Intermediate Scientific Report
SNF Project no. 200020-121594
"Bringing Models Closer to Code"

November 3, 2009

a) Summary of results

This project explores various ways of synchronizing software source code with implicit application domain knowledge. The key results achieved in the four tracks of this project in the first year include:

1. **Coordinating models and code:** we have developed novel techniques to dynamically adapt software to non-functional requirements expressed declaratively at the level of the application domain, and we have explored several techniques to mine domain knowledge from source code, bug reports, version repositories, and other resources.

2. **Embedding domain models in the code:** we have developed a framework, called *Helvetia*, for embedding domain-specification languages into general-purpose host languages, and we have explored various novel applications of source code annotations to express high-level domain concepts.

3. **Bringing dynamic models to the IDE:** we have developed novel approaches to integrating dynamic information in the static views of software presented to developers in the development environment, and we have carried out empirical studies to demonstrate the effectiveness of this additional information for supporting typical development and maintenance tasks. We have also developed a novel and efficient back-in time debugger that exploits historical data gathered at run-time.

4. **Model-centric development:** we are developing an experimental language platform called *Pinocchio* in which the language semantics are specified in a fully reflective way. This will enable radical approaches to model-driven development in which semantic models can be manipulated at run-time.

This research has resulted in 5 journal papers, 11 full papers in international, peer-reviewed conferences, as well as numerous other papers, reports and theses.

Results

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

Coordinating models and code

In this track we explore ways to embed multiple models in code and to mine higher-level abstractions from code.

Concurrency control in most programming languages is expressed at a low level of abstraction, and is tangled with functional code. We have started to explore the use of high-level *synchronization specifications* which declaratively express concurrency requirements. A dynamic synchronization system adapts functional code accordingly to meet those requirements [RN09] (in print). We are exploring how an explicit notion of *context* can be used to adapt running code to meet concurrency constraints.

The *vocabulary* of software source code can be a good indicator of the relationship of that code to various domain concepts. *Software cartography*[1] mines vocabulary to establish a graphical map of software components and how they are related to one another [KLN08, Kuh09]. We have investigated the use of vocabulary in bug reports and in software to match developers who are likely to have the right expertise to resolve software defects, and we have empirically validated the quality of the result using historical information from large open-source projects [MKN09] [Mat09] (MSc).

We have also explored other approaches to mining high-level information about software systems from various source. In particular, we have analyzed conference proceedings in the domain of Computer Assisted Orthopedic Surgery (CAOS) to assess the evolution of features over several years [DTNL09]. In another project, we have mined historical information from software repositories to detect patters in the evolution of the software [Jun09] (MSc).

Finally, we have explored ways to reconstruct higher-level models of software (i) by exploiting the internal representations of software generated by the Eclipse IDE [Lan09] (Bachelors project); and (ii) by applying genetic programming to automatically recognize implicit structure in source code [Zan09] (MSc).

Embedding domain models in the code

In this track we explore techniques for expressing higher-level domain models directly in software source code as domain specific languages.

Helvetia[2] is a framework for developing domain-specific languages (DSLs) that can be directly integrated into a general-purpose host language. Helvetia makes use of language transformations to leverage the development tools of the host. Source code editors, syntax highlighting, and debuggers, for instance, work seamlessly with newly introduced DSLs [RDN09] (in print).

In related work, we have explored the use of high-level annotations to declaratively express transactional behavior [RN09] in concurrent code, and to express dynamic adaptation of run-time behavior to different contexts [Str08] (MSc).

Finally, we have continued to explore the use of annotations to express dependencies between test cases [HKN08], and we have carried out an empirical study to validate the effectiveness of dependencies to support defect isolation [Hae09] (MSc; paper submitted for publication).

Bringing dynamic models to the IDE

This track is concerned with the integration of run-time information into the static views of source code typically offered by Integrated Development Environments (IDEs). Empirical validation is a key component of this task.

We have developed a prototype of an IDE, called *Hermion*[3], in which the source code of an application under development is decorated with additional information concerning its run-time behaviour. Applications are instrumented to record actual control flow data, and to correlate software components with end-user features. This information is made available to the developer through a variety of mechanisms (supplementary visualizations, additional navigation widgets, and so on) [RG08a, RG08b, Röt08].

