Project-Team PASS

Processes for Adaptive Systems

Brest

Activity Report

2012

1 Team

Team leader
Antoine Beugnard, Professor, Telecom Bretagne

Administrative assistant
Armelle Lannuzel, Telecom Bretagne

Telecom Bretagne staff
Fabien Dagnat, Associate Professor
Julien Mallet, Associate Professor
Siegfried Rouvrais, Associate Professor
Maria-Teresa Segarra, Associate Professor

PhD students
Iyass Alloush, since October 2011
Vanea Chiprianov, October 2008 - January 2012
Thang Pham Quyet, October 2008 - December 2012
Christophe Guychard, since October 2008
Jean-Baptiste Lézoray, since October 2008
Fahad Golra, SFERE grant, since December 2008
Ali Hassan, October 2009 - September 2012
Anthony Lee, Cifre Orange Labs, since January 2011
Sébatien Martinez, Région Bretagne grant, since September 2012

Associate members
Jean-Marie Gilliot, Associate Professor, Telecom Bretagne, Associate Member
Claire Lassudrie, Associate Professor, Telecom Bretagne, Associate Member

2 Overall Objectives

2.1 Overview

Adaptive software aims at adjusting various artifacts or attributes in response to changes in the system operating environment. By operating environment we mean everything that may affect the software system behavior and its properties, for example changing requirements, execution platform, and resources. Adjustment can be performed at design time and leads to the disruption of software operation. However, the higher demand for ubiquitous, pervasive, embedded, and mobile environments, has led to the need of runtime adaptation, i.e., without software disruption. Runtime adaptation increases software complexity as the software itself has to cope with variability. Complexity is even increased if the software system is distributed.

Indeed, adjusting operations may involve artifacts running on different execution platforms and adaptation mechanisms available on them may vary.

On the other hand, software complexity is traditionally tackled by focusing on the software product itself or its related design process. The first one mainly proposes artifacts such as languages, software architectures, frameworks, and middleware that help on mastering complexity at a manageable degree for designers and maintainers. The second one is interested in providing a methodology (process) to design a software system that satisfies user requirements. It guides developers in taking into account important functional and non functional concerns and may be reused for each development. Both areas offer different but interesting approaches that help in managing software complexity.

The PASS team aims at combining both approaches to master adaptive software complexity. More specifically, it aims at defining system and process models and their relationships in order to ease the development and the adaptability of software systems. The mottos of the PASS research are:

1. *Adaptation requires to reify the software design process and trace its enactment.*

   In order to build adaptive software, it is necessary to reify its design process, rationales behind it, and to consider adaptivity as a first class property. Simply knowing the current variant of the software and the targeted one, as in traditional reflection-based approaches or more recently in DSPLs (Dynamic Software Product Lines), is not enough. Indeed, they do not consider relevant information for adaptation such as when it is possible to change, what can be changed, and how building blocs state should be managed.

2. *Processes are software.*

   The description, specification, and design of processes should follow the same process as software [Ost87]. As a consequence, current standard process models such as BPEL or BPMN are inappropriate since, they poorly distinguish specification from design and enactment.

3. *Components have distributed access points.*

   Classical software component models are bound to their physical execution targets (devices) and are connected through connectors (bindings) that ensure the assembly. We propose a paradigm shift releasing the software component model dependency from its underlying infrastructure. As a consequence, the access points of a single component can be distributed among various devices and (remote) communications are mananged inside components.

Our research is intended to be *vertical* in the sense that all aspects of systems are of interest: specification, design and implementation.

Solutions to enact these mottos are at the intersection of model driven engineering, software architecture, process modeling, verification and validation, and distributed systems. We

[Ost87] L. Osterweil, "Software processes are software too", *in: Proceedings of the 9th international conference on Software Engineering*, IEEE Computer Society Press, p. 2–13, 1987.

intend to scientifically contribute to the three first domains (model driven engineering, software architecture, process modeling).

In this scientific context, we are interested in three kinds of systems: large scale, highly dynamic (mobile, P2P) and strongly constrained (real-time, embedded) where non functional properties such as safety, performance and reliability are important. For instance, ambient assisted living, satellites, cloud, mobile games, pervasive learning, etc.

