Activity Report 2015

Team ATLANMODELS

Modeling Technologies for Software Production, Operation, and Evolution

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Distributed Systems and middleware

Table of contents

Team ATLANMODELS

Creation of the Team: 2015 May 01

Keywords:

Computer Science and Digital Science:

2.1.10. - Domain-specific languages
2.4.2. - Verification
2.5. - Software engineering
3.1.1. - Modeling, representation

Other Research Topics and Application Domains:

6.1. - Software industry
6.1.1. - Software engineering
6.1.2. - Software evolution, maintenance
6.5. - Information systems

1. Members

Faculty Members

Gerson Sunye [Team leader, Univ. Nantes, Associate Professor, HdR]
Salvador Martinez Perez [EMN, Associate Professor]
Jean-Marie Mottu [Univ. Nantes, Associate Professor]
Massimo Tisi [EMN, Associate Professor]
Jordi Cabot [EMN, Associate Professor, until May 2015]

Engineers

Robin Boncorps [EMN, until Oct 2015]
Hugo Bruneliere [EMN]
Valerio Cosentino [Inria]
Jokin Garcia Perez [ARMINES]
Zied Saidi [Inria]
Javier Canovas Izquierdo [Inria]
Hamza Ed Douibi [ARMINES]

PhD Students

Michel Albonico [Univ Parana, Brazil]
Amine Benelallam [ARMINES]
Gwendal Daniel [Inria]

Post-Doctoral Fellow

Abel Gomez Llana [Inria, until Aug 2015]

Administrative Assistant

Anne Claire Binetruy [Inria]

2. Overall Objectives

2.1. Presentation

Model Driven Engineering (MDE) is a software engineering paradigm that advocates for the rigorous use of (software) models and model manipulation operations (known as model transformations) as the main artifacts in all software engineering activities. 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 when appropriate. The basic assumption of MDE is that models provide a better abstraction level than the classical programming code to manage the complexity of software development (and, in general, any other software-related task). When needed, executable code can be semi-automatically generated from (low-level) models of the system.

AtlanMod focuses on developing pioneering solutions to solve core research challenges in MDE and to ensure its successful application on relevant industrial problems. In particular, AtlanMod is focusing on three key challenges: evaluating the correctness of models and model transformations, ensuring the scalability of modeling techniques to deal with very large models and developing software comprehension and modernization methods by means of the application of model-driven reverse engineering techniques on running software.

2.2. Previous Achievements

AtlanMod has significantly contributed to the evolution of MDE and to the progressive emergence of a scientific community in this field. The team developed a complete modeling framework [40], [48] providing core MDE components for (meta)model definition and manipulation.

The iterative definition of this conceptual framework has been validated by the construction of an MDE toolbox based on the conclusion that MDE is in fact a branch of language engineering. Models can be formally defined by means of metamodels, considered as the definition of the abstract syntax of a language in the same way grammars are used to define programming languages. Modeling languages are then the combination of a metamodel (abstract syntax), a notation (concrete syntax) plus a definition of the language semantics. In this sense, our toolbox can be regarded as a kind of language workbench offering the building blocks to define and manipulate models and metamodels. All these mutually interrelated tools are available under *Eclipse.org* (projects or components: M2M, ATL, AM3, AMW). They are currently in use in research, teaching, and industry and they have a broad user community.

Beyond the development of core MDE components, AtlanMod has also expressed a strong interest in the application and adaptation of these building blocks for specially relevant and challenging industrial problems. As an example, in this context, AtlanMod has been leading the MoDisco project [1] to build reverse engineering solutions for legacy systems.

3. Research Program

3.1. MDE Foundations

Traditionally, models were often used as initial design sketches mainly aimed for communicating ideas among developers. On the contrary, MDE promotes models as the primary artifacts that drive all software engineering activities (i.e. not only software development but also evolution, reverse engineering, interoperability and so on) and are considered as the unifying concept [40]. Therefore, rigorous techniques for model definition and manipulation are the basis of any MDE framework.

[1] http://eclipse.org/MoDisco/

The MDE community distinguishes three levels of models: (terminal) model, metamodel, and metametamodel. A terminal model is a (partial) representation of a system/domain that captures some of its characteristics (different models can provide different knowledge views on the domain and be combined later on to provide a global view). In MDE we are interested in terminal models expressed in precise modeling languages. The abstract syntax of a language, when expressed itself as a model, is called a metamodel. A complete language definition is given by an abstract syntax (a metamodel), one or more concrete syntaxes (the graphical or textual syntaxes that designers use to express models in that language) plus one or more definition of its semantics. The relation between a model expressed in a language and the metamodel of that language is called *conformsTo*. Metamodels are in turn expressed in a modeling language called metamodeling language. Similar to the model/metamodel relationship, the abstract syntax of a metamodeling language is called a metametamodel and metamodels defined using a given metamodeling language must conform to its 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 [47] and [41]. MDE promotes *unification by models*, like object technology proposed in the eighties *unification by objects* [38]. These MDE principles may be implemented in several standards. For example, OMG proposes a standard metametamodel called Meta Object Facility (MOF) while the most popular example of metamodel in the context of OMG standards is the UML metamodel.

In our view the main way to automate MDE is by providing model manipulation facilities in the form of model transformation operations that taking one or more models as input generate one or more models as output (where input and output models are not necessarily conforming to the same metamodel). More specifically, a model transformation Mt defines the production of a model Mb from a model Ma. When the source and target metamodels (MMs) 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 accessor methods for each transformed attribute. Many other operations may be considered as transformations as well. For example verifications or measurements on a model can be expressed as transformations [43]. One can see then why large libraries of reusable modeling artifacts (mainly metamodels and transformations) will be needed.

Another important idea is the fact that a model transformation is itself a model [39]. This means that the transformation program Mt can be expressed as a model and as such conforms to a metamodel MMt. This allows an homogeneous treatment of all kinds of terminal models, including transformations. Mt can be manipulated using the same existing MDE techniques already developed for other kinds of models. For instance, it is possible to apply a model transformation Mt' to manipulate Mt models. In that case, we say that Mt' is a higher order transformation (HOT), i.e. a transformation taking other transformations (expressed as transformation models) as input or/and producing other transformations as output.

As MDE developed, it became apparent that this was a branch of language engineering [42]. In particular, MDE offers an improved way to develop DSLs (Domain-Specific Languages). DSLs are programming or modeling languages that are tailored to solve specific kinds of problems in contrast with General Purpose Languages (GPLs) that aim to handle any kind of problem. Java is an example of a programming GPL and UML an example of a modeling GPL. DSLs are already widely used for certain kinds of programming; probably the best-known example is SQL, a language specifically designed for the manipulation of relational data in databases. The main benefit of DSLs is that they allow everybody to write programs/models 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 the expression of simple computations 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. Similar benefits are obtained when using modeling DSLs. In MDE, new DSLs can be easily specified by using the metamodel concept to define their abstract syntax. Models specified with those DSLs can then be manipulated by means of model transformations (with ATL for example [46]).

When following the previously described principles, one may take advantage of the uniformity of the MDE organization. As an example, considering similarly models of the static architecture and models of the dynamic behavior of a system allows at the same time economy of concepts and economy of implementation.

The following sections describe the main MDE research challenges the team is addressing. They go beyond the development of core MDE techniques (topic on which the team, as mentioned above, has largely contributed in the past, and that we believe is quite well-covered already) and focus on new aspects that are critical for the successful application of MDE in industrial contexts.

3.2. Reverse Engineering

One important domain that is being investigated by the AtlanMod team is the reverse engineering of existing IT systems. We do believe that efficiently dealing with such legacy systems is one of the main challenges in Software Engineering and related industry today. Having a better understanding of these systems in order to document, maintain, improve or migrate them is thus a key requirement for both academic and industrial actors in this area. However, it is not an easy task and it still raises interesting challenging issues to be explored [44].

