéric Jouault, Kelly Garcés.

FLFS means "Families of Languages for Families of Systems" The objective of the FLFS project (with Ascola, Phoenix) is to study the continuum domain modeling/implementation with the complementary technologies of model engineering, domain specific languages and aspect oriented programming. In this project, the AtlanMod team develops concrete solutions based on the AmmA platform and more specifically on the ATL and TCS languages.

From a software development viewpoint, a program family represents a domain of expertise, that is, a vocabulary, notations, rules and protocols that are specific to a domain. For example, the telephony domain consists of a set of concepts, rules, protocols and interfaces that represent a precise framework to be used for the development of telephony services.

Our goal is to place domain expertise at the centre of the software development process. It is aimed to lift the current limitations of software engineering regarding large scale software production, robustness, reliability, maintenance and evolution of software components. Our key innovation is to introduce a software development process parameterized with respect to a specific domain of expertise. This process covers all the stages of software development and combines the following three emerging approaches:

- Domain-specific modelling, also known as model engineering;
- Domain-specific languages, in contrast with general-purpose languages;
- Generative programming and in particular aspect-oriented programming as a means to transform models and programs.

Our partners in this project are are the Ascola (M. Südholt) and Phoenix (C. Consel) INRIA teams. The duration of the project is 36 months. It started in December 2006. The AtlanMod funding part amounts to 70 kEUR. The Web page is: http://flfs.emn.fr. This project has been terminated in 2009.

7.4. EDONA, Paris Competitivity Cluster "System@tic" (2007-2010)

Participants: Frédéric Jouault, Jean Bézivin.

EDONA (for *Environnements de Développement Ouverts aux Normes de l'Automobile*: Environments of Development Open to the Standards of the Car) is a project of the pole of competitiveness System@tic Paris-Area. It has as an objective the construction of an open platform facilitating the realization of chains of development trade modular, interoperable, and adaptable the various needs of the actors and trades of the car industry. The project is directed by Renault and the form chosen for EDONA is the creation of a technological platform of reference then its specialization on applicative products of the sector. In this project, the AtlanMod team works with Obeo on the industrialization of ATL.

7.5. IdM++, ANR (2008-2011)

Participants: Jean Bézivin, Hugo Brunelière, Guillaume Doux, Frédéric Jouault.

IdM++ is a project involving ILOG, CEA, Prima Solutions, and AtlanMod. The main goal is to investigate advanced issues in model engineering. The team is particularly in charge of WP 2, on global model management. In this context, the navigability between various DSLs will be studied. One of the case studies is provided by ILOG and the other one is a typical project management organization. In addition to these direct tasks, the team will participate to joint study with ILOG of the configuration problem. Total allocated budget amounts 810 kEUR of which the team is sharing 250kEUR. on three years. (http://www.emn.fr/x-info/idmpp/index.php/Accueil)

7.6. Happy, Images et Réseaux Competitivity Cluster région Bretagne (2008-2009)

Participants: Guillaume Doux, Frédéric Jouault, Jean Bézivin.

In this project the main goal is to study sustainable development architecture which enables one to design a sustainable application, taking early into account that technologies will change. An application is composed of a set of models; these models may include other models. These models, which define the application and the processed data, were designed by computer scientists with help of tools they master (Rational Rose, Visual Paradigm, Entreprise Architect, etc.). Users (functional specialists) use a simpler tool which enables them, initially, to enrich only these models by modifying the user interface. These modifications are enabled with help of a very simple Web 2.0 tool working through a web navigator. The focus of the project is on Model-Driven Web Engineering (MDWE). MDWE usually splits the definition of a web application in terms of data (a.k.a. business-model), presentation, and navigation models, following thus a separation similar to the Model-View-Controller (MVC) pattern. The MDWE community is today leaded by several actors who have already defined some metamodels for web-application design (e.g., WebML or UWE).

7.7. Lambda, Paris Competitivity Cluster "System@tic" (2008-2010)

Participants: Jean-Sébastien Sottet, Jean Bézivin.

In the context of embedded software deployed on "off-the-shelf" execution platforms, the LAMBDA project (System@tic Paris-Region) has two major goals: to demonstrate the technical feasibility and the interest of model libraries by formalising the key properties of execution platforms, and to reconcile appropriated standards (SysML, MARTE, AADL, IP-XACT) with de facto standards (already implemented by widespread analysis and simulation tools). Lambda is a three-year project gathering 14 partners with an overall budget of 5.30 MEUR. LAMBDA means Libraries for Applying Model Based Development Approaches. The project started on June 1, 2008. AtlanMod is involved in Task T2.2: analysis of requirement for scalability.

