Intermediate Scientific Report SNF Project no. 200020-131827 *"Synchronizing Models and Code"*

November 10, 2011

a) Summary of results

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

1. **Dynamic meta-objects:** A running system should be able to dynamically respond to changes in its environment (fine-grained and short-term adaptations). Most common approaches to realize runtime adaptation are low-level, heterogenous, and unsuitable for composition. We have developed Bifröst which models Dynamic meta-objects. Dynamic meta-objects provide the means for a clear and simple interface where objects and meta-objects coexist. Meta-objects define adaptations on any object, including meta-objects, providing a homogenous approach.

   As a result of the application of Bifröst model we were able to address challenging problems in a unified manner. We have proposed a dynamic reuse approach called talents. Domain-specific profiling was developed to address the gap between traditional profilers and domain abstractions. We developed Subjectivity which subsumes existing subjective behavior approaches. All these approaches have been realized as working applications which have been used on real world systems. We have also published on each of this approaches on international reviewed conferences and journals.

2. **First-class, active contexts:** To be safe, dynamic updates are usually conducted globally to the system, and need to be well-timed. Instead, first-class contexts enable to incrementally update running applications which relaxes the timing constraint. Contexts can be loaded, instantiated and activated programmatically. The behavior and state of objects is dynamically adapted to reflect the active context and objects can by consequence safely interoperate, even when several contexts coexist.

   We have defined a Smalltalk dialect which implements first-class contexts, and built an initial version of ActiveContext which demonstrates the benefits of our approach. We have also published the details of the approach in a peer-reviewed international workshop.

3. **Linked, active source code:**

   Current software development tools fail to address synchronization of code shared between independent systems (medium-grained and medium-term as well as long-term adaptations). As software evolves, libraries, components and even fragments are frequently duplicated, adapted and specialized across software projects. We have built up a database of all versions of all source code on the Squeaksource repository. This enables us to connect identical source snippets across repositories. It also allows us to name the amount of code duplication in the ecosystem, across repositories.

   To ease the implementation of this project, we improved on current dependency injection frameworks by making ours part of the programming language. We have brought this approach to both Java and Smalltalk. As a result, our code is more testable and more modular.

This research has resulted in 9 full papers in international, peer-reviewed conferences, 6 peer-reviewed international workshop papers, and several theses.

Results

We present the results obtained during the period from 2010-10-01 to 2011-09-30.

Dynamic meta-objects

In this track we are exploring techniques to dynamically adapt running systems to changing requirements. To this end, we have been developing *Bifröst* [1], a fine-grained approach to structural and behavioral reflection based on *explicit meta-objects*. A unique feature of Bifröst is the ability to *compose* adaptations by combining multiple metaobjects, each of which brings its own adaptations. We have explored the application of the Bifröst model to several challenging problems.

Talents are dynamically composable units of reuse that can be applied to individual objects at runtime. Like *stateful traits*, talents can encapsulate sets of related methods and instance variables for reuse in new contexts. Unlike traits, talents can be applied at runtime, and can be applied to individual objects, not just to classes. This work was presented at IWST 2011 (International Workshop on Smalltalk Technologies) [RGN$^+$11]. An extended journal version of this paper is in preparation.

Domain-specific profiling refers to the ability to adapt the activity of profiling an application to the needs of a particular application domain. Traditional profiling tools perform well on host language code itself but they often fail to provide meaningful results related to a domain abstractions. MetaSpy is an infrastructure that enables developers to quickly prototype new profilers for their domain-specific languages and models. Bifröst meta-objects are used for instrumenting the domain to feed the profilers with their required data. This work was presented at TOOLS 2011 [BNRR11], a long-standing international conference in this domain.