On the application side, the PASS team explores the application to software systems such as satellite, home-automation, multi-users games, etc but also to non-software systems such as educational systems.

2.2 Key Issues

In its current state, PASS studies the following key issues:

- Adaptation decisions require information to trigger adaptations.

  What is the information to be traced? How the process can keep track of this information?

- Non-functional properties (e.g., security, performance, dependability, scalability) and other quality attributes can be considered as drivers for software and enterprise architecture design.

  How methods, tools, models and processes could support non-functional specifications for analysis?

- Many software component models have been developed over the years. They all rely on an explicit binding with connector.

  Can't we provide an efficient component model that hides connectors?

- Software deployment relies on packet distribution with coarse-grain updating operators.

  Are there other levels of deployment? Are there other units of adaptation? What are the proper control primitives for updating software?

3 Scientific Foundations

3.1 Modeling and Meta-Modeling

This would be more detailled later.

3.2 Type Systems

This would be more detailled later.

3.3 Process Modeling

This would be more detailled later.

3.4 Software Components

Keywords: Abstraction, reusability, composability, component.

Glossary :

Abstraction a mechanism and practice to reduce and factor out details so that one can focus on few concepts at a time.

Composition a system design principle that deals with the inter-relationships of components. A highly composable system provides recombinant components that can be selected and assembled in various combinations to satisfy specific user requirements.

Component a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.

We have developed an original *component* model with high level of *abstraction* and a simple *composition* mechanism.

3.5 Adaptation and reconfiguration

In this research area we are interested in providing developpers with methods and tools to easily build (distributed) adaptive software. To acheive this goal, we are currently working on the definition of 1) a generic, customizable model for distributed dynamic adaptation mechanisms at the middleware level, and 2) a set of tools that may be used by adaptation designers in order to ease the customization of our model for their needs.

Models for distributed dynamic adaptation mechanisms. In a distributed component-based software, a set of components located in different nodes, cooperate in order to provide services. Making such a system adaptive requires including adaptation mechanisms that are themselves distributed. Several works exist that propose distributed algorithms for either deciding or planning changes [FDP+12]. Although those approaches contribute to simplify the task of building an adaptive software, they fail to provide an architectural view of the adaptation mechanisms. Such a view would allow designers to choose different distribution strategies according to their needs, i.e., some components may be managed in a distributed manner while others may not.

To tackle this limitation we have been working on a model for distributed and coordinated dynamic adaptation [14]. We have proposed a model that constrains designers considering what are the dynamic adaptation services they need for their application and how many instances (and location) of the selected services should be created. According to the number of instances of the different services, coordination components may be mandatory and different coordination strategies are available to be used by designers, whether for decision, planning or execution services. In order to validate this work, we considered the distributed data management application domain and we intend to dynamically change data replication policies depending on the execution environment.

Development process for adaptive distributed software. The second research area is related to the automatic generation of dynamic adaptation mechanisms: decision making,

[FDP+12] F. Fouquet, E. Daubert, N. Plouzeau, O. Barais, J. Bourcier, J.-M. Jézéquel, "Dissemination of reconfiguration policies on mesh networks", *in: DAIS 2012*, Stockholm, Sweden, June 2012, `http://hal.inria.fr/hal-00688707`.

planning, and execution. This work is based on the main result of Eric Cariou and Chantal Kaboré PhD work, the Cloud Component approach [7][9], a communication abstraction (Cloud Component) used to model distributed services and a refinement process that allows for automatic generation of architectural variants. It is included in a current research trend that proposes DSPLs (Dynamic Software Product Lines) and MDE (Model-Driven Engineering) to automatically compute the actions to be performed to build the target variant when an adapdation is needed, i.e., to compute the "adaptation plan" or "reconfiguration plan". In current approaches, these actions correspond to operations on the application architecture (add component, remove component, etc...). These works propose general enough approaches that can be applied to all model-based applications, but they fail to provide a systemic approach for automatic generation of adaptation plans.

