ERCIM "Alain Bensoussan" Fellowship Scientific Report

Fellow:	Anthony Cleve
Visited Location :	INRIA Lille Nord Europe (ADAM team)
Duration of Visit:	12 months

I - Scientific activity

My scientific achievements during my ERCIM post-doctoral fellowship at INRIA Lille Nord Europe are various. They are related to different domains, including (dynamic) software product lines, software adaptation, context-aware & self-adaptive systems, data-intensive system evolution, data reverse engineering.

First, I have been involved in an ADAM's research line aiming at bridging the gap between software product lines and software (architecture) adaptation. In particular, we have developed an approach to feature-based architecture composition, that necessitates the cross-checking of feature dependencies and adaptation specification dependencies [4]. We have also proposed to unify design time and runtime adaptations through aspect models [9]. Each aspect model is in turn associated to a feature of the feature model. This approach permits both design time product derivation from feature configuration and runtime (de)selection of features based on context events.

Second, I have developed a comprehensive, automated approach to reverse-engineering architectural variability of plugin-based systems. The approach consists in several steps:
(1) Obtaining a 150% architectural model, by composing all the existing plugins of the system.
(2) Extracting an architectural feature model from the 150% architectural model. This feature model represents the architectural commonalities and variabilities among the family of systems.
(3) Extracting a plugin feature model from the set of plugins and their dependencies
(4) Merging and linking the architectural feature model and the plugin feature model, in order to obtain a full feature model of the system family. The full feature model obtained can then be used to safely configure the system. The selection of architectural features is automatically propagated to the selection of corresponding plugins (and vice versa).
We have then validated the approach based on FraSCAti, a plugin-based platform for runtime adaptation of component-based, service-oriented software systems. FraSCAti can be seen as a dynamic software product line, that can be configured and reconfigured "à la carte".

Third, I have participated in the development of a formal approach to contract-based dynamic software reconfiguration [8]. This approach aims at addressing the current lack of formal foundation for model transformations [3], and is based on e-graph rewriting theory. Software architecture and contracts are modeled as e-graphs. Software reconfigurations are seen as e-graph rewrite rules. Checking fulfillment of contracts and other properties of software reconfiguration at runtime can then reuse the e-graph theory as formal foundation.

Last, I have worked on extending and publishing some of my PhD research results, related to data reverse-engineering [7, 10] and co-evolution of databases and programs [2, 5, 6].

I have also been involved in the organization and/or program committee of several international scientific events:

- Demonstration co-Chair of the *25th IEEE/ACM International Conference on Automated Software Engineering (ASE 2010), Antwerp, Belgium, 2010.*

- Publicity co-Chair and Program Committee member of the *3rd International Conference on Software Language Engineering (SLE 2010),* Eindhoven, The Netherlands, 2010.

- Program co-Chair of the *IWPSE-EVOL 2010*, the *joint International Workshop on Principles of Software Evolution and ERCIM Workshop on Software Evolution and Evolvability*, Antwerp, Belgium, 2010.

- co-Organizer of the *3rd International Workshop on Academic Software Development Tools and Techniques (WASDeTT'2010)* – held in conjunction with – held in conjunction with *ASE'2010,* Antwerp, Belgium, 2010.

- Program Committee member of the *15th European Conference on Software Maintenance and Reengineering (CSMR'2011),* Oldenburg, Germany, 2011.

- Program Committee member of the *17th International Working Conference on Reverse Engineering (WCRE'2010),* USA, 2010

II- Publication(s) during your fellowship

In total, the ERCIM fellowship period has led to 8 publications, including 1 edited proceedings, 2 journal papers, 4 conference papers, and 1 workshop paper. Three additional submissions are currently under review, including 2 journal papers and 1 conference paper.

Edited Proceedings

[1] Andrea Capiluppi, **Anthony Cleve**, and Naouel Moha, editors. *Proceedings of the Joint ERCIM Workshop on Software Evolution and International Workshop on Principles of Software Evolution (IWPSE-EVOL'10).* ACM, New York, NY, USA, 2010.

Abstract : *« The IWPSE-EVOL workshop is the merger of the International Workshop on Principles of Software Evolution (IWPSE) and the annual ERCIM Workshop on Software Evolution (EVOL). The objectives of this joint event is to provide a forum to discuss a wide range of topics in software evolution, to foster the better understanding of the nature of software evolution, and to accelerate research activities on the subject. The 2010 edition of the IWPSE-EVOL focuses on the special theme of automation in the context of software evolution. This theme has been chosen to pay tribute to the host conference, the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'2010). IWPSE-EVOL'2010 attracted 31 submissions. The program committee selected 13 submissions for presentation and publication. The accepted submissions include 5 full research papers, 6 position papers, 1 industrial paper and 1 tool demonstration paper. The acceptance rate for full research papers was 29.4%. The workshop program also features two invited talks by two internationally recognized experts in software maintenance and evolution : Andrian Marcus and Massimiliano Di Penta.»*

Journal Papers

[2] **Anthony Cleve**, Tom Mens, and Jean-Luc Hainaut. Data-Intensive System Evolution. *IEEE Computer*, 43(8):110–112, August 2010.

