ERCIM "Alain Bensoussan" Fellowship Scientific Report

Fellow: Ellen Van Paesschen
Visited Location: INRIA project JACQUARD/ADAM in Lille, France
Duration of Visit: 8 months (01/01/07-31/08/07)

I - Scientific activity

Context-Aware Computing and Ambient Intelligence With the proliferation of mobile devices and the mobile networks they create, the notion of "context" becomes increasingly important. More specifically, mobile devices sense the environment they are in and react accordingly. This is usually enabled by a set of *coordination* rules that infer how a device is supposed to react given a certain situation. In the Ambient domain, Context-Aware Computing (CAC) is commonly referred to as the new paradigm that employs this idea of context in order to enmesh computing in our daily lives. Many efforts exist today that focus on human-computer interaction based on context. On the other hand, computational models, middleware and/or programming languages are being developed that take the inherent characteristics of ad-hoc networks into account, such as connection volatility, ambient resources, etc.. An important challenge is to bridge the gap between the domain level and the computational level. The former is concerned with the expected behaviour of the system from a users viewpoint, such as how and when a system responds to changes in the context, when information can be made public, etc. On the other hand, the computational level deals with the inherent and very stringent hardware phenomena of ad-hoc networks. Nevertheless, both levels have to coexist: the computational level needs to be steered by the concepts, behaviour and rules which exist at the domain level, whereas the domain needs to adapt to the particulars of the ever changing network that is monitored and managed by the computational level.

In order to address this challenge, it is required to investigate representations at the domain level of concepts of Ambient Computing, such as the context. Furthermore, a mapping has to be devised between these concepts and generic concepts at the computational level, the latter being as independent as possible from concrete platforms or languages. This mapping has to be bidirectional: the computational level needs to be steered by the concepts, behaviour and rules that exist at the domain level, whereas the domain needs to adapt to the particulars of the everchanging mobile network that is monitored and managed at the computational level. Furthermore, the mapping has to be dynamic since the changes have to be propagated between the levels at run-time. The concept model of Ambient Computing is not only useful for bridging the aforementioned gap, but also for designing and developing open, task-specific languages at the domain level, which allow users to dynamically adapt the behaviour of the applications running on their mobile devices in well-defined ways. For illustrating the complexity of this matter I developed a number of futuristic scenarios dealing with *advanced user customisation* and *coordination in smart environments*.

Context Modelling Modelling has always been an essential part of software development. Models represent the domain of the software or make abstractions from the implementation. A recent trend in software development is Model-Driven Engineering, where models play a pivotal role in that programs or other models are generated from them. As such, it seems that concepts, behaviour and rules of Ambient Computing, and Context-Aware Computing in particular, have to be expressed at the modelling level. Evidently, the models have to be active and intimately connected to the actual implementation at the computational level, rather than serving passive documentation purposes. For this subject I studied the state-of-the-art in *context models* and their suitability for Ambient Computing. One of the results was that current context models lack standard and reusability. Defining how the application can adapt to the context is still a question that has no definite answer. Another result was the development of a new kind of context model in collaboration with other members of the ADAM team, which is a context-aware extension of the Unified Modelling Language (UML). Together with a set of high-level rules, this extended context model is employed for transforming non-context-aware applications into a context-aware ones (see publications 1 and 6). In order to cover the operational level I made a study of different kinds of *middleware*. The results on the state-of-the-art are that a specific group of middleware, more precisely reflective and context-aware middleware, shows the most adaptivity compared to other kinds such as tuple-space and event-driven middleware.

Other Research On a side track I also have been active in the domain of *Change-Oriented Model-Driven Software Engineering* (see publications 2 and 3) and on the subject of *Separation of Concerns for Supporting UI Evolution and Adaptation* (see publications 4 and 5). Also in this last context, I studied the currently available *UIs suitable for adaptation* to changing contexts and for Ambient Intelligence. Examples are adaptive UIs, plastic UIs, tangible UIs, multi-modal UIs, etc. The conclusion I derived was that there exists a myriad of different kinds of suitable UIs, all with their proper terminology and (often overlapping) functionality. Related to adaptive UIs, I researched the process of *user customization*. I concluded that the most advanced form of this activity named *end-user programming* performed by *programming by example/specification*, is the most interesting technique for bridging the gap between the domain and operational levels in Ambient Computing.

II - Publications during the Fellowship

1. **Introducing Context-Awareness in Applications by Transforming High-Level Rules**. Carlos Andrés Parra, Maja D'Hondt, Carlos Noguera and Ellen Van Paesschen. In the Workshop on Object Technology for Ambient Intelligence and Pervasive Systems (OT4AMI) at ECOOP 2007.

   In the last years, we have witnessed the increase in the popularity and capabilities of mobile technologies. This evolution has enforced the idea of smart environments, in which devices are aware and able to react to changes in their environment. In this position paper we describe a specific approach for the development of context-aware software. We propose to make existing applications context-aware by means of three main components: context models, high-level rules and code-generation processors. We present each component and analyze the issues related to the development of context-aware software following this strategy.

2. **DEUCE : A Declarative Framework for Extricating User Interface Concerns**. Sofie

Goderis, Dirk Deridder, Ellen Van Paesschen and Theo D'Hondt. In the Journal of Object Technology, Vol. 6, N 9; TOOLS 2007, ETH Zurich.