The principle ideas of Hermion have been re-implemented and extended in *Senseo*[4], a plug-in to the popular Eclipse IDE for Java [RHV+09a, RHV+09b]. Senseo makes use of a dedicated Java instrumentation framework developed by the research group of Prof. Walter Binder at the University of Lugano. An extensive empirical study with a large group of professional developers from industry has been carried out to assess the effectiveness of the extended IDE for typical maintenance tasks, and a first paper has been submitted for publication (author notification due December 18).

In related work, we have developed *HeatMaps*, a simple configurable technique for augmenting static views of software with a variety of software metrics [RND+09a] [RND+09b] (Technical report).

[1] `http://scg.unibe.ch/research/softwarecartography`
[2] `http://scg.unibe.ch/research/helvetia`
[3] `http://scg.unibe.ch/research/hermion`
[4] `http://scg.unibe.ch/research/senseo`

In a parallel activity, we have developed a prototype of an efficient back-in-time debugger [LFN09] [Fie09] (MSc) based on the Object Flow Virtual Machine[5] [LDG09] developed by Adrian Lienhard in his PhD thesis [Lie08]. (Dr. Lienhard was funded by the predecessor project, 200020-113342. His PhD thesis was awarded the prestigious Ernst Denert-Stiftung prize for Software Engineering 2009[6].)

Model-centric development

This track explores fundamentally new ways to see software as a direct reflection of executable domain models. We experiment with highly reflective and evolving computational models and programming languages.

We are developing an open and fully reflective language platform called *Pinocchio*[7]. In contrast to typical dynamic systems in which reflective features stop when they encounter the virtual machine boundary, Pinocchio's entire semantics is accessible at run-time. This foundation will enable radical forms of model-centric software analysis to be performed, without the need to develop a dedicated virtual machine (as is the case with the Object Flow Virtual Machine, mentioned in the previous track). Pinocchio is under intensive development since a few months, and no publications are in the pipeline yet, though numerous papers concerning many of the underlying principles have appeared in print, as described below.

SchemeTalk is a proof-of-concept prototype of Pinocchio's bootstrapped meta-modeling approach to VM implementation [VR09].

PyGirl is an experimental VM prototype of the Nintendo Game Boy realized by means of model-driven development using the PyPy toolchain [BV09] [BVD09] (Technical report) [Bru09] (Bachelors project). A high-level interpreter for the Game Boy was written in Python (a dynamic object-oriented language), and this interpreter was then iteratively transformed using a model-driven approach to a full virtual machine using PyPy, a self-hosting interpreter for Python.

Cell is a programming language model and running prototype in which new programming abstractions can be built from fine-grained fragments called "cells" [KN08].

The Calculus of Evolving Objects [DCGN08] is a first attempt to develop a formal foundation for highly reflective and dynamically evolving object systems, such as Pinocchio. This work was carried out in collaboration with researchers at the Università di Torino and the Università del Piemonte Orientale.

We have also published several papers outlining our vision of model-centric software development and its underlying foundations [NDG+08, NDR09, DGKR09].

Staff contributions

- Fabrizio Perin was initially hired to work on this SNF project, but due to his research interests and expertise, he was moved to the ongoing Hasler Foundation project, "Enabling the evolution of J2EE applications through reverse engineering and quality assurance"[8]. Mr Verwaest, who was initially working for the Hasler project, for identical reasons, moved to the SNF project. Mr Perin is making excellent progress in the Hasler project.

- Lukas Renggli has completed the third year of his PhD, and is rapidly approaching completion of his dissertation. He has developed a novel approach to integrating internal DSLs within a general purpose programming language, and is in the process of completing various case studies to validate the technical contributions. He has been extremely productive, participating in numerous paper related to his research, either as first author or as co-author [DGKR09, NDR09, RN09, RG09, RDN09, VR09]. He is expected to defend his thesis in the Summer of 2010.