7.8. CONICYT Chili-INRIA HOT MaTE Project (2008-2010)

Participants: Jean Bézivin, Frédéric Jouault, Hugo Brunelière, Andrés Vignaga.

HOT MaTE stands for Higher-Order Transformation Model and Transformation Engineering. This CONICYT-INRIA project is a collaboration with the MaTE research group from the University of Chile. The objective of this project is to advance the state of the art in model transformation, and global model management. In particular, we study how Higher-Order Transformations can be represented in a megamodel, as well as megamodel evolution. Prototypes are developed on the AmmA platform, and more especially using ATL for transformations, and AM3 for megamodeling.

7.9. CESAR, ARTEMIS JOINT UNDERTAKING (2009-2012)

Participants: Jean Bézivin, Frédéric Jouault, Cauê Clasen.

CESAR (http://cesarproject.eu/) stands for Cost-Efficient methods and processes for SAfety Relevant embedded systems, and is a European funded project from ARTEMIS JOINT UNDERTAKING.

The three transportation domains automotive, aerospace, and rail, as well as the automation domain share the need to develop ultra-reliable embedded systems to meet societal demands for increased mobility and ensuring safety in a highly competitive global market. To maintain the European leading edge position in the transportation as well as automation market, CESAR aims to boost cost efficiency of embedded systems development and safety and certification processes by an order of magnitude.

CESAR pursuits a multi-domain approach integrating large enterprises, suppliers, SME's and vendors of cross sectoral domains and cooperating with leading research organizations and innovative SME's. In particular, we work on the Reference Technology Platform, which aims at tool integration.

8. Other Grants and Activities

8.1. Regional Actions

In 2009, the AtlanMod team initiated a regular series of events called *Les Jeudis des Modèles*[1]. Every two months approximately, an international expert from the model engineering community is invited to come and present a subject of research or innovation of interest to the scientific and industrial community. These events typically attract between 60 and 80 researchers, students and industrials. *Les Jeudis des Modèles* event has become in 2009 a regional rendez-vous of the model engineering community, attracting people from Rennes, Vannes, La Rochelle, and many other places beyond Nantes. We have invited Patrick Albert (IBM), Nicolas Rouquette (NASA/JPL), Ed Merks (Macro Modeling), Ivar Jacobson (IJ company), and Sridhar Iyengar (IBM). It should be noted that these visits allow to organize different meetings between the researchers of the AtlanMod team and the various industrials attending the main presentation.

9. Dissemination

9.1. Interaction with the Scientific Community

In 2009, the AtlanMod team has organized the 1st International Workshop on Model Transformation with ATL[2] (MtATL 2009) in the context of the RMLL meeting in Nantes (Rencontres Mondiales du Logiciel Libre). This event attracted several thousand attendees to Nantes and the MtATL workshop gained high visibility in this context. The workshop was organized as a scientific event with paper selection presentation, and publication in an open source repository.

9.2. Editorial Boards and Program Committees

Participation to editorial boards of scientific journals:

- Jean Bézivin: SoSym, IBIS, JOT.

Participation to steering committees:

- Jean Bézivin: AITO (ECOOP), TOOLS, ICMT.

Participation to conference program committees:

- Jean Bézivin: MODELS'2009, ECMDA'2009, ICMT'2009, EDOC'2009, ICEIS'2009, SLE'2009, GTTSE'2009
- Frédéric Jouault: ICMT'2009, IDM'2009, SLE'2009, MtATL'2009 (chair).

9.3. PhD and Habilitation Juries

Participant: Jean Bézivin.

Jean Bézivin was a member of the following PhD juries:

- A. Hesselunds, PhD committee, IT University of Copenhagen, Denmark, 06/02/2009
- J.M. Vara PhD committee, Universidad Rey Juan Carlos, Madrid, Spain, 06/11/2009
- S. Luquet, Doctorat, Université Blaise Pascal Clermond Ferrand (rapporteur), 14/12/2009
- D. Ouagne, Université Pierre et Marie Curie (rapporteur), 15/12/2009
- S. Ciraci, PhD committee, University of Twente, The Netherlands, 17/12/2009

[1] http://www.emn.fr/z-info/jmodeles/
[2] http://www.emn.fr/z-info/atlanmod/index.php/MtATL2009

9.4. Invitations and Participations to Seminars

JISBD'2009, September 2009, San Sebastian, Spain.: *Advances in Model Driven Engineering*, Invited talk by Jean Bezivin.