We have shown how reverse engineering practices may be advantageously revisited with the help of the MDE approach and techniques, applying (as base principle) the systematic representation as models of the required information discovered from the legacy software artifacts (e.g. source code, configuration files, documentation, metadata, etc). The rise in abstraction allowed by MDE can bring new hopes that reverse engineering is now able to move beyond more traditional ad-hoc practices. For instance, a industrial PhD in partnership with IBM France aimed to investigate the possibilities of conceptualizing a generic framework enabling the extraction of business rules from a legacy application, as much as possible, independently of the language used to code it. Moreover, different pragmatic solutions for improving the overall scalability when dealing with large-scale legacy systems (handling huge data volumes) are intensively studied by the team.

In this context, AtlandMod has set up within the past years and is still developing the open source Eclipse MoDisco project (see 6.14). MoDisco is notably being referenced by the OMG ADM (Architecture Driven Modernization) normalization task force as the reference implementation for several of its standard metamodels. It is also used practically and improved in various collaborative projects the team is currently involved in (e.g. FP7 ARTIST). Complementary to the work based on MoDisco, we have also been experimenting (still in an industrial context, cf. TEAP FUI project) on the related problem of data federation from heterogeneous sources in the domain of Enterprise Architecture. This has notably resulted in a prototype called EMF Views that can be practically used in such reverse engineering scenarios.

Reverse engineering techniques have also been used in the context of the Web. In the last years the development of Web APIs has become a discipline that companies have to master to succeed in the Web. The so-called API economy requires, on the one hand, companies to provide access to their data by means of Web APIs and, on the other hand, web developers to study and integrate such APIs into their applications. The exchange of data with these APIs is usually performed by using JSON, a schemaless data format easy for computers to parse and use. While JSON data is easy to read, its structure is implicit, thus entailing serious problems when integrating APIs coming from different vendors. Web developers have therefore to understand the domain behind each API and study how they can be composed. We tackle this problem by developing a MDE-based process able to reverse engineer the domain of Web APIs and to identify composition links among them. The approach therefore allows developers to easily visualize what is behind the API and the connections points that may be used in their applications.

We have recently opened a new research line in the context software analysis, in particular, in the Open-Source Software (OSS) field. The development of OSS follows a collaborative model where any developer can contribute to the advance of the project. To enable this collaboration, OSS projects use a plethora of tools such as forums, issue-trackers and Q&A websites, that developers can adopt to coordinate each other in the development process. Such a collaboration environment includes adapted solutions and provides effective communication means, but also causes scattering of the collaboration data, which hamper the understanding of the whole development process (e.g., who is leading the development or making the decisions). In this context,

we propose to use reverse engineering techniques to better understand how OSS projects are developed in a broad sense, thus taking into account the different collaboration tools used and how they influence in the development of OSS projects.

3.3. Security Engineering

Several components are required to build up a system security architecture, such as firewalls, database user access control, intrusion detection systems, and VPN (Virtual Private Network) routers. These components must be properly configured to provide an appropriate degree of security to the system. The configuration process is highly complex and error-prone. In most organizations, security components are either manually configured based on security administrators expertise and flair; or simply recycled from existing configurations already deployed in other systems (even if they may not be appropriated for the current one). These practices put at risk the security of the whole organization.

As a first step we intend to apply model-driven techniques for the extraction of high level model representations of security policies enforced by system components like networks of firewalls, RDBMS and CMSs. Firewalls, core components in network security systems, are generally configured by using very low level vendor specific rule-based languages, difficult to understand and to maintain. As a consequence, as the configuration files grow, understanding which security policy is being actually enforced or checking if inconsistencies has been introduced becomes a very complex and time consuming task. Similarly, in RDBMSs and CMSs policies are configured and stored by using different, often low-level, mechanisms.

We propose to raise the level of abstraction so that the user can deal directly with the high level policies. Once a model representation of the enforced policy is available, model-driven techniques will ease some of the tasks we need to perform, like consistency checking, validation, querying and visualization. Easy migration between different vendors will be also enabled.

As a further step we intend to apply model-driven techniques for the integration of the diverse security policies extracted from concrete system components. In the case of complex systems composed of a number of interacting heterogeneous subsystems, access-control is pervasive with respect to their architecture. As mentioned above, we can find access-control enforcement rules in different components placed at different architectural levels where rules in a component may impact the execution of the security rules of another component. In addition, the access-control techniques implemented in each component may follow different AC models in order to best suit the needs of the component. Thus, ideally, a global representation of the access-control policy of the whole system should be available, as analysing a component policy in isolation does not provide enough information. Unfortunately, most times this global policy is not explicit or is outdated. This step requires to unveil the implicit dependencies between the set of policies working in an encompassing system, so that a model representing the global AC policy can be built and the global analysis of the AC security is enabled

3.4. Software Quality

As with any type of production, an essential part of software production is determining the quality of the software. The level of quality associated to a software product is inevitably tied to properties such as how well it was developed and how useful it is to its users. AtlanMod team focus on researching techniques for the formal verification and testing of software models and model transformations.

These techniques must be applied at the model level (to evaluate the quality of specific software designs) and at the metamodel level (to evaluate the quality of modeling languages). In both cases, the Object Constraint Language (OCL) of the OMG is widely accepted as a standard textual language to complement (meta)model specifications with all those rules/constraints that cannot be easily defined using graphical modeling constructs.

Among all possible properties to verify, we take as the basic property the *satisfiability* property, from which many others may be derived (as liveliness, redundancy, subsumption,...). Satisfiability checks whether it is possible to create a valid instantiation (i.e. one that respects all modeling constraints) of a give (meta)model. Satisfiability is an undedicable problem when general OCL constraints are used as part of the model definition.

To deal with this problem, the team maintains the tool EMFtoCSP which translates the model verification challenge into the domain of constraint logic programming (CLP) for which sophisticated decision procedures exist. The tool integrates the described functionality in the Eclipse Modeling Framework (EMF) and the Eclipse Modeling Tools (MDT), making the functionality available for MDE in practice.

To complement these formal verification techniques we are also working on testing techniques, specially to optimize the testing of model transformations. White-box testing for model transformations is a technique that involves the extraction of knowledge embedded in the transformation code to generate test models. In our work, we apply static analysis techniques to model transformation specifications and represent the extracted knowledge as partial models that can drive the generation of highly effective test models (specially in terms of coverage).

3.5. Collaborative Development

Software development processes are collaborative in nature. The active participation of end-users in the early phases of the software development life-cycle is key when developing software. Among other benefits, the collaboration promotes a continual validation of the software to be build, thus guaranteeing that the final software will satisfy the users' needs. In this context, we have opened two novel research lines focused on the collaborative development *in* MDE and the collaborative development *with* MDE. The former is aimed at promoting the collaboration in the context of MDE while the latter uses MDE techniques to promote the participation in software development processes.

Collaboration is important in the context of MDE, in particular, when creating Domain-Specific Modeling Languages (DSMLs) which are (modeling) languages specifically designed to carry out the tasks of a particular domain. While end-users are actually the experts of the domain for which a DSML is developed, their participation in the DSML specification process is still rather limited nowadays (they are normally only involved in providing domain knowledge or testing the resulting language). This means that the MDE technical experts and not end-users are the ones in control of the DSML construction and evolution. This is a problem because errors in understanding the domain may hamper the development process and the quality of the resulting DSML. Thus, it would be beneficial to promote a more active participation of end-users in the DSML development process.

We have been working on the required support to make effective this participation, in particular, we have developed Collaboro, an approach which enables the involvement of the community (i.e., end-users and developers) in the DSML creation process. Collaboro allows modeling the collaborations between community members taking place during the definition of a new DSML and supports both the collaborative definition of the abstract (i.e., metamodel) and concrete (i.e., notation) syntaxes for DSMLs by providing specific constructs to enable the discussion. Thus, each community member will have the chance to request changes, propose solutions and give an opinion (and vote) about those from others. We believe this discussion will enrich the language definition significantly and ensure that the end result satisfies as much as possible the expectations of the end-users. Collaboro has also been extended to support the example-driven development of DSMLs, thus promoting the engagement of end-users in the process.

