Second Intermediate Scientific Report
Hasler Project no. 2234
"Enabling the evolution of J2EE applications through reverse engineering and quality assurance'

September 25, 2008

a) Summary of results

Enterprise applications are typically complex and thus require elaborate techniques to understand them. This project focuses on developing dedicated reverse engineering techniques for such applications. The most significant results touch the meta-modeling of enterprise applications and the interactive presentation of the information in these applications.

Results

We present the results obtained during the period from 2008-10-01 to 2009-09-30.

Case Studies

Enterprise systems are typically proprietary. Thus, to study them we need access to case studies provided by industrial partners. During the past year we continued to work closely with one company, namely the Eidgenössischen Institut für Geistiges Eigentum (IGE). On the one hand, they described several real-life problems that they encounter, and on the other hand we presented several new analyses to solve their problems. The topics typically reside in the area of quality assurance, with problems spanning from identifying unsafe database queries to identifying inadequate layering.

Meta-modeling and Parsing

To analyze large systems, we first need to create a meta-model to represent the structure of these systems in. We are now in the process of finalizing such a meta-model that covers the Java Enterprise Beans, the structure of databases and the mapping between the code and the database. We did not build the complete meta-model from scratch, but instead we integrated it into FAMIX, a language-independent meta-model for source code analysis. Figure 1 shows the relevant excerpt from FAMIX, and in bold the J2EE-specific extensions we created.

In parallel, we built a parser for database schemas, and we started to build an interpreter that can be used for reasoning about the SQL queries that are embedded in Java code. Furthermore, we built an interpreter of the EJB (Enterprise Java Beans) XML specifications that allows us to map the beans to Java code.

Visualizations

To understand enterprise systems we need to visualize the data in ways that expose the complexity of their structure. We used the information inferred by interpreting the extra information from XML and SQL, and we developed several dedicated visualizations [Per09b].

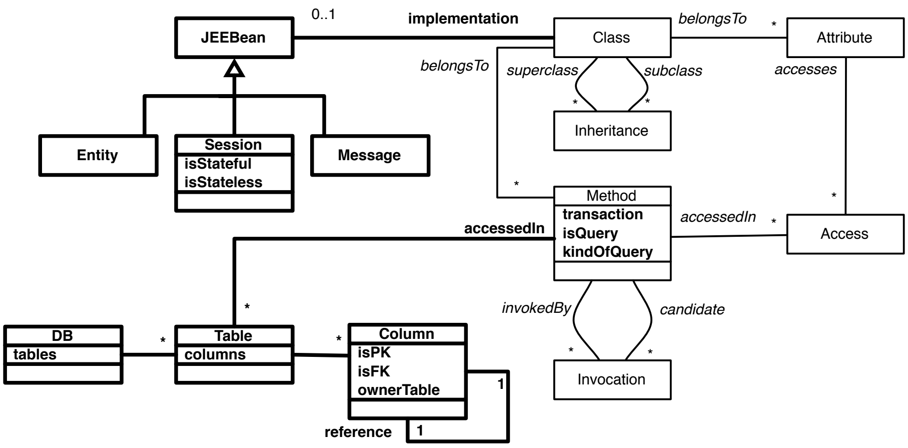

Figure 1: Enriched FAMIX Meta-Model

A tricky problem raised by the heterogeneous nature of JEAs is how to identify the scope of transactions. We developed visualizations which clearly expose which methods of a Java enterprise application are involved in a transaction, and which methods that access the database through JDBC are not involved in any transaction.

Figure 2 and Figure 3 show two visualizations that we call respectively Transaction flow and Unsafe queries. The visualizations depict facts from a large industrial case study. In both of them small squares or dots are methods and bigger squares that surround small ones are classes. Dots are methods with the particular property that they query the database using the *java.sql* package. Gray edges are invocations between classes, if they start from large rectangles to other rectangles, or between methods, if they start from small squares or dots. Classes and methods are organized into hierarchies expressing invocation order and inheritance. This means that methods of classes on top invoke methods of classes below them, or that the classes below are subclasses of the class above.