Abstract : *« What happens if you combine the virtues of software engineering with database engineering? Developers managing the evolution of data-intensive software systems face numerous crucial challenges. »*

[3] Gabriel Tamura and **Anthony Cleve**. A Comparison of Taxonomies for Model Transformation Languages. *Paradigma*, 4(1), March 2010. Revista electronica en construccion de software. ISSN: 2011-0065.

Abstract : *« Since the introduction of the MDE/MDA/MDD ideas for software systems development several years ago, a number of different (meta)modeling and model transformation languages have been proposed. Although the OMG's QVT standard specification has somewhat stabilized the number of new model transformation languages, it is likely that new ones will continue to appear, following different paradigms and approaches. However, the evolution towards the consolidation of models as a unifying, foundational and consistent concept for software-intensive system development, requires the realization of a set of ideal characteristics for the specification of model transformation languages. Several works have been proposed for defining characterization and classification schemes, and the set of these ideal characteristics. In this paper we present a comparison of characterization and classification schemes for model transformation languages, and an analysis of the OMG's QVT specification with respect to this set of ideal characteristics. Although the MOF 2 QVT specification presents a good coverage of these characteristics, it still lacks a formal foundation, from which it could obtain several benefits. »*

Conference Papers

[4] Carlos Parra, **Anthony Cleve**, Xavier Blanc, and Laurence Duchien. Feature-based Composition of Software Architectures. In *Proceedings of the 4th European Conference on Software Architercture (ECSA'2010)*, volume 6285 of *Lecture Notes in Computer Science*, pages 230–245. Springer, 2010.

Abstract : *« In the context of Software Product Lines, variability refers to the de_nition and utilization of di_erences between several products. A well-known approach to expressing variability is by means of Feature Diagrams (FD), which allow designers to express commonalities and variabilities among the members of the same product family. Although feature diagrams are a useful tool in the automated derivation of software products, this process may prove highly complex due to possible interactions between the selected features and the artifacts that realize them. Deriving a concrete product typically involves the composition of such inter-dependent software artifacts belonging to di_erent levels of abstraction. This paper presents a feature-based composition approach allowing to automatically derive a product architecture from a given feature configuration. The proposed approach relies on the combination of Model-Driven Engineering (MDE) and Aspect-Oriented Modeling (AOM) techniques. We introduce a metamodel allowing to reify each feature as a high-level aspect model. Product derivation is achieved by weaving the set of aspect models corresponding to a particular feature configuration. The weaving strategy is derived from an in-depth cross-analysis of both the feature interactions and the aspect model dependencies. »*

[5] **Anthony Cleve**. Program Analysis and Transformation for Data-Intensive System Evolution. In Michele Lanza and Andrian Marcus, editors, *Proceedings of the 26th IEEE International Conference on Software Maintenance (ICSM'2010)*. IEEE Computer Society, 2010.

Abstract : *« Data-intensive software systems are generally made of a database and a collection of application programs in strong interaction with the former. They constitute critical assets in most enterprises, since they support business activities in all production and management domains. Data-intensive systems form most of the so-called legacy systems: they typically are one or more decades old, they are very large, heterogeneous and highly complex. Many of them significantly resist modifications and change due to the lack of documentation, to the use of aging technologies and to inflexible architectures. Therefore, the evolution of data-intensive systems clearly calls for automated support. This paper explores the use of automated program analysis and transformation techniques in support to the evolution of the database component of the system. The program analysis techniques aim to ease the database evolution process, by helping the developers to understand the data structures that are to be changed, despite the lack of precise and up-to-date documentation. The objective of the program transformation techniques is to support the adaptation of the application programs to the new database. This adaptation process is studied in the context of two realistic database evolution scenarios, namely database database schema refactoring and database platform migration. »*

[6] **Anthony Cleve**, Anne-France Brogneaux, and Jean-Luc Hainaut. A Conceptual Approach to Database Applications Evolution. In *Proceedings of the 27th International Conference on Conceptual Modeling (ER'2010)*, volume 6412 of *Lecture Notes in Computer Science*, pages 132–145. Springer, 2010.

