Network Of Reengineering Expertise (NOREX)
SNF SCOPES/JRP Project IB7320-110997

Final Scientific Report

1 General Information

Title of the JRP	**NOREX — Network of Reengineering Expertise**
Number of the JRP	IB7320-110997
Starting Date	November 1 2005
Duration	24 months
Coordinator	**Michele Lanza**
	Faculty of Informatics, University of Lugano, Switzerland
Partner 1	**Radu Marinescu**
	Institute e-Austria, Timişoara, Romania
Partner 2	**Oscar Nierstrasz**
	University of Berne, Software Composition Group, Bern, Switzerland

2 Overview of Research Activities

2.1 Objectives of the JRP

Reengineering is a complex task which requires multiple techniques such as parsing, modeling, data mining, visualization, etc. The past few years have shown that single researchers and even single research groups are cannot tackle all problems involved in building working solutions for reengineering large industrial systems. Moreover, as research groups tend to specialize in one particular field, *e.g.*, data mining or parsing, they must rely on external, often commercial, tools to be able to perform a full analysis.

Our intention was to bridge this gap, as the participating research groups have all built powerful research prototypes in complementary areas: the tools should collaborate with each other at the level of infrastructure. We wanted tools to collaborate in real-time as web services by building a distributed reengineering environment which is able to make all these techniques and models complement each other. Then we wanted to use this environment to integrate different reengineering techniques to create new and composite ones.

2.2 Work carried out by the teams

Our research activities have been organized in 3 work packages in conformity with the original proposal:

- Creating a Distributed Reengineering Environment (WP1)
- Enabling Complex (Distributed) Reengineering Activities (WP2)
- Performing Large-Scale Distributed Reengineering Experiments (WP3)

2.2.1 Creating a Distributed Reengineering Environment (WP1)

The goal of this WP has been to build a prototype of the NOREX environment. The first task of the environment is to ensure that different tools can understand each other by understanding the semantics of their models. The semantics of a model are expressed in its meta-model. Therefore, we have first specified a common meta-meta-model, using the implementation of MOOSE and iPlasma as a starting point, since they implement similar meta-meta-models. Furthermore, the tools need to be implemented in different languages. Consequently, we need to ensure that the different tools can communicate with each other.

The main achievement of this WP has been the creation of a prototypical version of NOREX. Our main goal with NOREX is to allow for different groups to build tools in a loose fashion, that is without imposing hard-coded dependencies between these tools. Our solution is based on the following approach: each reengineering tool or service that is provided to the other nodes of the network decouples the actual algorithm from the data (*i.e.*, the meta-model specific part) on which it performs. Thus, each service expresses its dependency on a meta-model in terms of a set of Command objects [GHJV95], each one representing a particular information retrieval action performed on the concrete model. In other words, a service will describe what information it needs to run, but it completely decouples from how this information is provided by its clients. Part of this achievement is the fact that we have defined a common meta-meta-model that allows us to abstract from the particularities of the various meta-models used by our groups. Thus, we can describe any of these meta-models in terms of several major concepts (*e.g.*, entities having entity types, group entities as a special type of entity, etc.). While we wrote a paper dedicated to the entire construction of NOREX ([BMGM07] is a position paper on NOREX while a tool demo has been already presented at ICSM07 [BMML07]), the idea of a common meta-meta-model is partially described in [DG06]. Furthermore, in [NDG05] we emphasize the benefits of an "agile" reengineering environment. Using the common meta-model, we have defined a so-called service manifesto by which a reengineering service describes itself to its clients.

One important advantage of this approach is that it provides a "lazy mapping" between the meta-models used in the different nodes (*i.e.*, formed by the different groups) of the network. To use a NOREX service, one needs only to bind the entity type and command identifiers from the service manifesto to concrete meta-model objects. The second significant advantage of this approach is its incremental nature, *i.e.*, a binding must be done only once. Consequently, the more NOREX services are used the less extra work is expected before the actual usage of the service (as it is very probable that most, if not all the bindings are already encoded). The deliverables of this WP are:

- (D1.1) The common meta-meta-model was completely specified and implemented. The fact that both MOOSE and iPlasma use a similar meta-meta-model has simplified this task.
- (D1.2) The communication protocol is completely specified and implemented for both programming languages (*i.e.*, Java and Smalltalk).
- (D1.3) The design and implementation of the registration mechanism for the reengineering tools has been fully completed.