Compared to these approaches, in this research area we are interested in investigating how to generate not only architectural modification actions but also (all the) actions of an adaptation plan including application-dependent actions. We claim that during the development process relevant information may be collected then exploited to compute adaptation plans.

We have been working on one type of application-dependent actions: those related to the data managed by the components affected by an adaptation. Indeed, when a component is modified or replaced its data should also taken into account. These data are ignored by most of current approaches that assume stateless components. To tackle this issue, we have been working on two areas :

- The first one aims at helping a developper to explicitly add information related to the data managed by the application [10]. This information is related to the entities that manage data and the operations used to managed them. Such information is then used to compute data-related actions to be added to adaptation plans. More specifically, data transfers between components can be computed and added to the adaptation plan.

- The second one is related to the concept of variation point in DSPLs and its reification at runtime. Indeed, current approaches assume that all variation points in a DSPL of an application should be available at runtime: each time an optional feature is present in the DSPL, the application should be able to change at runtime to include or remove the feature. In this work, we claim that each variation point identified in the DSPL may be reified at runtime and we investigate a development process that allows to reify such variation points [11].

Glossary :

Adaptive software a software that has the ability to adapt at runtime to handle such things as changing user needs, system intrusions or faults, changing operational environment, and resource variability.

Dynamic or runtime adaptation a change on a software that takes place after deploying it. It includes deciding about the change to perform, planing the necessary operations on the software, and executing them.

4 Application Domains

4.1 Component Software Adaptation

Participants: Antoine Beugnard, Maria-Teresa Segarra Segarra, Julien Mallet, Fabien Dagnat, Anthony Lee, Sébastien Martinez.

Apply the mottos 1, 2 and 3.
Domains : Home-automation, HPC, Large scale applications.

4.2 Process Design

Participants: Antoine Beugnard, Fabien Dagnat, Fahad Golra.

Apply the mottos 1, 2.

4.3 Educational Systems

Participants: Siegfried Rouvrais, Antoine Beugnard.

An educational system provides services to students and society. For strategic alignment, to adjust in response to requirements or resources changes in the societal environment, the management of educational transformation is of decisive importance to Higher Education Institutions (HEI) and regulation bodies. During the last decade, various models of quality management have emerged in the form of quality assurance standards (e.g. accreditation bodies, European or international frameworks) that guide programme leaders, designers or deans of academics to evaluate and improve educational systems including syllabus, curricula, workforce, workspaces, support services, or administrative processes. However, to date, educational system complexity is traditionally tackled by focusing restrictively on the course or curricula contents. There is no standard commonly accepted for conceptually describing the overall complexity of educational systems, restricting the modernization agenda of HEI nationally, in Europe, and internationally.

As the functional, regulatory and quality requirements (e.g. non functional properties) of educational systems increase, it is a major challenge today to unambiguously describe and organize concepts among the various stakeholders involved in programme design or transformation, so as service consumption. The actual complexity of educational systems and the dynamic of their requirements call today for sound engineering principles. The aim is this research is to define and provide methods, processes and tools so as to design and verify educational systems that satisfy requirements under controlled resources. Based on modeling, meta-modeling and business process modeling foundations, well defined models and processes for continuous improvement could stimulate a programme reform, transformation or renewal so as to better align with requirements (e.g. learning outcomes [16]). A system modeling approach is an obvious choice for meeting such challenges [7] since it makes it possible to meaningfully, unambiguously, and accurately specify concepts, relations, and viewpoints among stakeholders, so as to flexibly manage change. Educational architectures and frameworks [7] can thus be defined, with varying degrees of rigor, so as to prepare and organize various views and models for stakeholders, manage processes [15] for the various engineering phases, and systematize some analysis for better quality fulfillment.

From this perspective, applying the mottos 1, 2 and based on a convergence of system engineering, enterprise architecture, and education engineering principles, this PASS research track focuses on architecting educational systems [7] and expects to provide in-depth and usable models, methods, and tools to address issues of educational system design, transformation [15], and adaptability.