The lessons learnt from this MDE-focused collaboration research are now being applied to the more general context of software development. In particular, our interest is to study how software development processes are governed (i.e. how the collaboration among developers and user takes place). Any software development project has to cope with a huge number of tasks consisting of either implementing new issues or fixing bugs. Thus, effective and precise prioritization of these tasks is key for the success of the project. Governance rules enable the coordination of developers in order to advance the project. Despite their importance, in practice governance rules are hardly ever explicitly defined, specially in the context of Open Source Systems (OSS), where it is hard to find a explicit system-level design, a project plan, schedule or list of deliverables. To alleviate this situation, mechanisms to facilitate the communication and the assignment of work are considered crucial for the success of the development. Tracking and issue-tracking systems, mailing lists and forums are broadly used to manage the tasks to be performed. While these tools provide a convenient compartmentalization of work and effective means of communication, they fall short in providing adequate support for specifying and

enforcing governance rules (e.g. supporting the voting of tasks, easy tracking of decisions made in the project, etc.).

Thus, we believe the explicit definition of governance rules along with the corresponding infrastructure to help developers follow them would have several benefits, including improvements in the transparency of the decision-making process, traceability (being able to track why a decision was made and who decided it) and the automation of the governance process (e.g. liberating developers from having to be aware and follow the rules manually, minimizing the risk of inconsistent behaviour in the evolution of the project). We resort on MDE techniques to tackle this problem and provide a DSL specially adapted to the domain of governance in software projects to let project managers easily define the governance rules of their projects.

3.6. Scalability

As MDE is increasingly applied to larger and more complex industrial applications, the current generation of modelling and model management technologies are being stressed to their limits in terms of their capacity to accommodate collaborative development, efficient management and persistence of models larger than a few hundreds of megabytes in size. Additional research and development is imperative in order to enable MDE to remain relevant with industrial practice and to continue delivering its widely recognised productivity, quality, and maintainability benefits. Achieving scalability in modelling and MDE involves being able to construct large models and domain-specific languages in a systematic manner, enabling teams of modellers to construct and refine large models in a collaborative manner, advancing the state-of-the-art in model querying and transformations tools so that they can cope with large models (of the scale of millions of model elements), and providing an infrastructure for efficient storage, indexing and retrieval of large models. AtlanMod wants to provide a solution for these aspects of scalability in MDE by extending the Eclipse modeling framework, to create an open-source solution to scalable modeling in industry.

3.7. Industrialization of open source tools

Research labs, as a source of innovation, are potential key actors of the Software Engineering market. However, an important collaborative effort with the other players in the software industry is still needed in order to actually transfer the corresponding techniques or technologies from the research lab to a company. Based on the AtlanMod concrete experience with the previously mentioned open source tools/projects, we have extracted a pragmatic approach [3] for transforming the results of scientific experimentation into practical industrial solutions.

While dealing with innovation, this approach is also innovation-driven itself, as the action is actually conducted by the research lab via a technology transfer. Three different partners are directly involved in this process, using open source as the medium for maintaining a constant interaction between all of them:

- **Use Case Provider.** Usually a company big enough to have to face real complex industrial scenarios which need to be solved (at least partially) by applying new innovative principles and techniques;
- **Research Lab.** Usually a group from a research institute (public or private) or university evaluating the scientific relevance of the problems, identifying the research challenges and prototyping possible solutions;
- **Technology Provider.** Usually a small or medium company, with a particular technical expertise on the given domain or Software Engineering field, building and delivering the industrial version of the designed solutions;

From our past and current experience, three main characteristics of this industrialization *business model* can be highlighted:

- **Win-win situation.** Each partner can actually focus on its core activity while also directly benefiting from the results obtained by the others (notably the research lab can continue to do research);
- **Application-driven context.** The end-user need is at the origin of the process, which finally makes the developed solution actually relevant;
- **Iterative process.** The fact of having three distinct partners requires different regular and consecutive exchanges between all of them.

4. Application Domains

4.1. Application Domain

By definition, MDE can be applied to any software domain. Core MDE techniques developed by the team have been successfully applied to a large variety of industrial domains from information systems to embedded systems. MDE is not even restricted to software engineering, but also applies to data engineering [45] and to system engineering [37]. There are a lot of problems in these application domains that may be addressed by means of modeling and model transformation techniques.

As a result, AtlanMod has collaborated with a great variety of different companies ranging from the Automotive to the Insurances domains and from SMEs to large enterprises through the projects described later on in this same report. AtlanMod hopes to continue this trend in the future.

5. Highlights of the Year

5.1. Highlights of the Year

"Software Modernization Revisited: Challenges and Prospects" appears in IEEE Computer Magazine. Based on our past and present experience in software migration projects, this article puts the focus on some important factors/challenges to take into consideration when dealing with such projects and propose corresponding recommendations to maximize the chance of success. In this respect, it notably presents some concrete findings we have made while collaborating with our partners during the 3 years of the ARTIST EU project.

6. New Software and Platforms

6.1. AM3

AtlanMod MegaModel Management
KEYWORDS: Modeling artifact - MDA - MDE - Megamodeling
FUNCTIONAL DESCRIPTION

AM3 (AtlanMod MegaModel Management) is a generic and extensible tool/framework dedicated to global model management, i.e., the management of different modeling artifacts as well as their interrelationships.

- Participant: Hugo Brunelière
- Partner: Ecole des Mines de Nantes
- Contact: Hugo Brunelière
- URL: http://wiki.eclipse.org/AM3

6.2. AMW

Atlas Model Weaver
KEYWORDS: MDA - MDE - Weaving - Link - Model element
FUNCTIONAL DESCRIPTION

AMW is a generic and extensible tool dedicated to the creation and handling of weavings between models, these weavings representing sets of links existing between elements potentially coming from different models.
Language

- Participant: Frédéric Jouault
- Contact: Hugo Brunelière
- URL: http://wiki.eclipse.org/AMW

6.3. ATL

Atlanmod Transformation Language
KEYWORDS: MDA - MDE - ATL - QVT - Model - Transformation
FUNCTIONAL DESCRIPTION

ATL is a tool dedicated to model transformation, complete, integrated into Eclipse and including its own development and execution environment, concrete examples as well as the corresponding associated documentation.

- Participants: Jean Bézivin, Frédéric Jouault and Patrick Valduriez
- Partner: Ecole des Mines de Nantes
- Contact: Hugo Brunelière
- URL: http://www.eclipse.org/m2m/atl/

6.4. ATL-MR

ATL-MapReduce
KEYWORDS: Transformation - Distributed
SCIENTIFIC DESCRIPTION

ATL-MapReduce

ATL-MapReduce (ATL-MR) is a prototype tool for running complex ATL transformation in the cloud using Hadoop MapReduce. ATL-MapReduce is implemented on top of an extended ATL VM that can be found on (https://github.com/atlanmod/org.eclipse.atl.atlMR/tree/master). Coupling ATL-MR with the [the extended VM](https://github.com/atlanmod/org.eclipse.atl.atlMR/tree/master) has proved a good performance, especially in terms of execution time. [In our experiments](http://www.emn.fr/z-info/atlanmod/index.php/Image:Atlmr-experiments-raw-data.zip), ATL-MR runs up to 6x faster compared to the regular VM while distributing it over 8 machines.

- Participants: Amine Benelallam, Abel Gomez Llana and Massimo Tisi
- Contact: Amine Benelallam
- URL: https://github.com/atlanmod/ATL_MR

6.5. ATLAS model weaver

ATLAS Model Weaver
KEYWORDS: Model - Metamodel - Weaving model
SCIENTIFIC DESCRIPTION

AMW is a component-based platform for model weaving that can be used to establish and manage abstract correspondences between models. The platform is generic and 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.
FUNCTIONAL DESCRIPTION

The AMW is a tool for establishing relationships (i.e., links) between models. The links are stored in a model, called weaving model. It is created conforming to a weaving metamodel.