- Jorge Ressia has completed the first year of his PhD. He is working on the track "Coordinating models and code", and has made good progress in developing a declarative approach to adapt software dynamically to domain concerns. Initial results are reported in an international workshop paper

[5] http://scg.unibe.ch/research/objectflow
[6] http://www.denert-stiftung.de/preistraegerarchiv
[7] http://scg.unibe.ch/research/pinocchio
[8] http://scg.unibe.ch/research/hasler07

[RN09]. Mr Ressia also participated in the organisation of CASTA[9], the first international workshop on Context-Aware Software Technology and Applications, co-located with the European Software Engineering Conference in Amsterdam this year.

- David Röthlisberger has completed the third year of his PhD, and is expected to complete his PhD in the coming months. He has invested considerable effort in developing novel techniques to bring dynamic information to the IDE, and in carrying out extensive user studies to empirically validate the value of this information for improving developer productivity. In the past year he has collaborated closely with the research group of Prof Walter Binder at the University of Lugano, who has developed novel techniques to instrument Java code. Mr Röthlisberger has published extensively on his research [NDG+08, RG08a, RG08b, RND+09a, RHV+09a, RHV+09b, RND+09b, RNDB09, RND09, Röt08]. Mr Röthlisberger also co-organized the PCODA workshop co-located with WCRE 2008 [ZGHLR08].

- Toon Verwaest has completed the second year of his PhD. Mr Verwaest transferred from the Hasler project mentioned above to this project, as his research interests and expertise are more closely aligned with the theme of "Model-centric development". Mr Verwaest has been working intensively in the past months to complete a first prototype of Pinocchio, a fully reflective language platform for experimenting with model-centric development. He has published some initial results related to this work, and is expected to make good progress in the coming months [BVD09, BV09, VR09].

Changes to the research plan

Toon Verwaest replaced Fabrizio Perin in the track on *Model-Centric Development*.

Important events

- We initiated collaboration with the research group of Prof Walter Binder at the University of Lugano in the track on *Bringing Dynamic Models to the IDE*

- Oscar Nierstrasz was a program committee member of the international conferences OOPSLA Onward! 2008, ETSM 2009 and ECOOP 2009, and the international workshops FAMOOSr 2008, Models@run.time, SUITE 2009, RAM-SE 2009, IWST09 and FOSD 2009

- Oscar Nierstrasz was program chair of the CASTA international workshop on Context-Aware Software Technology and Applications (Amsterdam) [Nie09]

- Lukas Renggli was a program committee member of the IDM 2009 conference and the IWST 2009 international workshop

- Jorge Ressia co-organized the CASTA workshop (Amsterdam)

- David Röthlisberger co-organized the PCODA international workshop on Program Comprehension through Dynamic Analysis (Antwerp) [ZGHLR08]

[9] `http://casta.unibe.ch`

b) Publications

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

`http://scg.unibe.ch/scgbib?query=snf09`

Please note that proceedings, theses, technical reports and student projects are *not* included with this report, but are nevertheless available electronically from the above URL. Publications in press will be delivered with the final report.

Published papers

[BV09] Camillo Bruni and Toon Verwaest. PyGirl: Generating whole-system VMs from high-level prototypes using PyPy. In *Objects, Components, Models and Patterns, Proceedings of TOOLS Europe 2009*, volume 33 of *LNBIP*, pages 328–347. Springer-Verlag, 2009.

[DCGN08] Mariangiola Dezani-Ciancaglini, Paola Giannini, and Oscar Nierstrasz. A calculus of evolving objects. *Scientific Annals of Computer Science*, XVIII:63–98, 2008.

[DGKR09] Stéphane Ducasse, Tudor Gîrba, Adrian Kuhn, and Lukas Renggli. Meta-environment and executable meta-language using Smalltalk: an experience report. *Journal of Software and Systems Modeling (SOSYM)*, 8(1):5–19, February 2009.

[DTNL09] Gisèle Douta, Haydar Talib, Oscar Nierstrasz, and Frank Langlotz. CompAS: A new approach to commonality and variability analysis with applications in computer assisted orthopaedic surgery. *Information and Software Technology*, 51(2):448–459, 2009.