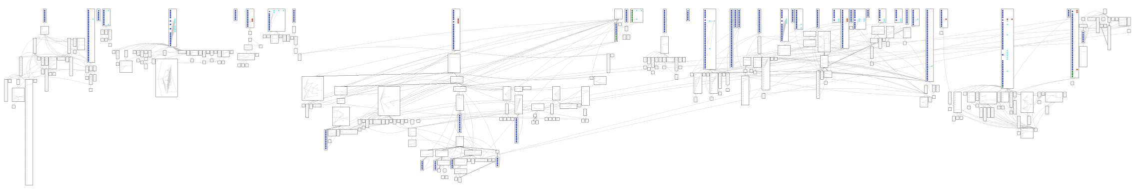

Figure 2: Transaction Flow visualization

Figure 2 shows the methods that are involved in a transaction. With the help of this visualization we can identity: (1) all methods involved in transaction, (2) methods that share a transaction scope with other methods, and (3) problems in the definition of the application's entry points. The visualization from Figure 3 is an instrument to identify which methods that access the database are not part in a transaction scope. Thus, we can use this visualization to detect the places that lead to inconsistencies in the application's data.

Conforming with the best practices for developing three tier applications, we also built a visualization that exposes these layers in a J2EE application. Figure 4 depicts the layers of an industrial application: the top layer contains all the service classes, the bottom layer contains all classes from the data access classes, and the middle layer contains the rest of the classes that are part of the control flow starting from the service layer classes. The edges denote invocations and are colored in orange if the invocation bypasses the middle layer, and gray otherwise. From the visualization we can detect behavior that breaks the architecture (like a database query coming directly from the service layer). Furthermore, this visualization offers a means to identify the domain model that should be present in the middle layer.

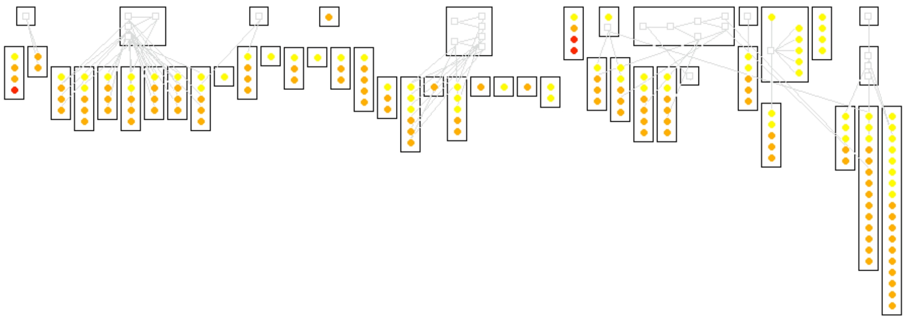

Figure 3: Unsafe queries visualization

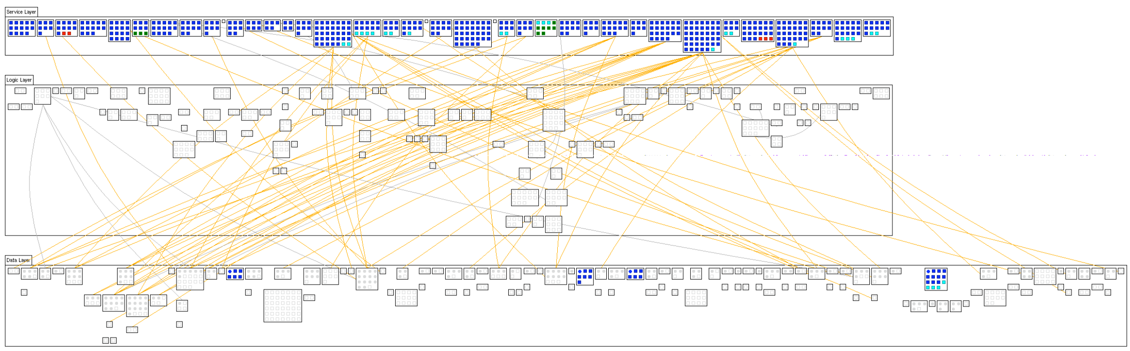

Figure 4: Exposing communication paths between layers.

Interactive Browsers

While visualizations are useful tools for revealing the structure of unknown data, the complexity of enterprise applications requires us to explore the various facets using interactive browsers. However, dedicated browsers are expensive to build and difficult to maintain. Thus, on a second track, we developed Glamour, a novel engine for specifying browsers that allows us to interactively navigate data [Bun09, BGR^{+}09]. Glamour browsers are specified using a domain specific language based on a component and connectors model. On-the-fly transformations enable Glamour to browse arbitrary domain models.