SLE'2009, October 2009, Denver, USA.: *If MDE is the answer, then what is the question?*, Invited talk by Jean Bezivin.

2nd workshop on interoperability of enterprises applications, October 2009, European INTEROP-Vlab, Tarbes: *Model Driven Interoperability*, Invited talk by Jean Bezivin.

MDDay 2009, November the 26th, 2009, Paris, France: *Si l'Ingénierie Dirigée par les modèles est la solution, alors quel est le problème?*, Invited talk by Jean Bezivin.

AiSE Workshop: Advances in Software Evolution, December 2009, University of Twente, The Netherlands, *The three ages of Model Driven Engineering*, Invited talk by Jean Bezivin.

9.5. Teaching

Participants: Jean Bézivin, Frédéric Jouault, Kelly Garcés, Hugo Brunelière.

The members of the AtlanMod team have presented the basis of Model Engineering to several types of students:

- **EMN GSI option.** Jean Bézivin and Frédéric Jouault delivered two introductory lectures to an industrial session on model transformation in this EMN program (final year).
- **International EMOOSE Master.** The AtlanMod team is in charge in 2009 of a 30 hours module on model engineering.

The two following tutorials have been given at EclipseCon 2009 by Frédéric Jouault:

- **Model Refactoring and other Advanced ATL Techniques** in collaboration with the Obeo company.
- **From Text to Models and Back Again**, a tutorial about TCS.

In 2009, AtlanMod has been working on the creation of the MDE international post-graduate specialization Diploma in Model Driven Engineering (MDE) for Software Management at École des Mines de Nantes[3]. Its objective is to train engineers to manage complex projects in various IT fields with the latest cutting-edge modeling technologies.

9.6. Involvment in the Eclipse Community

Participants: Jean Bézivin, Frédéric Jouault, Hugo Brunelière, Guillaume Doux, Mathias Kleiner, Kelly Garcés.

The AmmA platform components are made available on *Eclipse.org* by AtlanMod, which is involved in the Eclipse community as follows:

- Jean Bézivin is project lead of the GMT project.
- Frédéric Jouault is project lead of the M2M (Model-to-Model transformation) and TMF (Textual Modeling Framework) projects as well as component lead on M2M/ATL, and TMF/TCS. He is also committer on GMT/AM3, and GMT/AMW.
- Hugo Brunelière is component lead of GMT/AM3, and GMT/MoDisco.
- Guillaume Doux is committer on GMT/AM3.
- Mathias Kleiner is an M2M/ATL contributor.
- Kelly Garcés is an M2M/ATL contributor.

[3] http://www.mines-nantes.fr/go-mde

10. Bibliography

Major publications by the team in recent years

[1] J. BÉZIVIN, F. BÜTTNER, M. GOGOLLA, F. JOUAULT, I. KURTEV, A. LINDOW. *Model Transformations? Transformation Models!*, in "Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006, Genova, Italy, October 1-6, Proceedings, LNCS 4199", Springer Berlin / Heidelberg, 2006, p. 440–453.

[2] J. BÉZIVIN, F. JOUAULT, P. ROSENTHAL, P. VALDURIEZ. *Modeling in the Large and Modeling in the Small*, in "Proceedings of the European MDA Workshops: Foundations and Applications, MDAFA 2003 and MDAFA 2004, LNCS 3599", U. ASSMANN, M. AKSIT, A. RENSINK (editors), Springer-Verlag GmbH, 2005, p. 33–46.

[3] F. JOUAULT, F. ALLILAIRE, J. BÉZIVIN, I. KURTEV. *ATL: a Model Transformation Tool*, in "Science of Computer Programming", vol. 72, n^o 3, Special Issue on Second issue of experimental software and toolkits (EST), 2008, p. 31–39.

[4] F. JOUAULT, J. BÉZIVIN. *KM3: a DSL for Metamodel Specification*, in "Proceedings of 8th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems, LNCS 4037, Bologna, Italy", 2006, p. 171–185.

[5] F. JOUAULT, J. BÉZIVIN, I. KURTEV. *TCS: a DSL for the Specification of Textual Concrete Syntaxes in Model Engineering*, in "GPCE'06: Proceedings of the fifth international conference on Generative programming and Component Engineering, Portland, Oregon, USA", 2006, p. 249–254.