5 Software

5.1 ReCaml

Participants: Fabien Dagnat [contact point].

To fix bugs or to enhance a software system without service disruption, one has to update it dynamically during execution. Most prior dynamic software updating techniques require that the code to be changed is not running at the time of the update. However, this restriction precludes any change to the outermost loops of servers, OS scheduling loops and recursive functions. Permitting a dynamic update to more generally manipulate the program's execution state, including the runtime stack, alleviates this restriction but increases the likelihood of type errors. ReCaml [6] is a language for writing dynamic updates to running programs that views execution state as a delimited continuation. ReCaml includes a novel feature for introspecting continuations called match cont which is sufficiently powerful to implement a variety of updating policies. We have formalized the core of ReCaml and proved it sound (using the Coq proof assistant), thus ensuring that state-manipulating updates preserve type-safe execution of the updated program. We have implemented ReCaml as an extension to the Caml bytecode interpreter and used it for several examples.

5.2 MetaModMap

Participants: Thang Pham Quyet [previous PhD], Antoine Beugnard [contact point].

MetaModMap is an eclipse-based plugin which allows to define the intensional semantics correspondences of elements between two similar metamodels. From this definition, the plugin provides a facility which rewrites a legacy transformation definition for the old metamodels to a new one for the new similar metamodels.

5.3 Openflexo

Participants: Antoine Beugnard, Fabien Dagnat, Christophe Guychard [contact point], Sylvain Guérin [SCIC openflexo].

Openflexo is an open-source (GPL3) business architecture platform that supports collaborative agile missions by seamlessly transforming multifaceted enterprise models into business oriented deliverables. Openflexo Modeller provides frequent delivery production cycles (storyboard, documentation, operational prototypes and applications). Openflexo is business oriented. It's a tool gathering all actors around a shared visualization of a multifaceted model, continuously updated and enriched at each iteration.

6 New Results

6.1 Telecommunications Services. An Enterprise Architecture and Model Driven Engineering Method

Participants: Vanea Chiprianov[1].

The focus on the end consumer, the convergence with the Internet, the separation between the software and the hardware implementing a service, and the telecommunications market deregulation have led to a revolution and a new era in the telecommunications industry. Former national telecommunications providers have to reduce the construction time, from months to days, while affecting non-negatively other parameters (e.g., cost, quality of service, quality of experience) of new telecommunications services. To tackle this broad theme, we have proposed (i) a telecommunications service construction process, (ii) software tools to be used in this process and (iii) a tool building process to build these tools. The telecommunications service construction process reflects current practices in the telecommunications industry. The software tools (i.e., Domain Specific Modeling Languages designed as profiles of an Enterprise Architecture Modeling Language, graphical editors, code generators, Off the Shelf network simulators, a collaboration Design Rationale Domain Specific Modeling Language) contribute towards answering the challenges faced by telecommunications providers. The tool building process relies on models and provides a high automation degree, hence it allows building software tools more rapidly.

The results contribute to reducing the construction time of new telecommunications services, their early verification regarding non-functional properties such as performance, while providing the possibility of improved quality of service and increased involvement of the consumer. Faster provisioning of new telecommunications services, that answer better to consumers' needs, increase the rate of development of new economic services in general.

6.2 Adding spatial information to software component model – The localization effect

Participants: Ali Hasan[2].

Highly distributed environments (HDEs) are deployment environments that include powerful and robust machines in addition to resource-constrained and mobile devices such as laptops, personal digital assistants (or PDAs), smart- phones, GPS devices, sensors, etc. Developing software for HDEs is fundamen- tally different from the software development for central systems and stable distributed systems. This argument is discussed deeply and in-details throughout this disserta- tion. HDE applications are challenged by two problems: unreliable networks, and heterogeneity of hardware and software. Both challenges need careful han- dling, where the system must continue functioning and delivering the expected QoS. This dissertation is a direct response to the mentioned challenges of HDEs. The contribution of this dissertation is the cloud component model and its related formal language and tools. This is the general title. However, and to make this contribution clear, we prefer to present it in the following detailed form:

1. We propose a paradigm shift from distribution transparency to localization acknowledgment being the first class concern.
2. To achieve the above mentioned objective, we propose a novel component model called cloud component (CC).
3. In this dissertation we propose a new approach to assemble CCs using systematic methodology that maintains the properties of CC model.
4. Cloud component development process and cloud component based systems development process.
5. Location modeling and advanced localization for HDEs are the pivotal key in our contribution.
6. Formal language to model single CC, CC assembly, CC development process, and CC based systems.
7. We finally present our fully-developed supporting tools: the cloud component management system CCMS, and the Registry utility.