Work organization: Dr. Gîrba (Bern) together with Prof. Lanza (Lugano) and Prof. Marinescu (Timişoara) have designed the common meta-meta-model, the registration mechanism and the communication protocol. Researcher Mihai Balint and Ph.D. students Petru Mihancea and Cristina Marinescu (all from Timişoara) have done most of the implementation work on NOREX. Ph.D. student Dan Cosma (Timişoara) has supervised the WebService specific design and implementation decisions. Richard Wettel (Timişoara) and Mircea Lungu (Lugano) have implemented the NOREX mechanisms in Smalltalk.

2.2.2 Enabling Complex (Distributed) Reengineering Activities (WP2)

Using the NOREX environment we have explored the possibilities to build complex analyses by combining tools built by the participating groups. In particular, we have focused our attention on combining problem detection with visualization to get an overview of how design problems are distributed over a system.

The main achievements of this work package are related to a significant series of publications, including joint publications between the partners involved in the current project. The most significant one is the

book "Object-Oriented Metrics in Practice" [LM06]. It is co-authored by Prof. Lanza (Lugano) and Prof. Marinescu (Timişoara) and it shows how metrics-based detection strategies and visualizations are used together in form of a catalogue of patterns for understanding and assessing the design quality of object-oriented systems. Moreover, the book includes, in the form of a set of activity diagrams, restructuring alternatives for correcting typical object-oriented design flaws.

Apart from this book, the members of the three teams have published several papers on various types of reengineering activities. We developed a novel technique for analyzing the quality of class hierarchies [Mih06]. We extended our initial problem detection approach by taking into consideration contextual hints that would lead to an adequate restructuring [TM05]. We showed how both code understanding and problem detection approaches need to be adapted for enterprise software systems in [Mar06]. We also showed how we can use historical information to detect co-change patterns and design flaws [GDK+07].

As one of our goals has been to integrate in the NOREX environment efficient visualization techniques we developed advanced tools for visualization like Mondrian [MGL06] (and its Java version JMondrian) and published several papers on novel visualization techniques. We have also developed a series of visualizations to help grasp the complexity of software systems. In [BGM06] we investigate the correlation between code duplication and the authors of those duplications, and propose a visualization technique that reveals several such correlation patterns. In [LLG06] we show how the architecture of a system can be recovered by means of visual patterns related to the structure of packages using a prototype [LL06]. We have also defined dedicated views to identify structural characteristics of distributed software systems [CM07].

Systems seldom live alone, but rather they exist in a larger context that we call a *super-repository*. Thus, we have built several visualizations and we implemented them in the Small Project Observatory (SPO), a web-based exploratory tool [LG07]. We deployed our tool in a real-life setting and reported on our experience [LLGH07]. Furthermore, SPO obtained the 1st Prize at the Technology Innovation Awards during the 15th International Smalltalk Conference [LL07].

Additionally, to analyze enterprise applications, we have developed dedicated techniques to detect specific design problems in this type of systems [Mar07a, Mar07b]. These dedicated analysis means not only that extend the set of design flaws one can detect but also helps to eliminate many false positives (*i.e.*, design entities that apparently are affected by a design flaw from the object-oriented theory point of view but which are not design flaws in this kind of object-oriented systems) [Mar06]. These achievements are relevant in the context of our reengineering environment. Since these analyses (together with their implementations) are as far as we know the single ones which address such issues, many reengineering groups could make use of them via NOREX to refine their findings during the analyses of an enterprise application.

Analyzing systems highly depends on parsing them. To analyze C++ and C# systems, two model capturing tools have been created [MGV+07]. These tools are built around a common meta-model and thus can easily be integrated in NOREX. Implementing parsers for new languages is far from trivial, and we have also explored a novel approach to generate models from given mapping examples [NKG+07].