Subjectopia provides a unification of different approaches to subjective behavior. Many different solutions have been proposed in the past, based, for example, on perspectives, roles, contextual layers, and force trees. Although these approaches are somehow equally expressive, each imposes a particular world view which may not be appropriate for all applications. Subjectopia subsumes existing approaches, provides a more general foundation for modeling subjective behavior, and offers a means to alter subjective behavior in a running system. Bifröst meta-objects are used to adapt particular object to subjects at runtime. The required behavior and structure is added to the object, moving it away from the objective behavior imposed by its class. This work was presented at TOOLS 2011[LRN11]. Full details are provided in the Masters thesis of Daniel Langone [Lan11].

In related work, we analyzed the use of dynamic features in languages like Smalltalk [CRTR11].

First-class, active contexts

Here we are exploring techniques to dynamically update running software systems. Updating a software system typically entails modifying the software, stopping the running instance, migrating its data to new formats, and then restarting it. For certain critical software systems, stopping the running system to migrate it may not be an attractive option. We have been exploring the use of first-class contexts to enable dynamic updates of running software systems.

ActiveContext makes explicit the context in which a single, consistent instance of a software system runs. Modifications to the system trigger the creation of a new context. Objects of the running application may then exist simultaneously in multiple contexts. Clients using the system in one context are not interrupted when updates are installed. New clients automatically join the newer context and the old context can be eventually garbage-collected when all clients have left the old context. Bidirectional transformations are used to keep these multiple versions in sync. ActiveContext enables incremental updates of running applications.

We have defined a Smalltalk dialect with first-class context, and built an initial version of ActiveContext using *Pinocchio*[2]. Pinocchio is a Smalltalk virtual machine that fully reifies the language interpretation. It was extended with the support of first-class contexts. This work has been presented at IWST 2011 (International Workshop on Smalltalk Technologies) [WGN11]. This initial working version demonstrates

[1] http://scg.unibe.ch/research/bifrost

[2] http://scg.unibe.ch/research/pinocchio

the key benefits of our approach: lack of timing constraints, safety, and flexibility. An extended journal version of this paper is in preparation. Full details of this work are presented in the Masters thesis of David Gurtner [Gur11].

In related work, we explored the use of *modification models* to specify how runtime updates translate to low-level changes in the implementation. This work was presented at RAM-SE'11 (8th Workshop on Reflection, AOP and Meta-Data for Software Evolution) [VSW11].

Linked, active source code

In this track we are exploring means to link source code across projects and repositories to facilitate program understanding for software development and maintenance tasks. Connecting developers across projects and repositories gives developers information from outside their repository that is useful within their repository. We approach our goal from two ends: from the one end, we search for the most useful information, and present it to the developer in the best way. From the other end, we increase the usefulness of that information.

Seuss offers a means to disentangle dependencies between source code entities with the help of dependency injection. This work was presented at TOOLS 2011 [SLN11], where it was short-listed for the best paper award. Seuss improves modularity of separate parts of the source code. Modularity is key to outside information being useful. It allows developers from other projects and repositories to use only the part of our project that they need, thus clearing the way to fruitful collaboration between two projects.

DoodleDebug associates objects with their graphical representations. It is an alternative to the popular toString method, and improves on it by better handling composed objects. Initial results were presented at DYLA'11 (5th Workshop on Dynamic Languages and Applications) [Sch11].

Software Cartography offers means to graphically represent complex software systems using a map metaphor, by clustering software artifacts that are closely related. This work was presented at SOFTVIS 2010 [KEN10], and was described in detail in the PhD thesis of Adrian Kuhn [Kuh11] and the diploma thesis of Peter Loretan [Lor11].

SmartGroups aids program comprehension by automatically associating related software artifacts visited during development and maintenance tasks. In a related work in collaboration with the University of Lugano, we explored the usefulness of recording and replaying the developer interactions in the IDE to support answering developer questions related to software maintenance tasks [HDLL11]. Both these works were presented at ICPC 2011 (International Conference on Program Comprehension).