6.3 Model Transformation Reuse: A Graph-based Model Typing Approach

Participants: PHAM Quyet Thang[3].

Identical domain concepts reified in different (meta)modelling projects may be named, represented and connected differently. It turns out that a transformation de- fined for a particular metamodel cannot be directly used for another metamodel; that is, the reuse of transformations is restricted. To tackle this problem, in this dissertation, we propose a solution for automatically migrating legacy transforma- tions. Such a transformation is adapted to the new metamodel that has a slightly different representation in comparison with the original one, while preserving the ori- ginal semantics of the transformation. To this end, we first introduce MetaModMap, a Domain Specific Language that allows the description of the correspondences of intended semantics between the elements of two metamodels that model the same domain. Then we provide a rewriting mechanism using these user-defined correspon- dences to migrate the transformation automatically. The proposed solution uses a graph-based model typing relation that enables safe adaptations. Our approach has been prototyped with MOMENT2 and can be used with any framework based on the same graph transformation paradigm.

6.4 Other results

Publications: [6, 8, 10, 5, 12, 11, 13, 9, 16, 4, 14, 2, 3, 1]

6.5 Assessment of achievements

Participants: All participants.

The results achieved by the PASS team must be compared with the key issues presented in the objective part. Not all key issues have deserved attention yet. However, a few of them have been sufficiently well explored to start and draw conclusions.

The first key issue is “What are the information to be trace? How the process can keep trace of these information?” The information to trace are mainly the information related to the decision choices that have an impact on the component parts that are to be adapted or modified. We experiment different models where to attach those informations. Those models are close to process models or feature models; all are some kind of process abstractions.

The second key issue is “Can we provide an efficient component model that hides connectors?” We have proposed a component model that illustrates a paradigm shift. We can develop distributed application with components that encapsulate all the communications required by the distribution, but that are composed only locally. This component model should be the beginning of new experiences and developments.

7 Contracts and Grants with Industry

7.1 Projet Recherche Innovante Thalès

8 Other Grants and Activities

8.1 International Collaborations

- Vincent Englebert, from Université Catholique de Namur.

- Siegfried Rouvrais is the French representative at CDIO collaborator’s international meetings (e.g. Stanford 2011, QUT-Brisbane 2012, Aarhus 2013, MIT/Harvard 2013). The CDIO INITIATIVE (acronym for Conceive – Design – Implement – Operate) is a structured educational framework for preparing the next generation of engineers. The framework provides students with an education stressing engineering fundamentals set in the context of Conceiving & Designing & Implementing & Operating real-world systems and products [7]. Throughout the world, now mor than 100 recognized institutions have adopted CDIO as the framework of their curricular and programme design, planning, supported by defined verification and validation processes.

8.2 National Collaborations

- The PASS team has a contract with Région Bretagne for the funding of S. Martinez’s PhD (since september 2012). This thesis is co-advised by Jérémy Buisson of ArchWare Team of the University of Vannes / IRISA.

9 Dissemination

9.1 Involvement in the Scientific Community

- Antoine Beugnard has been invited as a reviewer for the JUCS Special Issue / SBCARS (Brazilian Symposium on Software Components, Architectures and Reuse) 2012. He has been member of the programming committee of WICSA/ECSA 2012, DAIS 2012 and CIEL 2012. He was also member in several doctoral committees: Jonathan Labejof on "R* Réflexivité au service de l'évolution des systèmes de systèmes", Zied Abid on "Gestion de la qualité du contexte pour l'intelligence ambiante".