All the software modules that have been integrated into the NOREX environment have been put together into a central repository of reengineering services. Although initially we planned to create a website to describe the developed and implemented techniques we have realized that a repository of reengineering services would be more efficient for a proper dissemination of our achievements and for a faster access to their implementation by other interested research groups. This repository can be inspected from both iPlasma and MOOSE via the "NOREX Services Browser". This browser can be used to inspect the manifest of each reengineering service from the repository and can also be used to configure a particular service. The deliverables of this WP are:

- (D2.1) The catalogue of reengineering patterns defined in [LM06]
- (D 2.2) As we have already mentioned, in [LM06] we also describe in a step by step manner the corrective actions that could be performed in order to eliminate some typical design problems
- (D2.3) The prototype tool for simulating refactorings on the model level has been finished
- (D 2.4) A central reengineering repository which links to all the available reengineering infrastructures has been created

Work organization: Prof. Lanza (Lugano) and Prof. Marinescu (Timişoara) have devised the structure of the catalogue of complex detection techniques for design flaws and helped by Dr. Gîrba (Bern). Ph.D. students Petru Mihancea and Cristina Marinescu (both from Timişoara) have defined a set of such techniques together with refactoring plans for the most common design problems. Dr. Gîrba and Ph.D. student Mircea Lungu (Lugano), and Prof. Lanza have been involved in the development of Mondrian. Ph.D. student Mircea Lungu (Lugano), together with Prof. Lanza and Dr. Gîrba constructed the Small Project Observatory. In Timişoara, Mihai Balint and Petru Mihancea worked on creating JMondrian. The refactoring simulator has been implemented by Mihai Balint and Cristina Marinescu. Dan Cosma has created, maintained and organized the repository of reengineering services.

2.2.3 Performing Large-Scale Distributed Reengineering Experiments (WP3)

The first achievement of this workpackage is that we have been able to identify several case-studies that are large enough to prove the feasibility of the NOREX platform. One of these is the ArgoUML modelling tool, written in Java. Another one is a very-large C++ proprietary system from one of our industrial partners that we also intend us in our experiments. In addition to these two systems, we identified a third medium-size case-study written in Java that we intend to use especially for the experiments with analyses dedicated for enterprise applications. Using these selected software systems, we have conducted several case studies to prove the validity of our developed techniques and the feasibility and scalability of our environment. The main idea was to exercise some of the complex reengineering activities defined as workflows in WP2 to provide insights into the design problems of the case studies and on their potential restructuring. All the implied activities have been performed using the tools integrated in the NOREX environment.

As a result of these case studies, we have been able to finalize the catalogue of frequent design problems and of reengineering activities that can help to eliminate them. Moreover, we have been able to synthesize a methodology of using various analyses implemented in our distributed environment based on some characteristics of the analyzed systems. The deliverables of this WP are:

- (D 3.1) As mentioned, we have successfully found 3 case-studies
- (D 3.2) A catalogue of frequent design problems and how to address them
- (D 3.3) A methodology for the effective usage of the various types of analyses defined and implemented within the distributed reengineering environment

Work organization: Prof. Lanza and Prof. Marinescu, helped by the other members of their teams have chosen the first two case-studies. The third case-study, *i.e.*, the enterprise system, was chosen by Cristina Marinescu (Timişoara). The catalogue of design problems, and the subsequent methodology, was mostly put together by Prof. Lanza and Prof. Marinescu with the help of Dr. Gîrba.

2.3 Scientists involved

Mihai Balint	M	28/09/82	PhD Student, Timisoara
Dan Cosma	M	16/09/73	PhD Student, Timisoara
Marco D'Ambros	M	12/10/79	PhD Student, Lugano
Dr. Tudor Gîrba	M	08/08/77	Post-doc, Bern
Dr. Orla Greevy	F	16/03/63	Post-doc, Bern
Prof. Michele Lanza	M	03/04/73	Project Co-ordinator, Lugano
Mircea Lungu	M	24/10/80	PhD Student, Lugano
Cristina Marinescu	F	06/09/79	PhD Student, Timisoara
Prof. Radu Marinescu	M	10/06/74	Project Partner, Timisoara
Petru Florin Mihancea	M	20/12/79	PhD Student, Timisoara
Prof. Oscar Nierstrasz	M	15/10/57	Project Partner, Bern
Richard Wettel	M	12/11/74	PhD Student, Lugano