Bringing the evolutionary analysis to the next level of abstraction, in collaboration with the University of Chile, we studied the *ripple effects* that changes to APIs produce inside complex software ecosystems. This work was presented at ICSE 2011 [RL11].

Staff contributions

- Jorge Ressia has completed the third year of his PhD. He has developed the Bifröst model of structural and behavioral reflection based on explicit metaobjects, and has been exploring its application to various domains, such as instrumentation, debugging, profiling, and subjective behavior. He has continued to publish peer-reviewed papers in dedicated conferences [RGN$^+$11, BNRR11, LRN11, PRR10]. He has also co-supervised the Masters thesis of Daniel Langone on subjective behavior [Lan11], and is co-supervising the Masters thesis of Andrea Quadri on an event-based approach to instrumentation. Ressia is scheduled to complete his PhD thesis in the next twelve months.

- Niko Schwarz has completed the second year of his PhD. He is working on *linked, active source code*. He has co-authored three papers over the past year related to this topic [SLN11, Sch11, VSW11].

  He led the project that built up a database of all versions of all source code on the squeaksource repository. It enables us to connect identical source snippets across repositories. It also allows us to name the amount of code duplication in the ecosystem, across repositories. He is currently expanding this work to include all major open source projects.

  He is supervising the Master's theses of Cedric Reichenbach and Simon Vogt. He is co-supervising the Master's thesis of Nicole Haenni.

- Erwann Wernli has completed the second year of his PhD. He is working on *first-class, active contexts*. He has co-authored two related papers in the past year [WGN11, VSW11], and he has co-supervised the Masters thesis of David Gurtner [Gur11].

Changes to the research plan

The proposed track on *Polyglot systems modeling and analyses* was not accepted for funding, and we do not report on those activities here.

Important events

- Oscar Nierstrasz was a keynote speaker for TOOLS 2011 (49th International Conference on Objects, Models, Components and Patterns)
- Oscar Nierstrasz was co-organizer of the Dagstuhl Workshop 11021 on Feature-Oriented Software Development
- Oscar Nierstrasz was a program committee member of the international conferences ICSM 2011 (27th IEEE International Conference on Software Maintenance), ICMT 2011 (International Conference on Model Transformation), CSMR 2011 (5th European Conference on Software Maintenance and Reengineering), SE 2011 (Software Engineering 2011), C5 (The Ninth International Conference on Creating, Connecting and Collaborating through Computing), and OOPS 2011.
- Mircea Lungu was program co-chair for the Tool Demonstrations Track of WCRE 2011 (The 18th Working Conference on Reverse Engineering)
- Mircea Lungu was a program committee member of: WSE 2011 (The 13th IEEE International Symposium on Web Systems Evolution); IWPSE-EVOL 2011 (The Joint ERCIM Workshop on Software Evolution and International Workshop on Principles of Software Evolution); The Tool Demonstration Track of ICSM 2011 (27th IEEE International Conference on Software Maintenance); The Tool Demonstration Track of ESEC/FSE 2011 (the 8th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering)
- Mircea Lungu was a reviewer for the Elsevier Science of Computer Programming Journal and the Journal of Systems and Software
- Jorge Ressia was a program committee member of the international workshops MDSM 2011 (First International Workshop on Model-Driven Software Migration at CSMR2011), DYLA 2011 (5th Workshop on Dynamic Languages and Applications at TOOLS2011).
- Jorge Ressia was a coreviewer of the international conferences and workshops CSMR 2011 (European Conference on Software Maintenance and Reengineering), SE 2011 (Software Engineering), Onward! 2011 (ACM Symposium on New Ideas in Programming and Reflections on Software), ICMT 2011 (International Conference on Model Transformation), ICSM 2011 (International Conference on Software Maintenance), IWST 2011 (International Workshop on Smalltalk Technologies).
- Erwann Wernli was a coreviewer of the international conferences and workshops SE 2011 (Software Engineering), Onward! 2011 (ACM Symposium on New Ideas in Programming and Reflections on Software), OOPSLA 2011 (Object-Oriented Programming, Systems, Languages & Applications), ICMT 2011 (International Conference on Model Transformation), ICSM 2011 (International Conference on Software Maintenance).
- Niko Schwarz was a coreviewer of the international conferences and workshops SE 2011 (Software Engineering), Onward! 2011 (ACM Symposium on New Ideas in Programming and Reflections on Software), OOPSLA 2011 (Object-Oriented Programming, Systems, Languages & Applications), ICMT 2011 (International Conference on Model Transformation), ICSM 2011 (International Conference on Software Maintenance).