- Participants: Jean Bézivin, Erwan Breton, Marcos Didonet Del Fabro, Guillaume Gueltas, Frédéric Jouault and Patrick Valduriez
- Contact: Frédéric Jouault
- URL: http://www.eclipse.org/gmt/amw/

6.6. AmmA

Atlas Model Management Architecture
FUNCTIONAL DESCRIPTION

The AMMA platform is a model management platform. It is composed of several elements. Three of them: AM3, AMW, and MoDisco are available as GMT components. ATL is a component of M2M project.

- Participants: Mikaël Barbero and Frédéric Jouault
- Contact: Frédéric Jouault
- URL: https://wiki.eclipse.org/AMMA

6.7. Collaboro

SCIENTIFIC DESCRIPTION

Collaboro is an approach to make language development processes more participative, meaning that both developers and users of the language can collaborate together to create and evolve it. Collaboro supports both the collaborative definition of the abstract (i.e., metamodel) and concrete (i.e., notation) syntaxes for your DSL by providing a collaborative enviroment enabling the discussion.

Anyone has the chance to request changes, propose solutions and give an opinion (and vote) about those from others. This discussion enrichs the language definition signicantly and ensures that the end result satisfies as much as possible the expectations of the end-users.

- Participants: Jordi Cabot, Robin Boncorps and Javier Canovas Izquierdo
- Contact: Hugo Brunelière
- URL: http://atlanmod.github.io/collaboro/#/

6.8. EMF Facet

KEYWORDS: MDE - Model - Metamodel - Extension - Dynamicity - Querying
FUNCTIONAL DESCRIPTION

EMF Facet is a tool/framework dedicated to the non-intrusive and dynamic extension of metamodels (adding of new types, attributes, references, etc), based on a model query generic mechanism

- Partners: Ecole des Mines de Nantes - Mia-Software
- Contact: Hugo Brunelière
- URL: http://www.eclipse.org/modeling/emft/facet/

6.9. EMF Views

Eclipse Modeling Framework Views
FUNCTIONAL DESCRIPTION

Some users only need to see some parts of a model, others have to get the full model extended with data from another model, and others simply access to a combination of information coming from different models. Based on the unquestionable success/usefulness of database views to solve similar problems in databases, EMF Views aims to bring the same concept to the modeling world.

- Contact: Hugo Brunelière
- URL: https://github.com/atlanmod/emfviews

6.10. EMF-REST

Eclipse Modeling Framework Rest
FUNCTIONAL DESCRIPTION

EMF is the modeling framework of the Eclipse community. While EMF is able to automatically generate Java APIs from Ecore models, it is still missing support to deal with Web APIs such as RESTful ones that could boost the use of modeling techniques in the Web. However, the creation of RESTful APIs requires from developers not only an investment in implementation but also a good understanding of the REST Principles to apply them correctly. We therefore created EMF-REST, a tool that empowers EMF to get Truly RESTFul APIs from Ecore models, thus allowing web developers to generate JSON-based Web APIs for their applications. It generates both a JavaScript API to work with models as Javascript Objects in the client-side (without any EMF dependency) and REST services in the server-side based on the Java JAX-RS specification.

- Contact: Gerson Sunyé
- URL: http://emf-rest.com/

6.11. EMFtoCSP

SCIENTIFIC DESCRIPTION

Essentially, the EMFtoCSP is a sophisticated bounded model finder that yields instances of the model that conform not only to the structural definition of the model (e.g. the multiplicity constraints), but also to the OCL constraints. Based on this core, several correctness properties can be verified:

Satisfiability – is the model able to express our domain? For this check, the minimal number of instances and links can be specified to ensure non-trivial instances.

Unsatisfiability – is the model unable to express undesirable states? To verify this, we add further constraints to the model that state undesired conditions. Then we can check if is it impossible to instantiate the amended model.

Constraint subsumption – is one constraint already implied by others (and could therefore be removed)?

Constraint redundancy – do different constraints express the same fact (and could therefore be removed)?

FUNCTIONAL DESCRIPTION

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT) provides mature tool support for such OCL-annotated models with respect to model definition, transformation, and validation.

- Contact: Hugo Brunelière
- URL: https://github.com/SOM-Research/EMFtoCSP

6.12. GiLA

GitHub Label Analyzer

FUNCTIONAL DESCRIPTION

Reporting bugs, asking for new features and in general giving any kind of feedback is the easiest way to contribute to an Open-Source Software (OSS) project. In GitHub, the largest code hosting service for OSS, this feedback is typically expressed as new issues for the project managed by an issue-tracking system available in each new project repository. Among other features, the issue tracker allows creating and assigning labels to issues with the goal of helping the project community to better classify and manage those issues (e.g., facilitating the identification of issues for top priority components or candidate developers that could solve them). Nevertheless, as the project grows a manual browsing of the project issues is no longer feasible.

- Contact: Gerson Sunyé

6.13. JSON Discoverer

FUNCTIONAL DESCRIPTION

The JSON discoverer allows you to discover the implicit schema of your JSON documents. Any JSON document includes both metadata (i.e., the schema) and data (i.e., the objects/values conforming to the schema). Given a (set of) JSON documents our discoverer analyzes the JSON definitions and generates for you a class diagram showing graphically the implicit JSON schema of your documents plus an object diagram representing their data.

- Participants: Jordi Cabot and Javier Canovas Izquierdo
- Contact: Hugo Brunelière
- URL: http://atlanmod.github.io/json-discoverer/#/

6.14. MODISCO

analyser, refactorer et comprendre le legacy
KEYWORDS: MDA - MDE - Model - Modernisation - Discoverer
SCIENTIFIC DESCRIPTION

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g. Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).

The MoDisco framework is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its Infrastructure layer, MoDisco offers the set of generic (i.e., legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g. views definition), etc. As part of its Technologies layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g. Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g. Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

The initiative continues to be developed within the context of the European FP7-ICT project named ARTIST (http://www.artist-project.eu/), and also to a lower extent within the context of the French FUI 13 project named TEAP.
FUNCTIONAL DESCRIPTION

MoDisco is an Eclipse generic and extensible tool/framework providing to developers a set of reusable base components and dedicated to the elaboration of model driven reverse engineering (MDRE) solutions.

- Participant: Hugo Brunelière
- Partners: Ecole des Mines de Nantes - Mia-Software
- Contact: Hugo Brunelière
- URL: http://www.eclipse.org/MoDisco/

6.15. NeoEMF

FUNCTIONAL DESCRIPTION

NeoEMF is an open source software distributed under the terms of the Eclipse Public License that provides a backend-agnostic persistence solution for big, complex and highly interconnected EMF models. NeoEMF is a model repository and persistence framework allowing on-demand loading, storage, and unloading of large-scale EMF models.

- Partner: Mia-Software
- Contact: Gerson Sunyé
- URL: http://www.neoemf.com/

6.16. Reactive-ATL

KEYWORDS: Transformation - Reactive
SCIENTIFIC DESCRIPTION

Reactive engine for the ATL transformation language. In a model-driven application environment it works by activating only the strictly needed computation in response to updates or requests of model elements. Computation is updated when necessary, in an autonomous and optimized way by using incrementality and lazy evaluation.