3 Scientific Results

3.1 Main scientific results

Enabling immediate communication between reengineering tools has been one of the major achievements of the NOREX project. Although of a rather technical nature this achievement is not without scientific significance, as reengineering is a complex task that requires several techniques to be employed together [DDN02] and communication between implementations is essential. Before the NOREX framework was developed, communication was possible only through data exchange using specific document formats (*e.g.*, XMI, CDIF) [HWS00, XMI05]. Several resolutions to the problem of tool interconnection have been proposed, but none has succeeded in establishing a significant foothold within the community because they fail to acknowledge the distributed nature of the development of reengineering techniques.

The most successful approaches so far have been the reengineering environments developed within some research groups. Infrastructures like MOOSE [DGLD05] and IPLASMA [MMM+05] have been successful because they provide a rich set of techniques as well as easy development of new techniques. However the features present in each one are constrained to those developed by their respective communities. The NOREX reengineering environment acknowledges the distributed nature of the research community and provides a means for developing and combining reengineering techniques, allowing researchers to focus on creative work rather than "gluing together" tools.

The other critical component of NOREX consists in the development of novel reengineering techniques. The authors have published several papers in high visibility conferences tackling different reengineering issues. Some of these techniques have been implemented as NOREX services and have been added to a central repository from which they can be easily accessed. Until now NOREX contains services for software metrics, detection strategies and software visualizations. Integrating other services (*i.e.*, source-code duplication analysis, refactoring simulation and model extraction from source files) is a straightforward engineering activity.

Most of the individual techniques have been collected and catalogued in a book by Prof. Lanza and Prof. Marinescu [LM06]. More than simply describing individual software design flaws, the book also provides possible courses of action for the removal of each flaw.

3.2 Publications

1. "Norex: A Distributed Reengineering Environment". Mihai Balint, Petru F. Mihancea, Tudor Gîrba, Radu Marinescu. In Proceedings of ICSM 2007 (23rd IEEE International Conference on Software Maintenance), Paris, France, Tool Demos Section, IEEE Computer Society Press, 2007.

2. "Using Concept Analysis to Detect Co-Change Patterns". Tudor Gîrba, Stephane Ducasse, Adrian Kuhn, Radu Marinescu and Daniel Ratiu. In Proceedings of IWPSE 2007 (9th International Workshop on Principles of Software Evolution), pp. 83-89, IEEE Computer Society Press, 2007.

3. "A small observatory for super-repositories". Mircea Lungu and Tudor Gîrba. In Proceedings of IWPSE 2007 (9th International Workshop on Principles of Software Evolution), pp. 106-109, IEEE Computer Society Press, 2007.

4. "Reverse engineering super-repositories". Mircea Lungu, Michele Lanza, Tudor Gîrba, and Reinout Heeck. In Proceedings of WCRE 2007 (14th Working Conference on Reverse Engineering), pp. 120-129, IEEE Computer Society Press, 2007.

5. "Discovering the Objectual Meaning of Foreign Key Constraints in Enterprise Applications". Cristina Marinescu, In Proceedings of WCRE 2007 (14th Working Conference on Reverse Engineering), pp. 100-109, IEEE Computer Society Press, 2007.

6. "Distributable Features View: Visualizing the Structural Characteristics of Distributed Software Systems". Dan Cosma, Radu Marinescu. In Proceedings of VISSOFT 2007 (4th IEEE International Workshop on Visualizing Software for Understanding and Analysis), pp. 55-62, IEEE Computer Society Press, 2007.

7. “Towards Reengineering 2.0”. Mihai Balint, Petru F. Mihancea, Radu Marinescu, Tudor Gîrba, Michele Lanza. Submitted to 12th European Conference on Software Maintenance and Reengineering (CSMR 2008), Athens 2008.

