Final Scientific Report
SNF Project no. 2000-067855.02
"Tools and Techniques for Decomposing and Composing Software"

November 8, 2004

a) Summary of results

This project addresses the problem of how to organize and structure software systems in such a way that they can be easily adapted to changing requirements. We focus on (1) tools and techniques for extracting architectural artifacts, *i.e.*, for *decomposing* software, and (2) mechanisms and language features for flexibly constructing software from parts, *i.e.*, for *composing* software.

The key results of the second year of this project include (1) empirical studies in applying Concept Analysis to the detection of implicit contracts in complex software systems, (2) first results in the development of a compositional approach to testing that explicitly links methods to their tests, (3) a semantic approach to reasoning about software composition, (4) development of a methodological approach to refactoring object-oriented class hierarchies in terms of *traits*, and (5) investigations into new applications of CLASSBOXes for managing the scope of change.

This project is carried out under close collaboration with RECAST*: Evolution of Object-Oriented Applications* (SNF Project No. 620-066077). Whereas RECAST concentrates more generally on modeling of object-oriented software, program understanding and software evolution, this project focuses on technical issues related to object-oriented languages and language design.

Results

The results obtained in the second year correspond largely with those outlined in the project workplan. We first present results related to software decomposition, and then those related to mechanisms for software composition.

Software Decomposition

The work on applying Formal Concept Analysis to detecting implicit contracts in complex software systems has matured. The technique has been applied to individual classes in order to detect recurring relationships within classes [ADN03]. It has also been applied to sets of classes to detect implicit collaboration patterns occurring throughout a software system [ABN04].

In order to draw a sharper distinction with the RECAST project, activities related more to reverse engineering have been transferred to RECAST, whereas activities that are more code-centric have been transferred to this project. For this reason, although work related to reengineering is continuing within the research group, we report on it in the context of RECAST.

We therefore have shifted emphasis in this research track towards testing strategies. Empirical evidence exists to support the notion that most unit tests can be ordered according to the degree of focus they provide on defects in source code [GLNW04]. We are consequently developing an approach to testing in which methods under test and their corresponding tests are closely correlated

[Gäl04]. A prototype class browser has been developed in which methods are associated to their tests. A central idea is that tests can and should focus on a single method, and that they are allowed to return the affected object of the method being tested. This scheme enables the possibility of composing tests from other test instances.

Software Composition

The work on the Piccola composition language has culminated in publication of semantic foundations of the language [NA03], and an account of how the calculus supports reasoning about compositional styles [AN05]. Various advanced examples have been developed to illustrate how Piccola supports the definition of compositional styles [Spe41]. A central notion to Piccola is the concept of *first-class namespaces* [NA05]. We have found that namespaces are also central to a semantic treatment of both traits and Classboxes, and we plan further pursue this link in the future.

The work on traits has matured considerably. Traits are essentially sets of related methods that can be used to compose classes in a way that avoid problems of fragility introduced by mixins and multiple inheritance. An extensive case study has been carried out, applying traits to the refactoring of the Smalltalk Collection hierarchy [BSD03]. Currently we are using traits to refactor the kernel of the Smalltalk system itself. Traits require methodological and tool support, and we have reported on the principles underlying the traits browser [SB04, BS04].

Traits have also led to new research directions, such as *encapsulation policies* which allow one to specify different policies for accessing and overriding methods for different classes of clients [SDNW04].

Classboxes offer a composition mechanism that functions at a much coarser level of granularity than traits. A classbox defines a scope within which certain class definitions and extensions are visible. A classbox may therefore safely extend (*i.e.*, modify) selected system classes without impacting parts of the system that are outside of the scope of the classbox [BDNW04]. We are also pursuing the possible synergies between traits and classboxes [Min04].

We have begun to explore compositional approaches as applied to application domains such as web applications [Lie03, Ren03, Vog04]. We are also exploring synergies with other, innovative and experimental programming languages, such as FScript [MD03] and Prototalk [BDD04].

Finally, we have been investigating the broader implications of giving a more prominent role to *software evolution* in software development tools, methods and programming languages [Nie04a, Nie04b].

Staff contributions

- Gabriela Arévalo has continued her investigation into applying Formal Concept Analysis as a means to discover implicit relationships and contracts in complex software systems [ADN03, ABN04]. She will defend her PhD thesis in January 2005.

