MoDisco: A Generic And Extensible Framework For Model Driven Reverse Engineering

Hugo Bruneliere, Jordi Cabot, Frédéric Jouault
AtlanMod, INRIA RBA Center & EMN
4, rue Alfred Kastler
44307 Nantes, France
+33 (0)2 51 85 82 21
Firstname.Lastname@inria.fr

Frédéric Madiot
Mia-Software
4, rue du Chateau de l'Eraudière
44324 Nantes, France
+33 (0)2 40 18 52 96
fmadiot@mia-software.com

ABSTRACT
Nowadays, almost all companies, independently of their size and type of activity, are facing the problematic of having to manage, maintain or even replace their legacy systems. Many times, the first problem they need to solve is to really understand what are the functionalities, architecture, data, etc of all these often huge legacy applications. As a consequence, reverse engineering still remains a major challenge for software engineering today. This paper introduces MoDisco, a generic and extensible open source reverse engineering solution. MoDisco intensively uses MDE principles and techniques to improve existing approaches for reverse engineering.

Categories and Subject Descriptors
D.2.7 [**Distribution, Maintenance, and Enhancement**]: Restructuring, reverse engineering, and reengineering, Documentation, Portability. D.2.13 [**Reusable Software**]: Domain Engineering, Reuse models.

General Terms
Management, Measurement, Documentation, Design, Experimentation, Standardization, Languages.

1. INTRODUCTION
Nowadays, many companies depend on a huge amount of heterogeneous legacy systems. Due to the critical importance of these legacy systems and the high cost of replacing them through modernization/migration processes, most of them are still heavily used today in production, limiting the benefits these companies could obtain from applying newer technologies and frameworks in their information systems. Situation gets worse when considering the current fast pace of evolution of technologies which implies that developed applications become rapidly obsolete.

As a consequence, the ability of quickly reengineering existing systems is still a major problem in software engineering, More specifically, the major challenge is to be able to discover and understand the functionalities, architecture, data, etc of the legacy systems and reverse engineer them into a meaningful representation that can be later on manipulated and re-implemented. The current growing adoption of Model Driven Engineering (MDE) principles and techniques, which generally consider models as first class entities in all processes, offers concrete opportunities for getting the general benefits of MDE when designing new reverse engineering solutions. Contrary to many development tools which are often specialized on generating UML models from a specific technology and vice-versa (e.g. Eclipse JDT for Java [1], Visual Studio for .NET [2]), the idea is to provide generic support for different target metamodels/formats (i.e. not only UML) and extensibility capabilities to other technologies (XML, SQL, COBOL, etc).

The MoDisco open source project [3] fills this gap by providing a generic and extensible MDE framework dedicated to reverse engineering. First, Section 2 generally describes what is Model Driven Reverse Engineering (MDRE) in MoDisco: the application of MDE in the context of reverse engineering. Then, Section 3 presents the MoDisco project as a concrete implementation. Finally, Section 4 concludes the paper.

ASE'10, September 20–24, 2010, Antwerp, Belgium.
ACM 978-1-4503-0116-9/10/09.

2. MODISCO MODEL DRIVEN REVERSE ENGINEERING APPROACH
The main issue in reverse engineering comes from the heterogeneity of the legacy applications to be considered, contrary to MDE where data is uniformly represented as models. Thus, Model Driven Reverse Engineering (MDRE) is about switching from the heterogeneous world of implementation technologies to the homogeneous world of models. The overall idea is to retrieve one or several models from a given system, depending on the needed viewpoints, and then to work directly on these models.

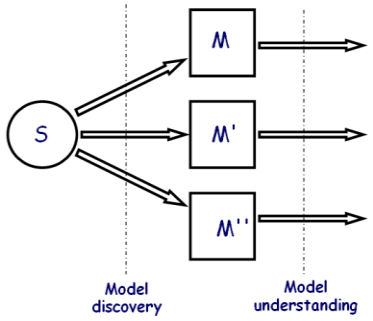

Figure 1. The two phases ("discovery" and "understanding")

When the system to be reverse engineered is represented as model(s), we are then more easily able to: 1) (re)use generic metamodels and related transformations to handle it and perform the required computations; 2) extend the support to several different kinds of technology (or tool) as both inputs and outputs of the reverse engineering process. As shown in Figure 1, we have identified two systematic consecutive phases for MDRE: "Model Discovery" and "Model Understanding".

2.1 Model Discovery

This is the initial step of every MDRE process: it consists in obtaining a model that represents a view on the legacy system (or at least parts of it) from its source code, raw data, available documentation, etc. This model provides a uniform representation of the system, which conforms to a given metamodel expressing the chosen viewpoint. This metamodel can be technology-specific or more generic, according to the expressed needs. The actual building of the model is realized thanks to a software component called "discoverer" and is always guided by the metamodel. A discoverer can have various and varied natures depending on the type of system to be reverse engineer. It can be either fully hard-coded or (preferably) partially generated (e.g. using model transformations) thanks to the metamodel.