8. “NOREX: Distributed collaborative reengineering”. Mihai Balint, Petru F. Mihancea, Tudor Gîrba, Radu Marinescu, Michele Lanza. In Proceedings of FAMOOSr (1st Workshop on FAMIX and MOOSE in Reengineering), Zurich, Switzerland, 2007

9. “McC and Mc#: Unified C++ and C# Design Facts Extractors Tools”. Petru F. Mihancea, George Ganea, Ioana Verebi, Cristina Marinescu, Radu Marinescu. In Proceedings of SYNASC 2007 (9th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing), IEEE Computer Society Press, 2007.

10. “A visual approach towards the detection of developer duplication patterns in software sourcecode”. Mihai Balint, Diploma thesis, from the “Politehnica” University of Timisoara, June 2006.

11. “How Developers Copy”. Mihai Balint, Tudor Gîrba, Radu Marinescu. In Proceedings of ICPC 2006 (14th IEEE International Conference on Program Comprehension), pp. 56-65, IEEE Computer Society Press, 2006.

12. “Towards a Client Driven Characterization of Class Hierarchies”. Petru F. Mihancea. In Proceedings of ICPC 2006 (14th IEEE International Conference on Program Comprehension), pp. 285-294, IEEE Computer Society Press, 2006.

13. “Example-driven reconstruction of software models”. Oscar Nierstrasz, Markus Kobel, Tudor Gîrba, Michele Lanza, and Horst Bunke. In Proceedings of CSMR 2007 (11th European Conference on Software Maintenance and Reengineering), pp. 275-284, IEEE Computer Society Press, 2007.

14. “Object-Oriented Metrics in Practice”. Michele Lanza and Radu Marinescu. Springer-Verlag, 2006.

4 Impact and Benefits

4.1 Impacts

NOREX was recently demoed at a prominent software maintenannce conference (ICSM 2007) [BMML07] and was received very well. We were pleasantly surprised that after the presentation we were contacted by other researchers interested both in using the techniques implemented in NOREX and in implementing their own techniques for others to use. We were expecting that eventually other researchers would be interested in NOREX however we we did not expect it to happen so soon. As a result, we have established relations with the group developing the Ptidej [DiJ07] reengineering tool suite at the software engineering laboratory of the university of Montral, Canada. We are currently working with the Ptidej team, guiding them to integrate some of their pattern identification techniques in NOREX whilst providing them with the already available visualizations and detection strategies.

We have also established contact with the LORE lab from University of Antwerp, Belgium. They are developing tool chains [oR07] to investigate techniques to reengineer software and are interested in source-code fact extractors for object-oriented languages. They are already using Moose [DGLD05] to analyze their models and are willing to investigate the possibility of integrating a NOREX service for model building (parser) that can be easily reused.

The idea of services for software maintenance and reengineering has been discussed with our contacts from Timişoara’s local software industry. This idea appealed to them especially since Timişoara has a vibrant software industry with companies like Alcatel-Lucent, Siemens VDO (recently acquired by Continental AG) OCE and several SMEs specializing in software testing and maintenance.

One of the major failures in the adoption by software companies of proprietary reengineering and maintenance products has been the lack of flexibility of such products. NOREX’s approach of maintenance-as-service is a lot more appealing simply because services can be easily integrated within the company’s business process and development tools. As such, we are currently planning to integrate NOREX with Eclipse — a major open source IDE that is widely used in the industry.

4.2 Benefits

Consolidated collaboration. As we expected at the beginning, this project has had a significant benefit for all the partners, and especially on maintaining and consolidating the already long-lasting collaboration links between the two research groups in Switzerland and the LOOSE Research Group in Timişoara (Romania). The significant number of common publications provides an objective measure of how beneficial this collaboration has been from a scientific point of view.

Better visibility for Romanian scientists. One of the major benefits for the Romanian side is that the project significantly contributed to improving the international visibility of young Romanian scientists (*i.e.*, Ph.D. students). This project has stimulated Ph.D. students from the research group in Timişoara to keep working on their ideas, as these ideas eventually became accessible through the proposed network, and have been disseminated at an international level (*i.e.*, renowned conferences) increasing thus their visibility. Furthermore, the financial support for the master and Ph.D. students from Timişoara who were involved in this project helped them to stay on a research track instead of switching to a (much better paid) job in the local software industry. Additionally, the various project meetings, doubled by the participation in main scientific events has added a significant amount of experience to the Romanian members of the projects.