[6] F. JOUAULT, I. KURTEV. *On the Architectural Alignment of ATL and QVT*, in "Proceedings of the 2006 ACM Symposium on Applied Computing (SAC 06), Dijon, France", ACM Press, 2006, p. 1188–1195.

[7] F. JOUAULT, I. KURTEV. *Transforming Models with ATL*, in "Satellite Events at the MoDELS 2005 Conference: MoDELS 2005 International Workshops OCLWS, MoDeVA, MARTES, AOM, MTiP, WiSME, MODAUI, NfC, MDD, WUsCAM, Montego Bay, Jamaica, October 2-7, 2005, Revised Selected Papers, LNCS 3844", J.-M. BRUEL (editor), Springer Berlin / Heidelberg, 2006, p. 128–138.

[8] F. JOUAULT, I. KURTEV. *On the Interoperability of Model-to-Model Transformation Languages*, in "Science of Computer Programming", vol. 68, n^o 3, 2007, p. 114-137, Special Issue on Model Transformation.

[9] I. KURTEV, J. BÉZIVIN, F. JOUAULT, P. VALDURIEZ. *Model-based DSL Frameworks*, in "Companion to the 21st Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2006, October 22-26, 2006, Portland, OR, USA", ACM, 2006, p. 602–616.

[10] R. LEMESLE. *Transformation rules based on meta-modeling*, in "Proceedings of the Second International Enterprise Distributed Object Computing conference (EDOC'98)", 1998, p. 113–122.

Year Publications

Articles in International Peer-Reviewed Journal

[11] F. JOUAULT, J. BÉZIVIN, M. BARBERO. *Towards an advanced model-driven engineering toolbox*, in "Innovations in Systems and Software Engineering", vol. 5, n^o 1, 2009, p. 5–12.

Articles in National Peer-Reviewed Journal

[12] J. BÉZIVIN, F. JOUAULT. *Acquis et Défis de l'Ingénierie Dirigée par les Modèles*, in "Génie Logiciel", vol. 2, n° 90, 2009, p. 55–60.

[13] O. CONSTANT, F. JOUAULT. *Mise en oeuvre d'ATL dans les chaînes de validation de systèmes*, in "Génie Logiciel", n° 89, 2009, p. 13–18.

International Peer-Reviewed Conference/Proceedings

[14] R. CHENOUARD, F. JOUAULT. *Automatically Discovering Hidden Transformation Chaining Constraints*, in "Proceedings of Models'O9, ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems", Springer Berlin / Heidelberg, 2009, p. 92–106.

[15] Z. DEMIREZEN, Y. SUN, J. GRAY, F. JOUAULT. *Supporting Tool Reuse with Model Transformation*, in "18th International Conference on Software Engineering and Data Engineering (SEDE-2009)", ISCA, 2009, p. 119-125.

[16] M. DIDONET DEL FABRO, P. ALBERT, J. BÉZIVIN, F. JOUAULT. *Achieving rule interoperability using chains of model transformations*, in "Proceedings of the International Conference on Model Transformation, ICMT'09", 2009, p. 249–259.

[17] M. FRITZSCHE, H. BRUNELIÈRE, B. VANHOOFF, Y. BERBERS, F. JOUAULT, W. GILANI. *Applying Megamodelling to Model-Driven Performance Engineering*, in "Proceedings of the 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS)", 2009, p. 244–253 NL UK .

[18] K. GARCÉS, F. JOUAULT, P. COINTE, J. BÉZIVIN. *Managing Model Adaptation by Precise Detection of Metamodel Changes*, in "Proceedings of ECMDA 2009 (Springer)", 2009, p. 34–49.

[19] F. JOUAULT, T. GUÉGUEN. *Integration by Model-driven Virtual Tool*, in "Proceedings of the 2nd ECMDA Workshop on Model-Driven Tool and Process Integration", 2009, p. 47–52.

[20] F. JOUAULT, B. VANHOOFF, H. BRUNELIÈRE, G. DOUX, Y. BERBERS, J. BÉZIVIN. *Inter-DSL Traceability and Navigability Support by Combining Megamodeling and Model Weaving*, in "Special Track on the Coordination Models, Languages and Applications at the 25th ACM Symposium On Applied Computing (SAC 2010)", 2010, to appear.

[21] M. KLEINER, P. ALBERT, J. BÉZIVIN. *Parsing SBVR-based controlled languages*, in "Proceedings of Models'O9, ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems", Springer Berlin / Heidelberg, 2009, p. 122–136.