Evolving a software system not only affects the source code responsible for the core application, but also the user interface. Unfortunately user interface code is often scattered through and entangled with the application code. In large and complex user-interfaces, this tangling renders the implementation complex and hard to main- tain. Currently, the application needs to perform both the necessary changes to the user-interface (e.g. disabling other buttons, propagating events, etc.) as well as invoke the required application logic. The Deuce framework (Declarative User Interface Con- cerns Extrication) intends to reduce the complexity of user-interface implementations by applying separation of concerns on three UI concerns: presentation logic, business and data logic, and connection logic. It does so by using a declarative meta-language (SOUL) on top of an object oriented language (Smalltalk) such that an adequate lan- guage is provided to describe the entire structure and behaviour of the user-interface, as well as to link it with the application.

3. **DEUCE : Separating Concerns in User Interfaces**. Sofie Goderis, Dirk Deridder and Ellen Van Paesschen. In the Proceedings of the Second International Conference on Software Engineering Advances (ICSEA 2007), Cap Esterel, France.

   As current software systems evolve continuously, both the application and its user interface (UI) have to be adapted. However, UI code is often scattered through and entangled with the application code. In large and complex UIs, this tangling renders the implementation complex and hard to maintain. The Deuce framework (Declarative User Interface Concerns Extrication) intends to reduce the complexity of UI implementations by applying separation of concerns on three UI concerns: presentation logic, business and data logic, and connection logic. It does so by using a declarative meta-language (SOUL) on top of an object oriented language (Smalltalk) such that an adequate language is provided to describe the entire structure and behavior of the UI, as well as linking it with the application.

4. **Change-Oriented Round-Trip Engineering: First-class changes for Agile Software Development**. Peter Ebraert, Ellen Van Paesschen and Theo D'Hondt. In the Proceedings of the Atelier sur l'évolution et la rétro-ingénierie du logiciel, workshop at the Langages et Modèles à Objets (LMO 2007) conference, Toulouse, France.

   In order to support runtime changes, automated testing and refactorings in Agile Software Development (AgSD), we propose to represent changes applied on a system under development, as first-class objects. We envision the integration of a Change Management System in an existing methodology for AgSD called Advanced Round-Trip Engineering (ARTE). We explain how Change-Oriented ARTE (COARTE) allows capturing, visualising, replaying and rewinding changes that have been applied on the modelling, implementation and runtime views of an ARTE environment, and automatically synchronizes them with the other views. In this setup tests and refactorings can be composed out of changes, while runtime change propagation is realized with different propagation strategies.

5. **Change-Oriented Software Engineering**. Peter Ebraert, Jorge Vallejos, Pascal Costanza, Ellen Van Paesschen and Theo D'Hondt. In the proceedings of the 15th International Smalltalk Joint Conference (ESUG 2007), Lugano, Switzerland.

   We propose a first-class change model for Change-Oriented Software Engineering (COSE). Based on an evolution scenario, we identify a lack of support in current Interactive Development Environments (IDEs) to apply COSE. We introduce a set of five extensions to an exist-

ing model of first-class changes and describe the desired behaviour of change-oriented IDEs to support COSE. With the help of an evolution scenario, we show why those extensions are required. Finally we describe ChEOPS: a prototypical implementation of a change-oriented IDE on top of VisualWorks and illustrate how it supports the extended first-class change model. ChEOPS is finally used to validate COSE as a solution for the shortcomings of existing IDEs.

6. *In preparation*: **Generation of Application Composition Support in Ubiquitous Computing Based on Context Models**. Carlos Noguera, Johan Fabry and Ellen Van Paesschen. Submitted at SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.

   This work will handle on an extension to the aforementioned approach of publication 1, in order to be able to deal with distributed contexts and sudden (dis)connections, we propose to use the concept of `futures`.

7. *In preparation*: **Workshop Report on the ERCIM Symposium on Software Evolution, 5 October 2007, Paris, France**. Tom Mens, Kim Mens, Maja D'Hondt and Ellen Van Paesschen.

   This report will handle on the results of the above symposium.

III - Attended Seminars, Workshops and Conferences

I attended the following events:

- ECOOP 2007, July 30 - August 3, 2007, Berlin, Germany
- CALA (INRIA équipe associée Jacquard - PROG - SSEL) Meeting - Ambient Days, February 13-14, 2007, Vrije Universiteit Brussel, Brussels, Belgium
- Will attend: ERCIM Symposium on Software Evolution, 5 October 2007, Paris, France (Co-organizer)

I was a member of the following committees[1]:

- Organizing Committee of International ERCIM workshop on Software Evolution, co-located with the International Conference on Software Maintenance (ICSM), in October 2007 in Paris (France),
- Programme Committee of the Atelier sur l'évolution et la rétro-ingénierie du logiciel, workshop at the Langages et Modèles à Objets (LMO) conference, in March 2007 in Toulouse (France)

I have been a reviewer for the following journals:

- Software and System Modeling (SoSyM) journal (Springer-Verlag), 2007
- Transactions on Knowledge and Data Engineering journal (IEEE Computer Society), 2007

[1] I have been co-reviewer for several conferences such as ESUG 2007, JFDLPA07, MoDELS07.