- Siegfried Rouvrais research interests are in the fields of software & system modeling and engineering education research & development. He has been a member of several international conferences in the both domains. He organized the CDIO international Fall meeting 2012 at Telecom Bretagne. In France, he has been invited as a speaker at a Meeting *de la Conférence des Grandes Écoles*, Paris, France, 2012 for a talk entitled: *Le cadre CDIO : un rérentiel d'objectifs d'apprentissage orienté activités professionnelles (ingénieurs) et des standards pour accompagner le changement : retour d'expérience comme outil d'amélioration continue.*

- Maria-Teresa Segarra has been a member of the programming commitee of ICOST 2012 and DSAI 2012.

9.2 Teaching

- Antoine Beugnard teaches software engineering, modeling and object programming. He teaches these courses at License and at Master level. He is the coordinator of the brittany research master (Master Recherche en Informatique de l'Université de Rennes 1) at Télécom Bretagne.

- Fabien Dagnat is the head of the computer science speciality at Telecom Bretagne. He teaches advanced programming (programming languages, functional programming, compilation), formal modeling (for concurrency mainly) and object oriented programming to master students of Telecom Bretagne. He teaches also a formal modeling course in the core of the brittany research master.

- Julien Mallet teaches software engineering, modeling, object oriented programming and computer security at License and at Master level. He is also involved in project-based learning.

- Siegfried Rouvrais is involved in a broad range of courses relating to computer science and software engineering both at B.Sc. and M.Sc. levels, as well as for adult professionals (e.g., Requirements, Software Architecture, Programming Languages, UML, Business Process Design and Modelling, Workflow, Enterprise Information Systems). In connection with these theoretical and practical courses, he investigates problem- and project-based learning styles aiming at favoring students' autonomy with a focus on process and reflectivity. Since 2003, as educational program designer, he has been particularly involved in integrating such models into Telecom Bretagne curricula and vocational trainings, with a clear emphasis on student's competency development, multidisciplinarity, and industry connections. Since several years, he trains faculty members and offer consultations for improving teaching and curriculum design (PjBL, Active Pedagogy, Reflectivity, Constructive Alignment, Programme Design, etc.).

- Maria-Teresa Segarra is the head of the computer science domain at Telecom Bretagne. She teaches advanced programming paradigms (component and service oriented), object-oriented programming and design and databases at License and Master levels.

10 Bibliography

Major publications by the team in recent years

[1] M. Belguidoum, *Conception d'une infrastructure pour un déploiement sûr et flexible des composants logiciels*, PhD Thesis, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), LIT - Laboratoire d'informatique des télécommunications (Institut Télécom-Télécom Bretagne), 2008.

[2] L. Besnard, T. Gautier, J. Ouy, J.-P. Talpin, J.-P. Bodeveix, A. Cortier, M. Pantel, M. Strecker, G. Garcia, A.-E. Rugina, J. Buisson, F. Dagnat, "Polychronous interpretation of synoptic, a domain specific modeling language for embedded flight-software.", *Formal methods for aerospace*, 18, 2010, p. 80 – 87.

[3] A. Beugnard, J.-M. Jezéquel, N. Plouzeau, D. Watkins, "Making components contract aware", *Computer 32*, 7, july 1999, p. 38 – 45.

[4] A. Beugnard, S. Sadou, "Method overloading and overriding cause distribution transparency and encapsulation flaws", *Journal of Object Technology, Special Issue OOPS Track at SAC 2006 6*, 2, february 2007, p. 31 – 45.

[5] A. Beugnard, "OO languages late-binding signature", *in: The Ninth International Workshop on Foundations of Object-Oriented languages, FOOL 9*, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), LIT - Laboratoire d'informatique des télécommunications (Institut Télécom-Télécom Bretagne), p. 1 – 6, 2002.

[6] J. Buisson, F. Dagnat, "ReCaml: execution state as the cornerstone of reconfigurations", *in: ACM SIGPLAN: 15th International Conference on Functional Programming*, ACM (editor), p. 27 – 38, New York, NY, USA, 2010.