Stimulative interaction of young scientists. Almost all members of the three research teams involved in this project are young scientists (under 35 years), many of them at the beginning of their Ph.D. studies. The NOREX project has stimulated an enlargement of horizon for all the participants, by allowing them to interact (via the reengineering network) with related achievements of their colleagues from the other groups.

5 Meetings and visits, co-operations, practical issues

5.1 Meetings and visits

- – **Person**: Prof. Radu Marinescu (Timişoara, Romania)
  – **Destination & Date**: Bern and Lugano, 14 - 22 November 2005
  – **Purpose**: First project meeting and visit for initial discussions on designing and planning the first tasks of the project.
- – **Person**: Dr. Tudor Gîrba (Bern, Switzerland)
  – **Destination & Date**: Lugano, 18 - 20 November 2005
  – **Purpose**: First project meeting and visit for initial discussions on designing and planning the first tasks of the project.
- – **Person**: Petru F. Mihancea (Timişoara, Romania)
  – **Destination & Date**: Bern, 16 - 19 February 2006
  – **Purpose**: Second project meeting, discussing the current status of the project and planning the next activities, with a special emphasis on designing and implementing the NOREX infrastructure.
- – **Person**: Mihai Balint (Timişoara, Romania)
  – **Destination & Date**: Athens, 13 - 17 June 2006
  – **Purpose**: Attending the 14th IEEE International Conference on Program Comprehension (ICPC) where he presented a full-paper related to the topic of the project.
- – **Person**: Prof. Radu Marinescu, Cristina Marinescu (Timişoara, Romania)
  – **Destination & Date**: Lugano and Bern, 13 - 17 June 2006

  - **Purpose**: Project meetings with Dr. Tudor Gîrba and Prof. Michele Lanza on the project status with a special emphasis on the performance and extensibility issues of the NOREX platform.

- – **Person**: Mircea Lungu (Lugano, Switzerland)
  - **Destination & Date**: Timisoara, 5 - 17 February 2007
  - **Purpose**: Face to face synchronization meeting.

- – **Person**: Mihai Balint (Timişoara, Romania)
  - **Destination & Date**: Paris, 30 September - 6 October 2007
  - **Purpose**: Participation at the 23rd IEEE International Conference on Software Maintenance (ICSM 2007), demoing the NOREX platform during the ICSM Tool Demo session and disseminating the collaboration possibilities created by the NOREX platform.

- – **Person**: Dr. Tudor Gîrba (Bern, Switzerland)
  - **Destination & Date**: Lugano, 24 - 31 August 2007
  - **Purpose**: Attending International Joint Smalltalk Conference (ISC 2007), presenting Moose and Small Project Observatory.

- – **Person**: Prof. Michele Lanza (Lugano, Switzerland), Prof. Radu Marinescu (Timişoara, Romania), Dr. Tudor Gîrba (Bern, Switzerland)
  - **Destination & Date**: Dubrovnik, 2 - 7 September 2007
  - **Purpose**: Final project meeting together with Prof. Michele Lanza and Dr. Tudor Gîrba summarizing the achievements of the project and exploring possibilities for a continuation of the project. Location was chosen due to the IWPSE event where Dr. Tudor Gîrba and Prof. Radu Marinescu presented a paper, partially related to the topic of this project.

5.2 Co-operations

As NOREX can be seen as a new player in the reengineering community, it has sparked great interest in other research groups that want to interface with it, but as these are ongoing or planned activities, they have not resulted in concrete co-operations yet.

5.3 Practical issues

No major problems were encountered during the entire lifetime of the project. Concerning the transfer of funds we proceeded as follows:

- In the first year, the entire scientific equipment was bought in Switzerland and also the entire budget for "Accommodation and Meals" and "Transport Costs" was managed from the University of Lugano, instead of transferring it to Romania. The rest of the money for the Romanian partner, *i.e.*, the money for "Individual Grants", "Consumables" and "Overheads" were transferred to the e-Austria Institute in Timişoara and was managed from there. The Romanian partner has provided us with all the receipts both for the "Individual Grants" and for their expenses on consumables.