[22] V. MAHÉ, F. JOUAULT, H. BRUNELIÈRE. *Megamodeling Software Platforms: Automated Discovery of Usable Cartography from Available Metadata*, in "Proceedings of International Workshop on Reverse Engineering Models from Software Artifacts (REM'09) Organized in conjunction with WCRE'09", 2009, p. 29–32.

[23] J.-S. SOTTET, F. JOUAULT, J. BÉZIVIN. *Coupling Static and Dynamic Models Information*, in "Proceedings of International Workshop on Reverse Engineering Models from Software Artifacts (REM'09) Organized in conjunction with WCRE'09", 2009, p. 25–28.

[24] J.-S. SOTTET, M. VENISSE, F. JOUAULT, J. BÉZIVIN, V. FADY. *Measuring Discovered Models*, in "Proceedings of International Workshop on Reverse Engineering Models from Software Artifacts (REM'09) Organized in conjunction with WCRE'09", 2009, p. 21–24.

[25] J. SÁNCHEZ CUADRADO, F. JOUAULT, J. G. MOLINA, J. BÉZIVIN. *Experiments sith a Higher-Level Navigation Language*, in "Proceedings of the International Conference on Model Transformation, ICMT'09", 2009, p. 229–238.

[26] M. TISI, F. JOUAULT, P. FRATERNALI, S. CERI, J. BÉZIVIN. *On the Use of Higher-Order Model Transformations*, in "Proceedings of ECMDA 2009 (Springer)", 2009, p. 18–33.

[27] A. VIGNAGA, F. JOUAULT, M. C. BASTARRICA, H. BRUNELIÈRE. *Typing in Model Management*, in "Proceedings of the International Conference on Model Transformation, ICMT'09", 2009, p. 197–212.

National Peer-Reviewed Conference/Proceedings

[28] M. DIDONET DEL FABRO, P. ALBERT, J. BÉZIVIN, F. JOUAULT. *Industrial-strength Rule Interoperability using Model Driven Engineering*, in "Proceedings of the 5ème Journée sur l'Ingénierie Dirigée par les Modèles (IDM09)", 2009, p. 163–178.

[29] K. GARCÉS, F. JOUAULT, P. COINTE, J. BÉZIVIN. *A Domain Specific Language for Expressing Model Matching*, in "Proceedings of the 5ème Journée sur l'Ingénierie Dirigée par les Modèles (IDM09)", 2009, p. 33–48.

Workshops without Proceedings

[30] H. BRUNELIÈRE, F. JOUAULT. *Possible Benefits of Bridging the Eclipse-EMF and Microsoft-Oslo Technical Spaces*, in "Long talk at Eclipse Summit Europe (ESE) 2009", 2009.

[31] G. DOUX, F. JOUAULT, J. BÉZIVIN. *Transforming BPMN process models to BPEL process definitions with ATL*, in "5th International Workshop on Graph-Based Tools, GraBaTs 2009", 2009.

[32] F. JOUAULT, J.-S. SOTTET. *An AmmA/ATL Solution for the GraBaTs 2009 Reverse Engineering Case Study*, in "5th International Workshop on Graph-Based Tools, GraBaTs 2009", 2009.

[33] F. JOUAULT, J.-S. SOTTET. *Program Comprehension Case Study for GraBaTs 2009*, in "5th International Workshop on Graph-Based Tools, GraBaTs 2009", 2009.

References in notes

[34] ATLANMOD TEAM. *Software Modernization meets open source: the MoDisco Eclipse Initiative*, n^{o} admtf/08-06-14, OMG, June 2008, http://www.omg.org/cgi-bin/doc?admtf/2008-06-14, OMG document.

[35] M. BARBERO, F. JOUAULT, J. BÉZIVIN. *Model Driven Management of Complex Systems: Implementing the Macroscope's vision*, in "Proceedings of the 15th Annual IEEE International Conference and Workshop on

Engineering of Computer Based Systems (ECBS 2008), 31 March - 4 April 2008, Belfast, Northern Ireland", IEEE Computer Society, 2008, p. 277–286.

[36] J. BÉZIVIN. *From Object Composition to Model Transformation with the MDA*, in "TOOLS '01: Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39), Washington, DC, USA", IEEE Computer Society, 2001, p. 350–354.