- Alexandre Bergel is continuing his PhD work on the development of Classboxes [BDD04, BDNW04]. A prototype implementation of Classboxes in Smalltalk exhibits excellent performance (10% overhead only where Classboxes are actually used), and is being used as the basis for further investigations, such as the dynamic application of Classboxes to model aspect-oriented programming, and the use of Classboxes to structure related traits into coarser-grained compositional entities. A thesis draft is in progress, and a first draft should be complete in the coming months.

- Markus Gälli has been studying the empirical relationships between units tests and their methods under test. As a result of this investigation, he has been developing an approach to testing in which objects (and their methods) are *explicitly* tied to their corresponding tests [Gäl04, GLNW04]. Future work includes the analysis and refactoring of existing test suites to demonstrate how this explicit connection is not only natural, but supports well the development and debugging processes.

- Laura Ponisio has been studying the impact of dependencies on complex software systems [DLP04]. Most recently, she has been studying how coupling and cohesion metrics in the literature fail to reflect the intuitive notions of coupling and cohesion as they apply, respectively, to object-oriented frameworks and their applications. As this work is more closely related to the goals of the Recast project, her research has been moved to that project, and the results will be reported in the next Recast report.

- Nathanael Schärli has developed traits and encapsulation policies [BS04, BSD03, SB04, SDNW04] in the context of his PhD thesis. He will be defending his PhD in February 2005.

Changes to the research plan

There have been no significant changes in the research plan.

However, in order to draw a clearer distinction between this project and the Recast project led by Prof. Ducasse, all activities related to reengineering are now part of Recast, whereas code-centric activities related to software composition and decomposition are explicitly part of this project and its sequel, *A Unified Approach to Composition and Extensibility* (SNF project No. 200020-105091/1).

Important events

- We have moved Laura Ponisio from this project to Recast, and Markus Gälli from Recast to this project (and its sequel), as it is now clear that their research activities fit better this way to the respective projects. In practice, the only concrete difference this leads to is that their research results will be reported in the context of different projects.

- Markus Gälli presented a paper on testing [GLNW04] at ICSM 2004.

- Nathanael Schärli presented a paper on Traits at ECOOP 2004 [SDNW04].

- Oscar Nierstrasz: Invited Speaker at SI-SE 2004 (Components – The Future of Software Engineering? – University of Zurich, March 18-19, 2004)

- Oscar Nierstrasz: Keynote speaker at CBSE7 (International Symposium on Component-Based Software Engineering – Edinburgh, Scotland, May 24-25, 2004) [Nie04a]

- Oscar Nierstrasz: PC Member of

  - ECOOP 2004 (European Conference for Object-Oriented Programming – Oslo, Norway, June 14-19, 2004)
  - Euromicro CBSE Track (30th Euromicro – Rennes, France, Sept 3, 2004)
  - IWPSE 2004 (International Workshop on Principles of Software Evolution – Kyoto, Japan, Sept. 6-7, 2004)

- Stéphane Ducasse: PC member of

  - ESUG 2004 (European Smalltalk User Group Conference – Bled, Slovenia, 6-10 September, 2004)
  - LMO 2004 (Languages et Models a Objets – Lilles, 17-18 March 2004, France)
  - UML 2004 (7th International Conference on the Unified Modeling Language – October 10-15, 2004, Lisbon, Portugal)
  - JFDLPA (Journée Francophone sur le Développement de Logiciels Par Aspects, 14th September 2004)

b) Publications

Copies of the refereed papers listed below are annexed to this report. Theses, student projects and Recast publications are not included, but, in most cases, are available electronically (together with the published papers) from the following url:

`www.iam.unibe.ch/∼scg/cgi-bin/oobib.cgi?snf04`

Recast publications for 2004 are available from this url:

`www.iam.unibe.ch/∼scg/cgi-bin/oobib.cgi?recast04`

Published papers

[ADN03] Gabriela Arévalo, Stéphane Ducasse, and Oscar Nierstrasz. Understanding classes using X-Ray views. In *Proceedings of 2nd International Workshop on MASPEGHI 2003 (ASE 2003)*, pages 9–18. CRIM - University of Montreal (Canada), October 2003.

[BDD04] Alexandre Bergel, Christophe Dony, and Stéphane Ducasse. Prototalk: an environment for teaching, understanding, designing and prototyping object-oriented languages. In *Proceedings of ESUG Academic Track 2004*, September 2004. PrototalkESUG04.

[BDNW04] Alexandre Bergel, Stéphane Ducasse, Oscar Nierstrasz, and Roel Wuyts. Classboxes: Controlling visibility of class extensions. Technical Report IAM-04-003, Institut für Informatik, Universität Bern, Switzerland, June 2004. ClassboxTR04.