An example of a browser built with Glamour can be seen in Figure 5. This browser is displaying the actual implementation of our J2EE meta-model. The top-left pane shows the list of entities in our meta-model, with the one for the implementation of the SessionBean being selected. The other panes show details about this entity, including the attributes (middle and right pane from the top), a graphic overview of the entire hierarchy (bottom left) and the comments (bottom right).

Development process

In parallel, we started to examine further information that can shed light on the development of software systems. In particular, we wanted to understand the paths of communication in a development team. Thus, we analyzed the information present in the development mailing lists and we built several visualizations to identify expertise [SMG09].

Enterprise systems rarely exist in isolation. Instead they co-exist and interact with other project within the larger context of the organization. Typically, these projects reside in large super-repositories. We worked closely with the University of Lugano to develop an approach and a tool to understand a project within the larger context [LLGR09].

Entreprise systems are an example of *software intensive systems*. We contributed to a special LNCS

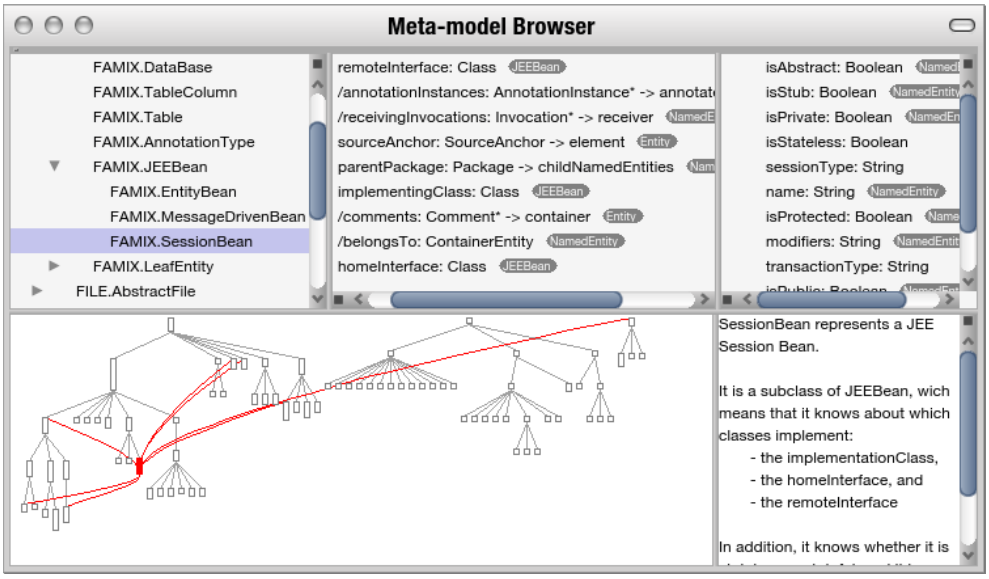

Figure 5: Example of a Glamour Browser showing the meta-model implementation.

volume dedicated to research issues related to software intensive systems by identifying several ways in which such systems can benefit by dedicated support to enable change [NDG+08].

Workshops and Tutorials

In this project, the implementation and research effort is based on the Moose platform[1]. Moose was originally built at the University of Berne and it recently started to attract interest from a number of other researchers. To strengthen the community and to exchange ideas we are organizing a third edition of the Workshop on FAMIX and Moose in Software Reengineering, which will take place October 17, 2009, co-located with WCRE 2009[2].

Staff contributions

- Fabrizio Perin is in the first year of the PhD. He built a meta-model for analyzing J2EE systems and started to investigate several dedicated visualizations and metrics. He actively worked with our industrial partner, Eidgenössischen Institut für Geistiges Eigentum (IGE), to identify design quality problems and develop solutions.
- Tudor Gîrba served as PostDoc. He lead the work on Glamour, a scripting engine for building dedicated interactive tools. Glamour was awarded the 3rd prize at the ESUG 2009 Innovation Technology Awards. He also worked on understanding the dynamics from mailing lists and on reverse engineering super-repositories.

Changes to the research plan

No major changes intervened in the research plan over the past year.