[37] J. BÉZIVIN. *sNets: A First Generation Model Engineering Platform*, in "Satellite Events at the MoDELS 2005 Conference: MoDELS 2005 International Workshops OCLWS, MoDeVA, MARTES, AOM, MTiP, WiSME, MODAUI, NfC, MDD, WUsCAM, Montego Bay, Jamaica, October 2-7, 2005, Revised Selected Papers, LNCS 3844", Springer Berlin / Heidelberg, January 2006, p. 169–181.

[38] J. BÉZIVIN, R. HECKEL. *04101 Summary – Language Engineering for Model-driven Software Development*, in "Language Engineering for Model-Driven Software Development, Dagstuhl, Germany", J. BÉZIVIN, R. HECKEL (editors), Dagstuhl Seminar Proceedings, n^{o} 04101, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.

[39] J. BÉZIVIN. *On the Unification Power of Models*, in "Software and System Modeling", vol. 4, n^{o} 2, 2005, p. 171–188.

[40] J. BÉZIVIN, O. GERBÉ. *Towards a Precise Definition of the OMG/MDA Framework*, in "ASE'01, Automated Software Engineering, San Diego, USA", IEEE Computer Society Press, 2001, p. 273-280.

[41] J. BÉZIVIN, F. JOUAULT. *Using ATL for Checking Models*, in "Proceedings of the International Workshop on Graph and Model Transformation (GraMoT), Tallinn, Estonia", 2005.

[42] M. DIDONET DEL FABRO, J. BÉZIVIN, F. JOUAULT, P. VALDURIEZ. *Applying Generic Model Management to Data Mapping*, in "Proceedings of the Journées Bases de Données Avancées (BDA05)", 2005.

[43] M. DIDONET DEL FABRO. *Gestion de métadonnées utilisant tissage et transformation de modèles*, Université de Nantes, 2007, Ph. D. Thesis.

[44] M. DIDONET DEL FABRO, P. VALDURIEZ. *Towards the Efficient Development of Model Transformations using Model Weaving and Matching Transformations*, in "Software and Systems Modeling", 2008, p. 305–324.

[45] F. JOUAULT. *Contribution à l'étude des langages de transformation de modèles*, Université de Nantes, 2006, Ph. D. Thesis.

[46] R. LEMESLE. *Transformation rules based on meta-modeling*, in "Proceedings of the Second International Enterprise Distributed Object Computing Workshop (EDOC'98), La Jolla, CA, USA", 1998, p. 113–122.

[47] Y. SUN, Z. DEMIREZEN, F. JOUAULT, R. TAIRAS, J. GRAY. *A Model Engineering Approach to Tool Interoperability*, in "Proceedings of the 1st International Conference on Software Language Engineering (SLE)", Springer Berlin / Heidelberg, 2008, p. 178–187.

[48] J. SÁNCHEZ CUADRADO, F. JOUAULT, J. GARCIA-MOLINA, J. BÉZIVIN. *Deriving OCL Optimization Patterns from Benchmarks*, in "Proceedings of the 8th OCL Workshop", vol. 15, Electronic Communications of the EASST, 2008.

[49] J. SÁNCHEZ CUADRADO, F. JOUAULT, J. GARCIA-MOLINA, J. BÉZIVIN. *Optimization patterns for OCL-based model transformations*, in "Satellite Events at the MODELS 2008 Conference: MODELS 2008 International Workshops, Toulouse, France, September 28 - October 3, 2008, Revised Selected Papers, LNCS", Springer Berlin / Heidelberg, 2008, p. 273–284.

[50] J. SÁNCHEZ CUADRADO, F. JOUAULT, J. GARCIA-MOLINA, J. BÉZIVIN. *Patrones de optimización de rendimiento para transformaciones de modelos*, in "Proceedings of V Taller sobre Desarrollo de Software Dirigido por Modelos (DSDM'08)", 2008, p. 32–41.

[51] T. ZHANG, F. JOUAULT, J. BÉZIVIN, X. LI. *An MDE-based method for bridging different design notations*, in "Innovations in Systems and Software Engineering", vol. 4, n° 3, 2008, p. 203–213.

[52] T. ZHANG, F. JOUAULT, J. BÉZIVIN, J. ZHAO. *A MDE Based Approach for Bridging Formal Models*, in "Proceedings of the Second IEEE/IFIP International Symposium on Theoretical Aspects of Software Engineering, TASE 2008, June 17-19, 2008, Nanjing, China", IEEE Computer Society, 2008, p. 113–116.