- Participants: Salvador Martinez Perez and Massimo Tisi
- Contact: Salvador Martinez Perez
- URL: http://atlanmod.github.io/org.eclipse.atl.reactive

7. New Results

7.1. Reverse Engineering & Evolution

Model Driven Reverse Engineering (MDRE), with its applications on software modernization or tool evolution for example, is a discipline in which model-based principles and techniques are used to treat various kinds of (sometimes very large) existing systems. In the continuity the work started several years ago, AtlanMod has been working actively on this research area this year again. The main contributions are the following:

- In the context of the ARTIST FP7 project, the work has been continued on reusing (and extending accordingly) MoDisco and several of its components to provide the Reverse Engineering support required within the project (and more particularly in the context of the use cases provided by our industrial partners). This has been an important year for the team in this project since it successfully ended in November 2015 after final review at the European Commission. At conceptual-level, the proposed overall approach (as a main result of the ARTIST project) and the main lessons we have learned from its application to concrete industrial scenarios have been published and promoted to a large and high-level audience [11]. The ARTIST project in itself, the various research aspects it addresses and the offered technical solutions have also been presented to the Modeling community [22]. At tooling-level, several (MoDisco-based) model discovery components from Java and SQL

have been enhanced while made available as part of a second version of the official ARTIST OS Release [2]. A promising work has also been started on studying deeper the automated discovery of behavioral aspects of software applications, notably by working on a pragmatic mapping between a programming language (Java) and a modeling language (the OMG fUML standard) that focuses on executable aspects.

- To facilitate the understanding of existing software applications via the different models describing them, a significant work has been performed related to providing a generic support for dealing with viewpoints and views expressed on set of possibly heterogeneous and large models. To this intent, and directly capitalizing on the work performed in the TEAP FUI project that ended by the end of 2014, the EMF Views prototype has been significantly refined and enhanced with a ViewPoint Definition Language (the VPDL domain-specific language having a SQL-like syntax) notably [18]. Based on this same model viewpoints/views approach, and more particularly on its underlying (meta)model virtualization support, the general problem of lightweight (meta)model extension has been studied more deeply in the context of our work within the MoNoGe FUI project (national). This has already resulted in a corresponding prototype and a DSL for expressing metamodel extensions [17]. Within the coming year, the plan is to continue further this global work on model viewpoints/views in a software understanding and evolution context.
- Software development projects are notoriously complex and difficult to deal with. Several support tools have been introduced in the past decades to ease the development activities such as issue tracking, code review and Source Control Management (SCM) systems. While such tools efficiently track the evolution of a given aspect of the project (e.g., issues or code), they provide just a partial view of the software project and they often lack of querying mechanisms beyond basic support (e.g., command line, simple gui). This is particularly true for projects that rely on Git, one of the most popular SCM systems. Nowadays many tools are built on top of it, however, they do not complement Git with query functionalities and currently none of them proposes a mechanism that unifies the project information scattered in such different tools. In [28], we propose a conceptual schema for Git and an approach that, given a Git repository, exports its data to a relational database in order to (1) promote data integration with other existing Git-based tools relying on databases and (2) provide query functionalities expressed through standard SQL syntax. To ensure efficiency, our approach comes with an incremental propagation mechanism that refreshes the database content with the latest modifications.

7.2. MDE Scalability

The increasing number of companies embracing MDE methods and tools have exceeded the limits of the current model-based technologies, presenting scalability issues while facing the growing complexity of their data. Since further research and development is imperative in order to maintain MDE techniques as relevant as they are in less complex contexts, we have focused our research in three axes, (i) scalable persistence solutions, (ii) scalable model transformation engines, and (iii) testing of large scale distributed systems.

In [33], we introduce and evaluate a map-based persistence model for MDE tools. We use this model to build a transparent persistence layer for modeling tools, on top of a map-based database engine. The layer can be plugged into the Eclipse Modeling Framework, lowering execution times and memory consumption levels of other existing approaches. Empirical tests are performed based on a typical industrial scenario, model-driven reverse engineering, where very large software models originate from the analysis of massive code bases. The layer is freely distributed and can be immediately used for enhancing the scalability of any existing Eclipse Modeling tool. We learned that—in terms of performance—typical model-access APIs, with fine-grained methods that only allow for one-step-navigation queries, do not benefit from complex relational or graph-based data structures. Much better results are potentially obtained by optimized low-level data structures, like hash-tables, which guarantee low and constant access times. Additional features that may be of interest in scenarios where performance is not an issue (such as versioning and transactional support provided by CDO) have

[2] http://www.artist-project.eu/tools-of-toolbox/193

not been considered. In [32] we extend our persistent mechanism to distributed environments by presenting NeoEMF/HBase, a model-persistence backend for the Eclipse Modeling Framework (EMF) built on top of the Apache HBase data store. Model distribution is hidden from client applications, that are transparently provided with the model elements they navigate. Access to remote model elements is decentralized, avoiding the bottleneck of a single access point. The persistence model is based on key-value stores that allow for efficient on-demand model persistence.

Once we develop a high-performance and distributed persistence mechanism for very-large models, we can exploit it to run high-performance computing over such models. One of the central operations in MDE is rule-based model transformation (MT). It is used to specify manipulation operations over structured data coming in the form of model graphs. However, being based on computationally expensive operations like subgraph isomorphism, MT tools are facing issues on both memory occupancy and execution time while dealing with the increasing model size and complexity. One way to overcome these issues is to exploit the wide availability of distributed clusters in the Cloud for the distributed execution of MT. In [24] and [23], we propose an approach to automatically distribute the execution of model transformations written in a popular MT language, ATL, on top of a well-known distributed programming model, MapReduce. We show how the execution semantics of ATL can be aligned with the MapReduce computation model. We describe the extensions to the ATL transformation engine to enable distribution, and we experimentally demonstrate the scalability of this solution in a reverse-engineering scenario.

Another fundamental operation in MDE is model querying. The Object Constraint Language (OCL) is the standard query language proposed by OMG and is a central component in other modeling and transformation languages such as the Unified Modeling Language (UML), the Meta Object Facility (MOF), and Query View Transformation (QVT). OCL is standardized as a strict functional language. In [34], we propose a lazy evaluation strategy for OCL. We argue that a lazy evaluation semantics is beneficial in some model-driven engineering scenarios for: i) lowering evaluation times on very large models; ii) simplifying expressions on models by using infinite data structures (e.g., infinite models); iii) increasing the reusability of OCL libraries. We implement the approach on the ATL virtual machine EMFTVM.

Finally an important class of operations in MDE is bidirectional (i.e. reversible) computation. Especially bidirectional model transformation is a key technology when two models that can change over time have to be kept constantly consistent with each other. In Hidaka et al. we clarify and visualize the space of design choices for bidirectional transformations from an MDE point of view, in the form of a feature model. The selected list of existing approaches are characterized by mapping them to the feature model. Then the feature model is used to highlight some unexplored research lines in bidirectional transformations, especially in the scalability of such systems.

7.3. Software Quality

We initiated a new line of research in order to investigate Novelty Search (NS) for the automatic generation of test data, in collaboration with the DiverSE team. Our goal is to explore the huge space of test data within the input domain. In this approach, we select test data based on a novelty score showing how different they are compared to all other solutions evaluated so far [25], [26].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Grants with Industry

Orange Labs (Cesson-Sévigné) is founding a PhD Thesis (CIFRE) on the topic of trust modeling on Web-RTC communications.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. FUI

Program: FUI - AAP 15

Project acronym: MoNoGe

Project title: Atelier de Modélisation de Nouvelle Génération

Duration: 2013 - 2016

Coordinator: Softeam

Other partners: Industry (DCNS), Research and University (ARMINES AtlanMod, LIP6) and Vendors and service providers (Softeam, Soft-Maint, Mia-Software)

Abstract: There is currently in companies a wide diversity of models and modeling tools according to the application domains, services or contexts which are concerned. This implies different problems forbidding their plain exploitation: traceability, global coherence, continuity between works, knowledge management, etc. All are largely penalized by this situation that harms the mastering of the complexity of the related systems and software. The MoNoGe project has for objective to bring innovative solutions allowing to ensure the agility of the models and modeling tools. The term agility is here referring to the properties of interoperability, extensibility and evolution of models. The dynamic extension mechanism to be developed in MoNoGe, potentially inspiring from the OMG MEF standard currently under definition, is intended to preserve the original metamodel which can be conserved, partially hidden or extended. Thus, the legacy data and models can stay operational with the extended metamodel. The user does not have to deal with heavy migration or conversion operations, and can this way focus on its modeling activities while continuously exploiting past models. Our focus within the project is on defining conceptually such a (meta)model extension solution and proposing an implementing prototype based on Eclipse/EMF. To this intent, we are already studying the potential reuse (and improvement) of our EMF Views prototype in this given context.