[1] `http://moose.unibe.ch`
[2] 16th Working Conference on Reverse Engineering, Lille, France

Important events

- Tudor Gîrba was an invited speaker at Club Qualimetrie (What software history can tell us, The humane software assessment — Paris, France, June 9, 2009).
- Tudor Gîrba was an invited speaker at ESUG 2009 (Scripting your browsers in 15 minutes with Glamour — Brest, France, Sept 1, 2009)
- Oscar Nierstrasz was PC Member:
  - PC Member of SUITE 2009 (First Intl. Workshop on Search-Driven Development — Users, Infrastructure, Tools and Evaluation — Colocated with ICSE 2009, Vancouver, Canada, May 16 2009).
  - PC Member of Models@run.time (Colocated with Models 2008 — Toulouse, France, Sept 28-Oct 3, 2008).
  - PC Member of FAMOOSr 2008 (2nd Workshop on FAMIX and Moose in Reengineering — Colocated with WCRE 2008 — Antwerp, Belgium, Oct. 17, 2008).
- Tudor Gîrba was PC Member:
  - PC Member of IWST09 (International Workshop on Smalltalk Technologies — Colocated with ESUG 2009, Brest, France, Aug 31, 2009)
  - PC Member of MSR 2009 (Working Conference on Mining Software Repositories).
  - PC Member of TOOLS 2009 (International Conference Objects, Models, Components, Patterns — Zurich)
  - PC member of ENASE 2009 (International Conference on Evaluation of Novel Approaches to Software Engineering —)
  - Co-organizer of FAMOOSr 2008 (2nd Workshop on FAMIX and Moose in Reengineering co-located with WCRE 2008).
  - PC Member of WCRE 2008 (Working Conference on Reverse Engineering).
  - PC Member of ICSM 2008 (International Conference on Software Maintenance — Beijing, China, Sept. 28 - Oct 4, 2008).
  - PC Member of MODELS 2008 (International Conference on Model Driven Engineering Languages and Systems).
  - PC Member of MCCM 2008 (International Workshop on Model Co-Evolution and Consistency Management - co-located with MODELS 2008).
  - Tool Demonstration PC Member for ASE 2008 (International Conference on Automated Software Engineering 2008).
- Tudor Gîrba was co-organizer of FAMOOSr 2009 (Workshop on FAMIX and Moose in Reengineering — co-located with WCRE 2009).

b) Publications

Published papers are annexed to this report. They are all available electronically as PDF files at the following url:

`http://www.iam.unibe.ch/~scg/cgi-bin/scgbib.cgi?hasler09`

Please note that theses and student projects are *not* included with this report, but are nevertheless available electronically from the above URL.

Published papers

[BGR+09] Philipp Bunge, Tudor Gîrba, Lukas Renggli, Jorge Ressia, and David Röthlisberger. Scripting browsers with Glamour. European Smalltalk User Group 2009 Technology Innovation Awards, August 2009. Glamour was awarded the 3rd prize.

[NDG+08] Oscar Nierstrasz, Marcus Denker, Tudor Gîrba, Adrian Lienhard, and David Röthlisberger. Change-enabled software systems. In Martin Wirsing, Jean-Pierre Banâtre, and Matthias Hölzl, editors, *Challenges for Software-Intensive Systems and New Computing Paradigms*, volume 5380 of *LNCS*, pages 64–79. Springer-Verlag, 2008.

Theses and Student projects

[Bun09] Philipp Bunge. Scripting browsers with Glamour. Master's thesis, University of Bern, April 2009.

c) Publications in press

Publications to appear

[LLGR09] Mircea Lungu, Michele Lanza, Tudor Girba, and Romain Robbes. The small project observatory: Visualizing software ecosystems. *Science of Computer Programming, Elsevier*, 2009. To appear.

[Per09a] Fabrizio Perin. Driving the refactoring of Java Enterprise applications by evaluating the distance between application elements. In *Proceedings of FAMOOSr at the Working Conference on Reverse Engineering (WCRE 2009))*, pages xxx–xxx, October 2009.

[Per09b] Fabrizio Perin. Enabling the evolution of J2EE applications through reverse engineering and quality assurance. In *Proceedings of the PhD Symposium at the Working Conference on Reverse Engineering (WCRE 2009))*, pages xxx–xxx. IEEE Computer Society Press, October 2009.

[SMG09] François Stephany, Tom Mens, and Tudor Gîrba. Maispion: A tool for analysing and visualizing open source software developer communities. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2009)*. ACM Digital Library, 2009. To appear.