[BS04] Andrew P. Black and Nathanael Schärli. Traits: Tools and methodology. In *Proceedings ICSE 2004*, pages 676–686, May 2004.

[BSD03] Andrew P. Black, Nathanael Schärli, and Stéphane Ducasse. Applying traits to the Smalltalk collection hierarchy. In *Proceedings OOPSLA'03 (International Conference on Object-Oriented Programming Systems, Languages and Applications)*, volume 38, pages 47–64, October 2003.

[Gäl04] Markus Gälli. PhD-symposium: Correlating Unit Tests and Methods under Test. In *5th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2004)*, volume 3092 of *LNCS*, page 317, June 2004.

[GLNW04] Markus Gälli, Michele Lanza, Oscar Nierstrasz, and Roel Wuyts. Ordering broken unit tests for focused debugging. In *20th International Conference on Software Maintenance (ICSM 2004)*, pages 114–123, 2004.

[MD03] Philippe Mougin and Stéphane Ducasse. OOPAL: Integrating array programming in object-oriented programming. In *Proceedings OOPSLA 2003 (International Conference on Object-Oriented Programming Systems, Languages and Applications)*, pages 65–77, October 2003.

[NA03] Oscar Nierstrasz and Franz Achermann. A calculus for modeling software components. In S. Graf F. S. De Boer, M. M. Bonsangue and W-P. de Roever, editors, *FMCO 2002 Proceedings*, volume 2852 of *LNCS*, pages 339–360. Springer-Verlag, 2003.

[NA05] Oscar Nierstrasz and Franz Achermann. Separating concerns with first-class namespaces. In Robert E. Filman, Tzilla Elrad, Siobhán Clarke, and sit Mehmet Ak editors, *Aspect-Oriented Software Development*, pages 243–259. Addison-Wesley, 2005.

[Nie04a] Oscar Nierstrasz. Putting change at the center of the software process. In I. Crnkovic, J.A. Stafford, H.W. Schmidt, and K. Wallnau, editors, *CBSE 2004*, volume 3054 of *LNCS*, pages 1–4. Springer-Verlag, 2004. Extended abstract of an invited talk.

[Nie04b] Oscar Nierstrasz. Software evolution as the key to productivity. In A. Knapp M. Wirsing and S. Balsamo, editors, *Radical Innovations of Software and Systems Engineering in the Future*, volume 2941 of *LNCS*, pages 274–282. Springer-Verlag, 2004.

[SB04] Nathanael Schärli and Andrew P. Black. A browser for incremental programming. *Computer Languages, Systems and Structures*, 30:79–95, 2004.

[SDNW04] Nathanael Schärli, Stéphane Ducasse, Oscar Nierstrasz, and Roel Wuyts. Composable encapsulation policies. In *Proceedings ECOOP 2004 (European Conference on Object-Oriented Programming)*, LNCS 3086, pages 248–274. Springer Verlag, June 2004.

Theses and Student projects

[Lie03] Adrian Lienhard. Mewa: Meta-level architecture for generic web-application construction. Informatikprojekt, University of Bern, November 2003.

[Min04] Florian Minjat. Vers une modélisation transverse et modulaire des collaborations par couplage des traits et des classboxes. DEA, Ecole des mines de Nantes, September 2004.

[Ren03] Lukas Renggli. Smallwiki: Collaborative content management. Informatikprojekt, University of Bern, 2003.

[Spe41] Daria Spescha. Software composition styles: Mixins for Piccola. Informatikprojekt, University of Bern, March 2041.

[Vog04] David Vogel. Management and security of collaborative web environments. Diploma thesis, University of Bern, June 2004.

Selected RECAST publications

[ABN04] Gabriela Arévalo, Frank Buchli, and Oscar Nierstrasz. Detecting implicit collaboration patterns. In *Proceedings of WCRE 2004 (11th Working Conference on Reverse Engineering)*. IEEE Computer Society Press, November 2004. to appear.

[DLP04] Stéphane Ducasse, Michele Lanza, and Laura Ponisio. A top-down program comprehension strategy for packages. Technical Report IAM-04-007, University of Berne, Institut of Applied Mathematics and Computer Sciences, 2004.

c) Publications in press

Publications to appear

[AN05] Franz Achermann and Oscar Nierstrasz. A calculus for reasoning about software components. *Theoretical Computer Science*, 2005. To appear.