Program: FUI - AAP 13

Project acronym: TEAP

Project title: TOGAF Entreprise Architecture Platform

Duration: 2012 - 2014

Coordinator: Obeo

Other partners: Industry (DCNS), Research and University (Inria AtlanMod) and Vendors and service providers (Obeo, Capgemini)

Abstract: The fast evolution of technologies (SOA, Cloud, mobile environments), the systems complexity and the growing need for agility require to be able to represent information systems as a whole. The high-level approach promoted by Enterprise Architecture (EA) is a key element in this context and intends to address all the systems dimensions: software components, associated physical resources, relationships with the companies requirements and business processes, implied actors/roles/structures, etc. The objective of the TEAP project was to specify and implement an EA platform based on the Open Group international standard named TOGAF and on the SmartEA technical solution. In addition to its base modeling capabilities, this platform now allows data federation from different existing sources (e.g. for reverse engineering purposes such as retro-cartography) as well as the definition of possible transformation chains (for governance and modernization). As part of this project, we have been notably using in practice (and improving accordingly) some of our works and corresponding prototypes such as EMF Views, ATL or some MoDisco components.

Program: FUI - AAP 13

Project acronym: ITM Factory

Project title: Information Technology Modernisation Factory

Duration: 04/2012 - 10/2014

Coordinator: Soft-Maint (Groupe SODIFRANCE)

Other partners: Mia-Software (Groupe SODIFRANCE), ACAPNOS, MMA and Inria AtlanMod.

Abstract: Application maintenance represents about 80 per cent of the computer market (at the French and global level). The challenge of software maintenance is to keep running applications with technologies that are no longer required to be maintained and with changing development teams and whose skills are not always validated on ancient languages. The main goal of the ITM Factory is to propose a software modernization framework, based on the ModDisco project and including: (i) an integrated workbench for software modernization engineers and (ii) a set of ready to use modernization cartirdges, i.e., a solution brick that meets a business challenge level, as opposed to a technical bricks that provides technical solutions that are integrated into a business solution.

9.2. European Initiatives

9.2.1. FP7 & H2020 Projects

Program: FP7 - COOPERATION (ICT)

Project acronym: MONDO

Project title: Scalable Modelling and Model Management on the Cloud

Duration: November 2013 - May 2016

Coordinator: The Open Group - X/Open Company

Other partners: The Open Group - X/Open Company (United Kingdom), University of York (United Kingdom), Universidad Autonoma de Madrid (Spain), Budapest University of Technology and Economics (Hungary), IKERLAN (Spain), Soft-Maint (France), SoftTeam (France), UNINOVA (Portugal)

Abstract: As Model Driven Engineering (MDE) is increasingly applied to larger and more complex systems, the current generation of modelling and model management technologies are being pushed to their limits in terms of capacity and efficiency, and as such, additional research is imperative in order to enable MDE to remain relevant with industrial practice and continue delivering its widely recognised productivity, quality, and maintainability benefits. The aim of MONDO is to tackle the increasingly important challenge of scalability in MDE in a comprehensive manner. Achieving scalability in modelling and MDE involves being able to construct large models and domain specific languages in a systematic manner, enabling teams of modellers to construct and refine large models in a collaborative manner, advancing the state-of-the-art in model querying and transformations tools so that they can cope with large models (of the scale of millions of model elements), and providing an infrastructure for efficient storage, indexing and retrieval of large models. To address these challenges, MONDO brings together partners with a long track record in performing internationally-leading research on software modelling and MDE, and delivering research results in the form of robust, widely-used and sustainable open-source software, with industrial partners active in the fields of reverse engineering and systems integration, and a global consortium including more than 400 organisations from all sectors of IT.

Program: FP7 - Research For SMEs

Project acronym: AutoMobile

Project title: Automated Mobile App Development Type: Research For SMEs

Duration: November 2013 - October 2015

Coordinator: WebRatio s.r.l. (Italy)

Other partners: Politecnico di Milano (Italy), AtlanMod-Armines, Moon Submarine (UK), Forward-Software (Rumania).

Abstract: The AutoMobile project aims at designing and bringing to the market innovative methodologies, software tools, and vertical applications for the cost-effective implementation of cross-platform, multi-device mobile applications, i.e. business applications that can be accessed by users on a variety of devices and operating systems, including PC, cellular / smart phones and tablets. Cross-platform and multi-device design, implementation and deployment is a barrier for today's IT solution providers, especially SME providers, due to the high cost and technical complexity of targeting development to a wide spectrum of devices, which differ in format, interaction paradigm, and software architecture. AutoMobile will exploit the modern paradigm of Model-Driven Engineering and code generation to dramatically simplify multi-device development, reducing substantially cost and development times, so as to increase the profit of SME solution providers and at the same time reduce the price and total cost of ownership for end-customers. AutoMobile will rely on modeling languages such as IFML (Interaction Flow Modeling Languages) and on tools like WebRatio.

9.3. International Initiatives

9.3.1. Inria International Partners

9.3.1.1. Informal International Partners

The six main research partners of the team are:

- Politecnico di Milano (Italy) - DB Group, especially Marco Brambilla
- TU Wien (Austria) - BiG Group, especially Manuel Wimmer
- Politecnica de Catalunya (Spain) - GESSI Group, especially Xavier Franch
- Universitat Polièncnica de Valèncnia (Spain) - ISSI Group, especially José H. Canós
- ICREA (Spain)- SOM Group, especially Jordi Cabot
- National Institute of Informatics in Tokyo - BiG group, especially Soichiro Hidaka

9.4. International Research Visitors

9.4.1. Visits of International Scientists

- In March, Soichiro Hidaka, from the National Institute of Informatics (NII) in Tokyo, Japan, visited AtlanModels for one month, in the frame of a collaboration on bidirectionalization of model-transformation languages.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific events organisation

10.1.1.1. Member of the organizing committees

- M. Tisi has coorganized the Scalable Model Driven Engineering Workshop (BigMDE'15) within the STAF federated conference.

10.1.2. Scientific events selection

10.1.2.1. Member of the conference program committees

- Gerson Sunye:

  - International: International Conference on IT Convergence and Security, International Conference on Software Engineering and Knowledge Engineering, International Workshop on Large-Scale Testing, International Workshop on Testing The Cloud, BigMDE Workshop
- Massimo Tisi:
  - International: International Conference on Software Language Engineering, International Conference on Model Transformation, International Conference on Current Trends in Theory and Practice of Computer Science, Transformation Tool Contest, Model Driven Requirement Engineering Workshop, Executable Modeling Workshop, BigMDE Workshop

10.1.2.2. Reviewer

- Hugo Brunelière
  - Maintenance and Evolution of Service-Oriented Systems and Cloud-Based Environments, MESOCA (ICSME 2015).
- Gerson Sunyé
  - International Conference on Software Engineering, ACM SIGSOFT Symposium on the Foundations of Software Engineering.