Abstract : *« Data-intensive systems are subject to continuous evolution that translates ever-changing business and technical requirements. System evolution usually constitutes a highly complex, expensive and risky process. This holds, in particular, when the evolution involves database schema changes, which in turn impact on data instances and application programs. This paper presents a comprehensive approach that supports the rapid development and the graceful evolution of data-intensive applications. The approach combines the automated derivation of a relational database from a conceptual schema, and the automated generation of a data manipulation API providing programs with a conceptual view of the relational database. The derivation of the database is achieved through a systematic transformation process, keeping track of the mapping between the successive versions of the schema. The generation of the conceptual API exploits the mapping between the conceptual and logical schemas. Database schema changes are propagated as conceptual API regeneration so that application programs are protected against changes that preserve the semantics of their view on the data. The paper describes the application of the approach to the development of an e-health system, built on a highly evolutive database.*

[7] Ravi Ramdoyal, **Anthony Cleve**, and Jean-Luc Hainaut. Reverse Engineering User Interfaces for Interactive Database Conceptual Analysis. In *Proceedings of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE'2010)*, volume 6051 of *Lecture Notes in Computer Science*, pages 332–347. Springer, 2010.

Abstract : *« The first step of most database design methodologies consists in eliciting part of the user requirements from various sources such as user interviews and corporate documents. These requirements formalize into a conceptual schema of the application domain, that has proved to be difficult to validate, especially since the visual representation of the ER model has*

shown understandability limitations from the end-users standpoint. In contrast, we claim that prototypical user interfaces can be used as a two-way channel to efficiently express, capture and validate data requirements. Considering these interfaces as a possibly populated physical view on the database to be developed, reverse engineering techniques can be applied to derive their underlying conceptual schema. We present an interactive tool-supported approach to derive data requirements from user interfaces. This approach, based on an intensive user involvement, addresses a significant subset of data requirements, especially when combined with other requirement elicitation techniques. »

Workshop Papers

[8] Gabriel Tamura, Ruby Casallas, **Anthony Cleve**, and Laurence Duchien. QoS Contracts-Aware Reconfiguration of Component Architectures using e-graphs. In *Proceedings of the 7th International Workshop on Formal Aspects of Component Software (FACS'2010)*, Lecture Notes in Computer Science. Springer, 2010. (to appear)

Abstract : *« In this paper we focus on the formalization of component based architecture self-reconguration as an action associated to quality-of-service (QoS) contracts violation. With this, we aim to develop on the vision of the component-based software engineering (CBSE) as a generator of software artifacts responsible for QoS contracts. This formalization, together with a definition of a QoS contract, forms the basis of the framework we propose to enable a system to preserve its QoS contracts. Our approach is built on a theory of extended graph (e-graph) rewriting as a formalism to represent QoS contracts, component-based architectural structures and architecture recon_guration. We use a rule-based strategy for the extensible part of our framework. The reconfiguration rules are expressed as e-graph rewriting rules whose left and right hand sides can be used to encode design patterns for addressing QoS properties. These rules, given by a QoS property domain expert, are checked as safe, i.e., terminating and con_uent, before its application by graph pattern-matching over the runtime representation of the system. »*

Papers submitted for publication

[9] Carlos Parra, Xavier Blanc, **Anthony Cleve**, and Laurance Duchien. Unifying Design and Runtime Software Adaptation Using Aspect Models. Submitted for publication in Elsevier's Science of Computer Programming journal (SCP). 2010.

Abstract : *« Software systems are seen more and more as evolutive systems. At the design phase, software is constantly in adaptation by the building process itself, and at runtime, it can be adapted in response to changing conditions in the executing environment such as location or resources. Adaptation is generally difficult to specify because of its crosscutting impact on software. This article introduces an approach to unify adaptation at design and at runtime based on Aspect Oriented Modeling. Our approach proposes a unified aspect metamodel and a platform that realizes two different weaving processes to achieve design and runtime adaptations. This approach is used in a Dynamic Software Product Line which derives products that can be configured at design time and adapted at runtime in order to dynamically fit new requirements or resource changes. Such products are implemented using the Service Component Architecture and Java. Finally, we illustrate the use of our approach based on an adaptive e-shopping scenario. The main advantages of this unification are: a clear separation of concerns, the selfcontained aspect model that can be weaved during the design and execution, and the platform independence guaranteed by two different types of weaving. »*

[10] **Anthony Cleve**, Jean-Roch Meurisse, and Jean-Luc Hainaut. Database Semantics Recovery through Dynamic Analysis of SQL Statements. Submitted for publication in Springer's Journal on Data Semantics (JoDS). 2010.