b) Research output

All reported publications are available electronically at the following url:
`http://scg.unibe.ch/scgbib?query=snf11&sortBy=category`

Published papers

[BNRR11] Alexandre Bergel, Oscar Nierstrasz, Lukas Renggli, and Jorge Ressia. Domain-specific profiling. In *Proceedings of the 49th International Conference on Objects, Models, Components and Patterns (TOOLS'11)*, volume 6705 of *LNCS*, pages 68–82. Springer-Verlag, June 2011. Available from: http://scg.unibe.ch/archive/papers/Berg11b-Profiling.pdf, doi:10.1007/978-3-642-21952-8_7.

[CRTR11] Oscar Callaú, Romain Robbes, Éric Tanter, and David Röthlisberger. How developers use the dynamic features of programming languages: The case of smalltalk. In *Proceedings of the 8th working conference on Mining software repositories (MSR 2011)*, pages 23–32, New York, NY, USA, 2011. IEEE Computer Society. Available from: http://scg.unibe.ch/archive/papers/Call11aDynamicFeaturesMSR2011.pdf, doi:10.1145/1985441.1985448.

[HDLL11] Lile Hattori, Marco D'Ambros, Michele Lanza, and Mircea Lungu. Software evolution comprehension: Replay to the rescue. In *Proceedings of the 19th International Conference on Program Comprehension*, pages 161–170. IEEE Computer Society Press, 2011. Available from: http://inf.unisi.ch/faculty/lanza/Downloads/Hatt2011a.pdf, doi:10.1109/ICPC.2011.39.

[KEN10] Adrian Kuhn, David Erni, and Oscar Nierstrasz. Embedding spatial software visualization in the IDE: an exploratory study. In *SOFTVIS 2010*, pages 113–122, October 2010. Available from: http://scg.unibe.ch/archive/papers/Kuhn10c-softvis2010-kuhn-codemap-userstudy.pdf, doi:10.1145/1879211.1879229.

[Kuh10] Adrian Kuhn. On recommending meaningful names in source and UML. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*, RSSE '10, pages 50–51, New York, NY, USA, 2010. ACM. Available from: http://scg.unibe.ch/archive/papers/Kuhn10d.pdf, doi:10.1145/1808920.1808932.

[LRN11] Daniel Langone, Jorge Ressia, and Oscar Nierstrasz. Unifying subjectivity. In *Proceedings of the 49th International Conference on Objects, Models, Components and Patterns (TOOLS'11)*, volume 6705 of *LNCS*, pages 115–130. Springer-Verlag, June 2011. Available from: http://scg.unibe.ch/archive/papers/Lang11aSubjectivity.pdf, doi:10.1007/978-3-642-21952-8_10.

[PRR10] Fabrizio Perin, Lukas Renggli, and Jorge Ressia. Natural language checking with program checking tools. In *Smalltalks 2010*, 2010. Available from: http://scg.unibe.ch/archive/papers/Peri10eNaturalLanguageChecking.pdf.

[RGN$^+$11] Jorge Ressia, Tudor Gîrba, Oscar Nierstrasz, Fabrizio Perin, and Lukas Renggli. Talents: Dynamically composable units of reuse. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2011)*, pages 109–118, 2011. Available from: http://scg.unibe.ch/archive/papers/Ress11a-Talents.pdf.