10.1.3. Journal

10.1.3.1. Reviewer - Reviewing activities

- Hugo Brunelière
  - Journal of Information and Software Technology (Elsevier), Journal of Software: Evolution and Process (Wiley), International Journal of Software and Systems Modeling (Springer).
- Gerson Sunyé
  - IEEE Transactions on Software Engineering, Journal of Systems and Software (2x), Software and Systems Modeling.
- Massimo Tisi
  - IEEE Transactions of Software Engineering, IEEE Software, Journal of Software: Evolution and Process, Journal of Systems and Software, Science of Computer Programming, Software and Systems Modeling, Software: Practice and Experience, Transactions on Software Engineering, Journal on Web Engineering, Journal of Logic and Algebraic Programming.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

License: M. Tisi, Projet Integrateur PRIME, 12h, L3, Ecole des Mines de Nantes

License: M. Tisi, Projet IPIPIP, 8h, L3, Ecole des Mines de Nantes

Master: M. Tisi, Bases de Données, 30h, M1, Ecole des Mines de Nantes

Master: M. Tisi, Interaction Homme-Machine, 14h, M1, Ecole des Mines de Nantes

Master: M. Tisi, Model-Driven Engineering II, 45h, M2, Ecole des Mines de Nantes, Formation par Apprentissage

Master: M. Tisi, Sensibilisation Recherche, 12h, M2, Ecole des Mines de Nantes, Formation par Apprentissage

Master: G. Sunye, Model-Driven Engineering, 30h, M2, Université de Nantes

Master: G. Sunye, Software Engineering, 48h, M1, Université de Nantes

Master: G. Sunye, Software Testing, 48h, M1, Université de Nantes

Master: G. Sunye, Software Development, 24h, M1, Université de Nantes

License: S. Martínez, Bases de Données, 28h, L3, Ecole des Mines de Nantes

License: S. Martínez, Interaction Homme-Machine, 14h, L3, Ecole des Mines de Nantes

Master: S. Martínez, Master MIAGE - Model Transformations, 6h, Université de Nantes

License: S. Martínez, Domain Specific Languages, 3h, L3, Ecole des Mines de Nantes

10.2.2. Supervision

PhD in progress: Amine Benellanem, Scalability of model transformations, Jordi Cabot, Massimo Tisi and Gerson Sunyé.

PhD in progress: Gwendal Daniel, Efficient storage of large models, Jordi Cabot, Massimo Tisi and Gerson Sunyé.

PhD in progress: Michel Albonico, Model-driven testing of cloud environments, Jordi Cabot and Gerson Sunyé.

PhD in progress: Mohamed Boussaa, An Architecture for Testing Large-Scale Dynamic Distributed Systems, Université de Rennes 1, Benoit Baudry, Olivier Barais, Gerson Sunyé.

PhD in progress: Kevin Corre, Model-based Trust, Université de Rennes 1–Orange, Olivier Barais, Gerson Sunyé, Jean-Michel Crom, Vincent Frey

11. Bibliography

Major publications by the team in recent years

[1] M. ALBERT, J. CABOT, C. GÓMEZ, V. PELECHANO. *Automatic generation of basic behavior schemas from UML class diagrams*, in "Software and System Modeling", 2010, vol. 9, n^{o} 1, pp. 47-67

[2] D. AMELLER, C. AYALA, J. CABOT, X. FRANCH. *Non-functional Requirements in Architectural Decision Making*, in "IEEE Software", March 2013, vol. 30, n^{o} 2, pp. 61-67, http://hal.inria.fr/hal-00822203

[3] H. BRUNELIÈRE. *An Open Source-Based Approach for Industrializing Research Tools*, in "OW2Con 2011, Session "Open Source Innovation Catalyst" - ICT Labs, IRILL", Paris, Issy Les Moulineaux, France, November 2011, http://hal.inria.fr/hal-00639559/en

[4] J. CABOT, R. CLARISÓ, E. GUERRA, J. DE LARA. *Verification and validation of declarative model-to-model transformations through invariants*, in "Journal of Systems and Software", 2010, vol. 83, n^{o} 2, pp. 283-302

[5] J. CABOT, R. CLARISÓ, D. RIERA. *On the verification of UML/OCL class diagrams using constraint programming*, in "Journal of Systems and Software", 2014, vol. 93, pp. 1–23, http://dx.doi.org/10.1016/j.jss.2014.03.023

[6] J. CABOT, R. PAU, R. RAVENTÓS. *From UML/OCL to SBVR specifications: A challenging transformation*, in "Inf. Syst.", 2010, vol. 35, n^{o} 4, pp. 417-440

[7] J. CANOVAS, F. JOUAULT, J. CABOT, J. GARCIA MOLINA. *API2MoL: Automating the building of bridges between APIs and Model-Driven Engineering*, in "Information and Software Technology", October 2011 [*DOI :* 10.1016/J.INFSOF.2011.09.006], http://hal.inria.fr/hal-00642154/en/

[8] B. MARCO, J. CABOT, M. WIMMER. *Model-Driven Software Engineering in Practice*, Morgan & Claypool, September 2012, 182 p. , http://hal.inria.fr/hal-00755006

[9] A. VIGNAGA, F. JOUAULT, M. C. BASTARRICA, H. BRUNELIÈRE. *Typing artifacts in megamodeling*, in "Springer Journal on Software and Systems Modeling", February 2011, vol. 10, n° 1 [*DOI :* 10.1007/s10270-011-0191-2], http://www.springerlink.com/content/j212100qq405n5q1/

Publications of the year

Doctoral Dissertations and Habilitation Theses

[10] G. SUNYÉ. *A Model-Based Approach for Testing Large Scale Systems*, Université de Nantes, November 2015, Habilitation à diriger des recherches, https://hal.inria.fr/tel-01255818

Articles in International Peer-Reviewed Journals

[11] H. BRUNELIERE, J. CABOT, J. L. CÁNOVAS IZQUIERDO, L. ORUE-ECHEVARRIA, O. STRAUSS, M. WIMMER. *Software Modernization Revisited: Challenges and Prospects*, in "Computer", August 2015, vol. 48, n° 8, pp. 76-80 [*DOI :* 10.1109/MC.2015.234], https://hal.inria.fr/hal-01186371

[12] J. L. CÁNOVAS IZQUIERDO, V. COSENTINO, J. CABOT. *Attracting Contributions to your GitHub Project*, in "The Journal of Object Technology", March 2015, https://hal.inria.fr/hal-01257474

[13] S. HIDAKA, M. TISI, J. CABOT, Z. HU. *Feature-Based Classification of Bidirectional Transformation Approaches*, in "Software and Systems Modeling", 2015, https://hal.inria.fr/hal-01257169

[14] R. TAIRAS, J. CABOT. *Corpus-based analysis of domain-specific languages*, in "Software and Systems Modeling", 2015, vol. 14, n° 2, pp. 889-904 [*DOI :* 10.1007/s10270-013-0352-6], http://hal-bioemco.ccsd.cnrs.fr/LINA/hal-01201686

Articles in National Peer-Reviewed Journals

[15] S. HIDAKA, J. BÉZIVIN, Z. HU, F. JOUAULT. *Principles and Applications of Model Driven Engineering (2) Metamodeling and Model Transformation*, in "Computer Software", 2015, vol. 32, n° 2, pp. 63-78 [*DOI :* 10.11309/jssst.32.2_63], https://hal.archives-ouvertes.fr/hal-01202832

International Conferences with Proceedings

[16] D. AMELLER, X. FRANCH, C. GÓMEZ, J. ARAUJO, R. BERNTSSON SVENSSON, S. BIFFL, J. CABOT, V. CORTELLESSA, M. DANEVA, D. MÉNDEZ FERNÁNDEZ, A. MOREIRA, H. MUCCINI, A. VALLECILLO, M. WIMMER, V. AMARAL, H. BRUNELIERE, L. BURGUEÑO, M. GOULÃO, B. SCHÄTZ, S. TEUFL. *Handling Non-functional Requirements in Model-Driven Development: An Ongoing Industrial Survey*, in "23rd IEEE International Requirements Engineering Conference (RE 2015)", Ottawa, Canada, August 2015, https://hal.inria.fr/hal-01186624