[HKN08] Lea Haensenberger, Adrian Kuhn, and Oscar Nierstrasz. Using dynamic analysis for API migration. In *Proceedings IEEE Workshop on Program Comprehension through Dynamic Analysis (PCODA 2008)*, pages 32–36, October 2008.

[KLN08] Adrian Kuhn, Peter Loretan, and Oscar Nierstrasz. Consistent layout for thematic software maps. In *Proceedings of 15th Working Conference on Reverse Engineering (WCRE'08)*, pages 209–218, Los Alamitos CA, October 2008. IEEE Computer Society Press.

[KN08] Adrian Kuhn and Oscar Nierstrasz. Composing new abstractions from object fragments. In Hridesh Rajan, editor, *Proceedings of the 2nd Workshop on Virtual Machines and Intermediate Languages for Emerging Modularization Mechanisms (VMIL 2008), Nashville, Tennessee, Oct. 19, 2008*, pages 1–12, 2008.

[Kuh09] Adrian Kuhn. Automatic labeling of software components and their evolution using log-likelihood ratio of word frequencies in source code. In *MSR '09: Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories*, pages 175–178. IEEE, 2009.

[LDG09] Adrian Lienhard, Stéphane Ducasse, and Tudor Gîrba. Taking an object-centric view on dynamic information with object flow analysis. *Journal of Computer Languages, Systems and Structures*, 35(1):63–79, 2009.

[LFN09] Adrian Lienhard, Julien Fierz, and Oscar Nierstrasz. Flow-centric, back-in-time debugging. In *Objects, Components, Models and Patterns, Proceedings of TOOLS Europe 2009*, volume 33 of *LNBIP*, pages 272–288. Springer-Verlag, 2009.

[MKN09] Dominique Matter, Adrian Kuhn, and Oscar Nierstrasz. Assigning bug reports using a vocabulary-based expertise model of developers. In *MSR '09: Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories*, pages 131–140. IEEE, 2009.

[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.

[NDR09] Oscar Nierstrasz, Marcus Denker, and Lukas Renggli. Model-centric, context-aware software adaptation. In Betty H.C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *LNCS*, pages 128–145. Springer-Verlag, 2009.

[RG08a] David Röthlisberger and Orla Greevy. Representing and integrating dynamic collaborations in IDEs. In *Proceedings of the 15th Working Conference on Reverse Engineering (WCRE 2008)*, pages 74–78, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[RG08b] David Röthlisberger and Orla Greevy. Towards seamless and ubiquitous availability of dynamic information in IDEs. In *Proceedings of the 4th International Workshop on Program Comprehension through Dynamic Analysis (PCODA 2008)*, pages 27–31. Technische Universiteit Delft, 2008.

[RHV+09a] David Röthlisberger, Marcel Härry, Alex Villazón, Danilo Ansaloni, Walter Binder, Oscar Nierstrasz, and Philippe Moret. Augmenting static source views in IDEs with dynamic metrics. In *Proceedings of the 25th International Conference on Software Maintenance (ICSM 2009)*, pages 253–262, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[RHV+09b] David Röthlisberger, Marcel Härry, Alex Villazón, Danilo Ansaloni, Walter Binder, Oscar Nierstrasz, and Philippe Moret. Senseo: Enriching Eclipse's static source views with dynamic metrics. In *Proceedings of the 25th International Conference on Software Maintenance (ICSM 2009)*, pages 383–384, Los Alamitos, CA, USA, 2009. IEEE Computer Society. Tool demo.

[RN09] Lukas Renggli and Oscar Nierstrasz. Transactional memory in a dynamic language. *Journal of Computer Languages, Systems and Structures*, 35(1):21–30, April 2009.

[RND+09a] David Röthlisberger, Oscar Nierstrasz, Stéphane Ducasse, Damien Pollet, and Romain Robbes. Supporting task-oriented navigation in IDEs with configurable HeatMaps. In *Proceedings of the 17th International Conference on Program Comprehension (ICPC 2009)*, pages 253–257, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[Röt08] David Röthlisberger. Embedding Moose facilities directly in IDEs. In *FAMOOSr, 2nd Workshop on FAMIX and Moose in Reengineering*, 2008.