Abstract : *« The documentation of a database includes its conceptual schema, that formalizes the semantics of the data, and its logical schema that translates the former according to an operational database model. Important engineering processes such as database and program maintenance and evolution rely on a complete and acurate database documentation. In many cases, however, these schemas are incomplète and outdated or even do not exist anymore. Their reconstruction is the goal of a specific process called database reverse engineering, that comprises two main sub-processes, namely the reconstruction of the logical schema and its conceptualization, that derives a conceptual schema from the former schema. The logical schema is recovered through DDL code analysis but, more important, by discovering constructs (data structures and constraints) that have been incompletely translated in the operational database schema. Many databases, for example, do not declare compound columns or foreign keys explicitly, therefore leaving them implicit. One of the richest discovery techniques is the analysis of the source code of the application programs, and in particular the SQL statements, for which efficient techniques have been designed in the recent years. However, the increasing use of dynamic SQL makes them inoperative, leading to the use of more complex techniques based on runtime dynamic analysis. This paper explains how the analysis of dynamic SQL statements can improve the semantic precision of database schemas by augmenting them with implicit constructs. It first describes the role of code analysis in database reverse engineering, compares static and dynamic SQL interfaces and examines the characteristics of various SQL analysis techniques. Then, it describes in more detail the steps of dynamic analysis of dynamic SQL, namely SQL statement capture and SQL statement analysis, with a particular emphasis on chains of SQL statements. A real world case study is presented and discussed.*

[11] Alberto Bacchelli, **Anthony Cleve**, Andrea Mocci, and Michele Lanza. Recovering Structured Knowledge from Natural Language Artifacts for Software Analysis. Submitted for publication to an international conference. 2010.

Abstract : *« The design and evolution of a software system leave traces in various kinds of artifacts. In software, produced by humans for humans, many artifacts are written in natural language by people involved in the project. Such entities contain structured information which constitute an important source of knowledge for the analysis and comprehension of a system's design and evolution. Nevertheless, the ambiguous and informal nature of narrative is a serious challenge in gathering such information, that is scattered throughout natural language sentences. We present an approach - based on island grammars- to recognize, parse, and model fragments of structured information that occur in natural language artifacts. The resulting model can be exploited to conduct further research. We evaluate our approach by applying it to mailing lists pertaining to several software systems. We show that structured data can be extracted from e-mails with a high precision and recall. We describe how we reconstruct a partial model of a system based on the data extracted from a mailing list, and discuss how such information can be used to perform different software analyses. »*

III -Attended Seminars, Workshops, and Conferences

- *29th International Conference on Conceptual Modeling (ER'2010),* Vancouver, Canada, November 1-4, 2010.

- *25th IEEE/ACM International Conference on Automated Software Engineering (ASE'2010),* Antwerp, Belgium, September 20-24, 2010.
- *Joint International Workshop on Principles of Software Evolution and annual ERCIM Workshop on Software Evolution (IWPSE-EVOL 2010),* Antwerp, Belgium, September 20-21, 2010.

- *3rd International Workshop on Academic Software Development Tools and Techniques (WASDeTT'2010), Antwerp, Belgium, September 21, 2010.*

- *26th IEEE International Conference on Software Maintenance (ICSM'2010),* Timisoara, Romania, September 12-18, 2010.

- *2nd International Workshop on Model-Driven Product Line Engineering (MDPLE 2010), Paris, France, June 16, 2010.*

- *Seminar on Advanced Tools & Techniques for Software Evolution (SATToSE 2010), Montpellier, France, April 26-29, 2010.*

IV – Research Exchange Programme (12 month scheme)

June 7-12, 2010 University of Lugano, Switzerland: research visit in the REVEAL team, led by Prof. Michele Lanza.

During this exciting visit, I have been working in collaboration with Alberto Bacchelli, Andrea Mocci and Michele Lanza. We have developed together an automated approach, based on Island-grammars, to identifying, extracting and analyzing source code fragments occuring in natural language documents (like emails, for instance). The analyzer is based on the ASF+SDF Meta-Environment (CWI, Amsterdam), that I have used intensively during my PhD thesis. As an additional outcome of this visit, a joint paper describing the approach has been prepared and submitted for publication to an international conference.

October 11-15, 2010 Lero research center, Limerick, Ireland : research visit in the team of Dr. Goetz Botterweck.

During this exciting visit, I've had short meetings with several members of the Lero center in order to identify opportunities for research collaboration with the ADAM team at INRIA Lille. I've also started to work with the S2T2 feature configurator, and in particular with its fmprimitives metamodel for representing feature models and configurations. I'm now using this technology intensively in the context of reverse-engineering architectural variability. We intend to work together in the near future on the topic of feature model evolution, based on the EvoFM approach developed at the Lero center.