[17] H. BRUNELIERE, J. GARCIA, P. DESFRAY, D. E. KHELLADI, R. HEBIG, R. BENDRAOU, J. CABOT. *On Lightweight Metamodel Extension to Support Modeling Tools Agility*, in "11th European Conference on Modelling Foundations and Applications (ECMFA 2015) (a STAF 2015 conference)", L'Aquila, Italy, July 2015, https://hal.inria.fr/hal-01146802

[18] H. BRUNELIERE, J. GARCIA PEREZ, M. WIMMER, J. CABOT. *EMF Views: A View Mechanism for Integrating Heterogeneous Models*, in "34th International Conference on Conceptual Modeling (ER 2015)", Stockholm, Sweden, October 2015, https://hal.inria.fr/hal-01159205

[19] V. COSENTINO, M. TISI, J. L. CÁNOVAS IZQUIERDO. *A Model-Driven Approach to Generate External DSLs from Object-Oriented APIs*, in "41st International Conference on Current Trends in Theory and Practice of Computer Science", Pec pod Sněžkou, Czech Republic, Book Series: Lecture Notes in Computer Science, January 2015, vol. 8939, pp. 423-435, https://hal.inria.fr/hal-01094214

[20] A. E. LABIB, M. C. PENADÉS, J. H. CANÓS, A. GÓMEZ. *Enforcing reuse and customization in the development of learning objects: a product line approach*, in "30th ACM/SIGAPP Symposium On Applied Computing (SAC 2015)", Salamanca, Spain, April 2015 [*DOI :* 10.1145/2695664.2695991], https://hal.inria.fr/hal-01095594

[21] S. MARTÍNEZ, J. GARCIA-ALFARO, F. CUPPENS, N. CUPPENS-BOULAHIA, J. CABOT. *Model-Driven Integration and Analysis of Access-control Policies in Multi-layer Information Systems*, in "IFIP SEC - 30th IFIP TC 11 International Conference", Hamburg, Germany, H. FEDERRATH, D. GOLLMANN (editors), ICT Systems Security and Privacy Protection - IFIP Advances in Information and Communication Technology, Springer, May 2015, vol. 455, pp. 218-233 [*DOI :* 10.1007/978-3-319-18467-8_15], https://hal.inria.fr/hal-01152528

[22] J. TROYA, H. BRUNELIERE, M. FLECK, M. WIMMER, L. ORUE-ECHEVARRIA, J. GORROÑOGOITIA. *ARTIST: Model-Based Stairway to the Cloud*, in "Projects Showcase at STAF 2015 conference", L'Aquila, Italy, July 2015, https://hal.inria.fr/hal-01166258

Conferences without Proceedings

[23] A. BENELALLAM, A. GÓMEZ, M. TISI. *ATL-MR: Model Transformation on MapReduce*, in "Proceedings of the Second Workshop on Software Engineering for Parallel Systems (SEPS) co-located with SPLASH 2015", Pittsburgh, United States, October 2015, https://hal.archives-ouvertes.fr/hal-01215268

[24] A. BENELALLAM, A. GÓMEZ, M. TISI, J. CABOT. *Distributed Model-to-Model Transformation with ATL on MapReduce*, in "Proceedings of 2015 ACM SIGPLAN International Conference on Software Language Engineering (SLE 2015)", Pittsburgh, United States, October 2015, https://hal.archives-ouvertes.fr/hal-01215228

[25] M. BOUSSAA, O. BARAIS, G. SUNYÉ, B. BAUDRY. *A Novelty Search Approach for Automatic Test Data Generation*, in "8th International Workshop on Search-Based Software Testing SBST@ICSE 2015", Firenze, Italy, May 2015, 4 p. , https://hal.archives-ouvertes.fr/hal-01121228

[26] M. BOUSSAA, O. BARAIS, G. SUNYÉ, B. BAUDRY. *A Novelty Search-based Test Data Generator for Object-oriented Programs*, in "GECCO 2015", Madrid, Spain, Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, July 2015, pp. 1359–1360 [*DOI :* 10.1145/2739482.2764716], https://hal.archives-ouvertes.fr/hal-01248177

[27] V. COSENTINO, J. L. CÁNOVAS IZQUIERDO, J. CABOT. *Assessing the bus factor of Git repositories*, in "22nd International Conference on Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE", Montreal, Canada, March 2015 [*DOI :* 10.1109/SANER.2015.7081864], https://hal.inria.fr/hal-01257471

[28] V. COSENTINO, J. L. CÁNOVAS IZQUIERDO, J. CABOT. *Gitana: a SQL-based Git Repository Inspector*, in "ER 2015 - 34th International Conference on Conceptual Modeling", Stockholm, Sweden, October 2015, https://hal.inria.fr/hal-01187769

[29] J. CRIADO, S. MARTÍNEZ, L. IRIBARNE, J. CABOT. *Enabling the Reuse of Stored Model Transformations Through Annotations*, in "Theory and Practice of Model Transformations", L'aquila, Italy, Theory and Practice of Model Transformations, Springer, July 2015, vol. 9152, 15 p. [*DOI :* 10.1007/978-3-319-21155-8_4], https://hal.inria.fr/hal-01257406

[30] J. L. CÁNOVAS IZQUIERDO, J. CABOT. *Enabling the Definition and Enforcement of Governance Rules in Open Source Systems*, in "International Conference on Software Engineering", Florence, Italy, May 2015, https://hal.inria.fr/hal-01115920

[31] H. ED-DOUIBI, H. BRUNELIÈRE. *Modeling software application front-ends: introducing the open source IFML graphical editor...*, in "EclipseCon North America 2015 - Modeling Symposium", San Francisco, United States, March 2015, https://hal.inria.fr/hal-01146785

[32] A. GÓMEZ, A. BENELALLAM, M. TISI. *Decentralized Model Persistence for Distributed Computing*, in "3rd BigMDE Workshop", L'Aquila, Italy, July 2015, https://hal.archives-ouvertes.fr/hal-01215280

[33] A. GÓMEZ, M. TISI, G. SUNYÉ, J. CABOT. *Map-Based Transparent Persistence for Very Large Models*, in "Fundamental Approaches to Software Engineering 18th International Conference, FASE 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings", London, United Kingdom, Lecture Notes in Computer Science, April 2015, vol. 9033, pp. 19-34 [*DOI :* 10.1007/978-3-662-46675-9_2], https://hal.inria.fr/hal-01140776

[34] M. TISI, R. DOUENCE, D. WAGELAAR. *Lazy Evaluation for OCL*, in "Models 2015: 15th International Workshop on OCL and Textual Modeling", Ottawa, Canada, September 2015, https://hal.inria.fr/hal-01221072

Scientific Popularization

[35] J. CABOT, J. L. CÁNOVAS IZQUIERDO, V. COSENTINO, B. ROLANDI. *Exploring the Use of Labels to Categorize Issues in Open-Source Software Projects*, in "International Conference on Software Analysis, Evolution and Reenginering (SANER)", Montreal, Canada, March 2015, https://hal.inria.fr/hal-01119395

[36] J. L. CÁNOVAS IZQUIERDO, V. COSENTINO, B. ROLANDI, A. BERGEL, J. CABOT. *GiLA: GitHub Label Analyzer*, in "International Conference on Software Analysis, Evolution and Reenginering (SANER)", Montreal, Canada, March 2015, https://hal.inria.fr/hal-01119396

References in notes

[37] 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, pp. 277–286

[38] 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, pp. 350–354

[39] 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", Genova, Italy, LNCS, Springer Berlin / Heidelberg, October 1-6 2006, n^{o} 4199, pp. 440–453

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

[41] 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, pp. 273-280

[42] 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, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005, n^{o} 04101

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

[44] G. CANFORA, M. DI PENTA, L. CERULO. *Achievements and challenges in software reverse engineering*, in "Commun. ACM", April 2011, vol. 54, pp. 142–151, http://doi.acm.org/10.1145/1924421.1924451

[45] 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

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

[47] 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, pp. 171–185

[48] 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, pp. 602–616