[RL11] Romain Robbes and Mircea Lungu. A study of ripple effects in software ecosystems (nier). In *Proceedings of the 33rd International Conference on Software Engineering (ICSE 2011)*, pages 904–907, May 2011. Available from: http://scg.unibe.ch/archive/papers/Robb11aRipples.pdf, doi:10.1145/1985793.1985940.

[RND11] David Röthlisberger, Oscar Nierstrasz, and Stéphane Ducasse. SmartGroups: Focusing on task-relevant source artifacts in ides. In *Proceedings of the 19th International Conference on Program Comprehension (ICPC 2011)*, pages 246–257, Los Alamitos, CA, USA, 2011. IEEE Computer Society. Available from: http://scg.unibe.ch/archive/papers/Roet11bSmartGroupsICPC2011.pdf, doi:10.1109/ICPC.2011.20.

[Sch11] Niko Schwarz. DoodleDebug, objects should sketch themselves for code understanding. In *Proceedings of the TOOLS 2011, 5th Workshop on Dynamic Languages and Applications (DYLA'11).*, 2011. Available from: http://scg.unibe.ch/archive/papers/Schw11bDoodleDebug.pdf.

[SLN11] Niko Schwarz, Mircea Lungu, and Oscar Nierstrasz. Seuss: Cleaning up class responsibilities with language-based dependency injection. In *Objects, Components, Models and Patterns, Proceedings of TOOLS Europe 2011*, volume 33 of *LNCS*, pages 276–289. Springer-Verlag, 2011. Available from: http://scg.unibe.ch/archive/papers/Schw11aSeuss.pdf, doi:10.1007/978-3-642-21952-8_20.

[VBLN11] Toon Verwaest, Camillo Bruni, Mircea Lungu, and Oscar Nierstrasz. Flexible object layouts: enabling lightweight language extensions by intercepting slot access. In *Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications*, OOPSLA '11, pages 959–972, New York, NY, USA, 2011. ACM. Available from: http://scg.unibe.ch/archive/papers/Verw11bFlexibleObjectLayouts.pdf, doi:10.1145/2048066.2048138.

[VSW11] Toon Verwaest, Niko Schwarz, and Erwann Wernli. Runtime class updates using modification models. In *Proceedings of the TOOLS 2011 8th Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE'11)*, 2011. Available from: http://scg.unibe.ch/archive/papers/Verw11aRuntimeUpdates.pdf.

[WGN11] Erwann Wernli, David Gurtner, and Oscar Nierstrasz. Using first-class contexts to realize dynamic software updates. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2011)*, pages 21–31, 2011. http://esug.org/data/ESUG2011/IWST/Proceedings.pdf. Available from: http://scg.unibe.ch/archive/papers/Wern11a-ActiveContext.pdf.

Theses

[Bru11] Camillo Bruni. Optimizing Pinocchio. Master's thesis, University of Bern, January 2011. Available from: http://scg.unibe.ch/archive/masters/Brun11a.pdf.

[Gur11] David Gurtner. Safe dynamic software updates in multi-threaded systems with ActiveContext. Master's thesis, University of Bern, April 2011. Available from: http://scg.unibe.ch/archive/masters/Gurt11a.pdf.

[Kuh11] Adrian Kuhn. *Software Cartography and Code Navigation*. Phd thesis, University of Bern, April 2011. Available from: http://scg.unibe.ch/archive/phd/kuhn-phd.pdf.

[Lan11] Daniel Langone. Subjectopia — unifying subjectivity. Master's thesis, University of Bern, February 2011. Available from: http://scg.unibe.ch/archive/masters/Lang11b.pdf.

[Lor11] Peter Loretan. Software cartography — a prototype for thematic software maps. Diploma thesis, University of Bern, April 2011. Available from: http://scg.unibe.ch/archive/masters/Lore11a.pdf.