- During the second year the money from all budget categories was transferred to the e-Austria Institute in Timişoara (Romania) and was managed directly by our Romanian partners. The only money that was spent directly from Switzerland involved the buying of several books for our Romanian partner. As in the first year, the Romanian partner has provided us with all the receipts both for the "Individual Grants" and for their expenses on consumables.

6 Role of SCOPES

The SCOPES programme was crucial to our project, as otherwise the other only funding option for a transnational project would have been the European Community with its bureaucracy-heavy projects. For us it was ideal to have a low-overhead project which let us concentrate on the actual research.

The weak point of the current setup of SCOPES is that Swiss partners do not get funded apart from some travelling money. While we are aware that the focus of SCOPES is on funding low-budget countries, we think it would have been enough to get one Swiss PhD position funded to obtain even more research results. As such, the large part of the project was off-loaded to Eastern Europe with only little interaction at the implementation level, since only the foreign partner received enough funding to afford the necessary manpower.

References

[BGM06] Mihai Balint, Tudor Gîrba, and Radu Marinescu. How developers copy. In *Proceedings of International Conference on Program Comprehension (ICPC 2006)*, pages 56–65, 2006.

[BMGM07] Mihai Balint, Petru Florin Mihancea, Tudor Gîrba, and Radu Marinescu. Norex: A distributed reengineering environment. In *Proceedings of International Conference on Software Maintenance (ICSM 2007)*, pages 523–524, September 2007. Tool demo.

[BMML07] Mihai Balint, Petru Florin Mihancea, Radu Marinescu, and Michele Lanza. Norex: Distributed collaborative reengineering. In *Proceedings of FAMOOSR 2007 (1st Workshop on FAMIX and Moose in Reengineering)*, 2007.

[CM07] Dan Cosma and Radu Marinescu. Distributable features view: Visualizing the structural characteristics of distributed software systems. In *Proceedings of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2007), Alberta, Canada*. IEEE Computer Society Press, 2007.

[DDN02] Serge Demeyer, Stéphane Ducasse, and Oscar Nierstrasz. *Object-Oriented Reengineering Patterns*. Morgan Kaufmann, 2002.

[DG06] Stéphane Ducasse and Tudor Gîrba. Using Smalltalk as a reflective executable metalanguage. In *International Conference on Model Driven Engineering Languages and Systems (Models/UML 2006)*, volume 4199 of *LNCS*, pages 604–618, Berlin, Germany, 2006. Springer-Verlag.

[DGLD05] Stéphane Ducasse, Tudor Gîrba, Michele Lanza, and Serge Demeyer. Moose: a collaborative and extensible reengineering environment. In *Tools for Software Maintenance and Reengineering*, RCOST / Software Technology Series, pages 55–71. Franco Angeli, 2005.

[DiJ07] Pattern Trace Identification Detection and Enhancement in Java. Ptidej team, Software Engineering Laboratory, University of Montreal, Canada, 2007. http://ptidej.dyndns.org, http://www.iro.umontreal.ca/labs/gelo.

[GDK+07] Tudor Gîrba, Stéphane Ducasse, Adrian Kuhn, Radu Marinescu, and Daniel Raţiu. Using concept analysis to detect co-change patterns. In *Proceedings of International Workshop on Principles of Software Evolution (IWPSE 2007)*, pages 83–89, 2007.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, Reading, Mass., 1995.

[HWS00] Richard C. Holt, Andreas Winter, and Andy Schürr. GXL: Towards a standard exchange format. In *Proceedings WCRE '00*, November 2000.

[LG07] Mircea Lungu and Tudor Gîrba. A small observatory for super-repositories. In *Proceedings of IWPSE 2007 (9th International Workshop on Principles of Software Evolution*, pages 106–109. ACM Press, 2007.

[LL06] Mircea Lungu and Michele Lanza. Softwarenaut: Exploring hierarchical system decompositions. In *Proceedings of CSMR 2006 (10th European Conference on Software Maintenance and Reengineering)*, pages 351–354. IEEE CS Press, 2006.