[VLBN09] Rajesh Vasa, Markus Lumpe, Philip Branch, and Oscar Nierstrasz. Comparative analysis of evolving software systems using the Gini coefficient. In *Proceedings of the 25th International Conference on Software Maintenance (ICSM 2009)*, pages 179–188, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[VR09] Toon Verwaest and Lukas Renggli. Safe reflection through polymorphism. In *CASTA '09: Proceedings of the first international workshop on Context-aware software technology and applications*, pages 21–24, New York, NY, USA, 2009. ACM.

[ZGHLR08] Andy Zaidman, Orla Greevy, Abdelwahab Hamou-Lhadj, and David Röthlisberger. Workshop on program comprehension through dynamic analysis (PCODA). In *Proceedings of IEEE 15th Working Conference on Software Maintenance and Reengineering (WCRE)*, pages 345–346, October 2008.

Proceedings

[Nie09] Oscar Nierstrasz, editor. *CASTA '09: Proceedings of the first international workshop on Context-Aware Software Technology and Applications*, New York, NY, USA, 2009. ACM.

Technical reports

[BVD09] Camillo Bruni, Toon Verwaest, and Marcus Denker. PyGirl: Generating whole-system VMs from high-level models using PyPy. Technical Report IAM-09-002, University of Bern, Institute of Applied Mathematics and Computer Sciences, 2009.

[RND+09b] David Röthlisberger, Oscar Nierstrasz, Stéphane Ducasse, Damien Pollet, and Romain Robbes. Supporting task-oriented navigation in IDEs with configurable HeatMaps. Technical Report IAM-09-005, Institut für Informatik, Universität Bern, Switzerland, July 2009.

Theses and Student projects

[Bru09] Camillo Bruni. Development and debugging of a whole-system VM in RPython. Bachelor's thesis, University of Bern, January 2009.

[Fie09] Julien Fierz. Compass: Flow-centric back-in-time debugging. Master's thesis, University of Bern, January 2009.

[Hae09] Lea Haensenberger. Defect isolation as responsibility of the framework automated API migration from JUnit to JExample. Master's thesis, University of Bern, September 2009.

[Jun09] Matthias Junker. Kumpel: Visual exploration of file histories. Master's thesis, University of Bern, January 2009.

[Lan09] Daniel Langone. Recycling trees: Mapping Eclipse ASTs to Moose models. Bachelor's thesis, University of Bern, January 2009.

[Lie08] Adrian Lienhard. *Dynamic Object Flow Analysis*. Phd thesis, University of Bern, December 2008.

[Mat09] Dominique Matter. Who knows about that bug? — automatic bug report assignment with a vocabulary-based developer expertise model. Master's thesis, University of Bern, June 2009.

[Str08] Anselm Strauss. Dynamic aspects — an AOP implementation for Squeak. Master's thesis, University of Bern, November 2008.

[Zan09] Sandro De Zanet. Grammar generation with genetic programming — evolutionary grammar generation. Master's thesis, University of Bern, July 2009.

c) Publications in press

Publications to appear

[RDN09] Lukas Renggli, Marcus Denker, and Oscar Nierstrasz. Language boxes: Bending the host language with modular language changes. In *Software Language Engineering: Second International Conference, SLE 2009, Denver, Colorado, October 5-6, 2009*, LNCS. Springer, 2009. To appear.

[RG09] Lukas Renggli and Tudor Gîrba. Why Smalltalk wins the host languages shootout. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2009)*. ACM Digital Library, 2009. To appear.

[RN09] Jorge Ressia and Oscar Nierstrasz. Dynamic synchronization — a synchronization model through behavioral reflection. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2009)*. ACM Digital Library, 2009. To appear.

[RND09] David Röthlisberger, Oscar Nierstrasz, and Stéphane Ducasse. Autumn leaves: Curing the window plague in IDEs. In *Proceedings of the 16th Working Conference on Reverse Engineering (WCRE 2009)*, pages 237–246, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[RNDB09] David Röthlisberger, Oscar Nierstrasz, Stéphane Ducasse, and Alexandre Bergel. Tackling software navigation issues of the Smalltalk IDE. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2009)*. ACM Digital Library, 2009. To appear.