Obviously, in real cases, several models are sometimes necessary to fully reverse engineer a given system. This often involves the use of several different discoverers, based on distinct metamodels, which can be later combined to get a unified view.

2.2 Model Understanding

This is the second phase of a MDRE process: once we have the legacy system represented as models, these models can be exploited to effectively achieve the targeted reverse engineering scenario which can be retro-documentation, metrics computation, quality analysis, refactoring, code generation, etc.

During this phase, the content of these models is analyzed and computed, in particular (re)using (generic) model transformations or chains of transformations each time possible, until we obtain the final desired representations/data (i.e. source code, documentation, structured data, etc) of/on the system. Several intermediate representations, i.e. models and corresponding metamodels, are usually necessary during the process before reaching the targeted objective. Of course this process (or at least parts of it) can often be largely automated for some common already identified scenarios, and then specialized in the context of more specific use cases.

3. THE MODISCO ECLIPSE PROJECT

MoDisco has been implemented as an Eclipse open source project which provides an extensible and customizable MDRE framework to develop model-driven tools supporting different reverse engineering scenarios such as legacy migration or modernization, quality assurance, retro-documentation, etc.

The goal of MoDisco is to offer an open source generic and extensible MDRE framework, as shown by Figure 2. Considering as inputs all kinds of possible legacy artifact (e.g. source code, databases, configuration files, documentation, etc), MoDisco aims at providing the required capabilities for creating models and allowing there handling, analysis and computation. As outputs, the framework targets the production of different types of artifact depending on the selected reverse engineering objective(s) (e.g. source code, data, metrics, visualizations, documentation, etc).

Initially created as an experimental research framework by the AtlanMod Team (EMN & INRIA), the project has now evolved to an industrialized solution thanks to our collaboration with the MIA-Software company. This active joint work has resulted in an efficient and usable set of tools for the discovery, query and manipulation of reverse engineered software models.

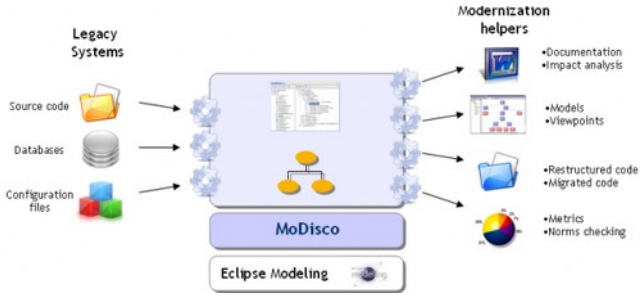

Figure 2. The MoDisco MDRE Framework

MoDisco is now fitted with generic and customizable core components such as a model browser (for large models), a model extension and customization mechanism, a model query manager, a discoverer manager and associated workflow, and some metrics visualization facilities. It also provides and uses concrete implementations of two OMG standard metamodels: Knowledge Discovery Metamodel (KDM) and Software Metrics Metamodel (SMM), which are both generic (technology-independent) metamodels. Moreover, it also offers extended technology-specific support for Java reverse engineering (including a full Java metamodel, corresponding discoverer and transformation to KDM) and XML reverse engineering (for some JEE framework configuration files such as Struts or Hibernate ones).

4. CONCLUSION

MoDisco, as a generic and extensible framework dedicated to MDRE, is a concrete proof that MDE principles and techniques can be efficiently applied in the context of important and complex domains such as reverse engineering. Moreover, as an Eclipse open source project, it shows an example of a successful joint effort between a research team and a technology provider (i.e. company) which results in the emergence of an interesting project and associated growing community.

In addition to the effective use of the MoDisco framework for new experimentations on various MDRE topics, its core components may be enhanced and the support for other implementation technologies such as C/C++, C# (.NET) or COBOL may also be developed as extensions for potential industrial applications.

5. ACKNOWLEDGMENTS

This work has been initially supported by the ModelPlex European integrated project FP6-IP 034081 [4].

6. REFERENCES

[1] Eclipse JDT: http://www.eclipse.org/jdt/

[2] Microsoft Visual Studio: http://msdn.microsoft.com/en-us/vstudio/default.aspx

[3] **MoDisco, a Model-Driven Platform to Support Real Legacy Modernization Use Cases**, By Gabriel Barbier, Hugo Bruneliere, Frédéric Jouault, Yves Lennon and Frédéric Madiot, From "Information Systems Transformation: Architecture-Driven Modernization Case Studies", The Morgan Kaufmann/OMG Press, pages 365-400, ISBN 978-0-12-374913-0, 2010

[4] ModelPlex EU Project: https://www.modelplex.org/