[LL07] Mircea Lungu and Michele Lanza. Reverse engineering through holistic software exploration. In *Proceedings of FAMOOSR 2007 (1st Workshop on FAMIX and Moose in Reengineering)*, 2007.

[LLG06] Mircea Lungu, Michele Lanza, and Tudor Gîrba. Package patterns for visual architecture recovery. In *Proceedings of CSMR 2006 (10th European Conference on Software Maintenance and Reengineering)*, pages 183–192. IEEE CS Press, 2006.

[LLGH07] Mircea Lungu, Michele Lanza, Tudor Gîrba, and Reinout Heeck. Reverse engineering super-repositories. In *Proceedings of WCRE 2007 (14th Working Conference on Reverse Engineering)*, pages 120–129. IEEE CS Press, 2007.

[LM06] Michele Lanza and Radu Marinescu. *Object-Oriented Metrics in Practice*. Springer-Verlag, 2006.

[Mar06] Cristina Marinescu. Identification of design roles for the assessment of design quality in enterprise applications. In *Proceedings of International Conference on Program Comprehension (ICPC 2006)*, pages 169–180, Los Alamitos CA, 2006. IEEE Computer Society Press.

[Mar07a] Cristina Marinescu. Discovering the objectual meaning of foreign key constraints in enterprise applications. In *Proceedings of the 14th Working Conference on Reverse Engineering, Vancouver, Canada*. IEEE Computer Society Press, 2007.

[Mar07b] Cristina Marinescu. Identification of relational discrepancies between database schemas and source-code in enterprice applications. In *Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientifuc Computing, Timisoara, Romania*. IEEE Computer Society Press, 2007.

[MGL06] Michael Meyer, Tudor Gîrba, and Mircea Lungu. Mondrian: An agile visualization framework. In *ACM Symposium on Software Visualization (SoftVis 2006)*, pages 135–144, New York, NY, USA, 2006. ACM Press.

[MGV+07] Petru F. Mihancea, George Ganea, Ioana Verebi, Cristina Marinescu, and Radu Marinescu. Mcc and mc#: Unified c++ and c# design facts extractors tools. In *Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientifuc Computing, Timisoara, Romania*. IEEE Computer Society Press, 2007.

[Mih06] Petru Florin Mihancea. Towards a client driven characterization of class hierarchies. In *Proceedings of International Conference on Program Comprehension (ICPC 2006)*, pages 285–294, Los Alamitos CA, 2006. IEEE Computer Society Press.

[MMM+05] Cristina Marinescu, Radu Marinescu, Petru Florin Mihancea, Daniel Ratiu, and Richard Wettel. iplasma: An integrated platform for quality assessment of object-oriented design. In *ICSM (Industrial and Tool Volume)*, pages 77–80, 2005.

[NDG05] Oscar Nierstrasz, Stéphane Ducasse, and Tudor Gîrba. The story of Moose: an agile reengineering environment. In *Proceedings of the European Software Engineering Conference (ESEC/FSE 2005)*, pages 1–10, New York NY, 2005. ACM Press. Invited paper.

[NKG+07] Oscar Nierstrasz, Markus Kobel, Tudor Gîrba, Michele Lanza, and Horst Bunke. Example-driven reconstruction of software models. In *Proceedings of CSMR 2007 (11th European Conference on Software Maintenance and Reengineering)*, pages 275–286. IEEE CS Press, 2007.

[oR07] The Lab on Reengineering. LORE team, Department of Mathematics and Computer Science, University of Antwerp, Belgium, 2007. http://www.lore.ua.ac.be/Research/.

[TM05] Adrian Trifu and Radu Marinescu. Diagnosing design problems in object oriented systems. In *Proceedings of 12th Working Conference on Reverse Engineering (WCRE 2005), 7-11 November 2005, Pittsburgh, PA, USA*, pages 155–164, Los Alamitos CA, 2005. IEEE Computer Society.

[XMI05] Xml metadata interchange (xmi), v2.0, 2005. http://www.omg.org/cgi-bin/doc?formal/05-05-01.