[7] E. Cariou, *Architecture des applications distribuées utilisant des médiums de communication*, PhD Thesis, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), LIT - Laboratoire d'informatique des télécommunications (Institut Télécom-Télécom Bretagne), UR1 - Université de Rennes 1, 2003.

[8] J.-M. Gilliot, A. P. Khac, A. Beugnard, M. T. Segarra, "L'ingénierie dirigée par les modèles pour la conception d'applications à architectures réparties adaptables", *Technique et science informatiques 30*, 1, janvier 2011.

[9] E. Kabore, *Contribution à l'automatisation d'un processus de construction d'abstractions de communication par transformations successives de modèles*, PhD Thesis, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), LIT - Laboratoire d'informatique des télécommunications (Institut Télécom-Télécom Bretagne), 2008.

[10] A. P. Khac, *A model-driven feature-based approach to runtime adaptation of distributed software architectures*, PhD Thesis, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), UR1 - Université de Rennes 1, UEB - Université Européenne de Bretagne (UEB), 2010.

[11] J.-B. Lezoray, M. T. Segarra, A. Beugnard, J.-M. Gilliot, "Modeling Dynamic Adaptations using Augmented Feature Models", *in: 28th Symposium On Applied Computing*, 2013.

[12] J. Mallet, S. Rouvrais, "Style-Based Model Transformation for Early Extrafunctional Analysis of Distributed Systems", *in: QoSA '08 : international conference on quality of software architecture, october, Karlsruhe, Germany*, S. verlag (editor), INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), p. 55 – 70, 2008.

[13] S. Matougui, *Proposition d'un processus de réification d'abstraction de communication comme un connecteur associé à des générateurs*, PhD Thesis, INFO - Dépt. Informatique (Institut Télécom-Télécom Bretagne), LIT - Laboratoire d'informatique des télécommunications (Institut Télécom-Télécom Bretagne), 2005.

[14] M. Zouari, M. T. Segarra, F. André, A. Thépaut, "An Architectural Model for Building Distributed Adaptation Systems", *in: 5th International Symposium on Intelligent Distributed Systems*, 2011.

Doctoral dissertations and "Habilitation" theses

[1] V. Chiprianov, *Collaborative Construction of Telecommunications Services. An Enterprise Architecture and Model Driven Engineering Method*, PhD Thesis, LUSSI - Dépt. Logique des Usages, Sciences Sociales et de l'Information (Institut Mines-Télécom-Télécom Bretagne-UEB), january 2012.

[2] A. Hassan, *Adding Spatial Information to Software Component Model - The Localization Effect*, PhD Thesis, INFO - Dépt. Informatique (Institut Mines-Télécom-Télécom Bretagne-UEB), September 2012.

[3] Q. T. PHAM, *Model Transformation Reuse: A Graph-based Model Typing Approach*, PhD Thesis, INFO - Dépt. Informatique (Institut Mines-Télécom-Télécom Bretagne-UEB), December 2012.

Articles in referred journals and book chapters

[4] V. Chiprianov, Y. Kermarrec, S. Rouvrais, J. Simonin, "Extending Enterprise Architecture Modeling Languages for Domain Specificity and Collaboration: Application to Telecommunications Service Design", *Software and systems modeling Theme Section paper*, november 2012, p. 1 – 11.

[5] V. Chiprianov, Y. Kermarrec, S. Rouvrais, *Integrating DSLs into a Software Engineering Process: Application to Collaborative Construction of Telecom Services*, IGI Global, Hershey, Pennsylvania, USA, 2012, ch. Formal and Practical Aspects of Domain-Specific Languages: Recent Developments, p. 415 – 442.

[6] J.-B. Lezoray, P. Roy, B. Abdulrazak, A. Thépaut, M. T. Segarra, "Comparison and Complementarity of Two Approaches to Implement AAL Systems", *Lecture notes in computer science 7251*, juin 2012, p. 210 – 213.

[7] S. Rouvrais, V. Chiprianov, "Architecting the CDIO Educational Framework Pursuant to Constructive Alignment Principles", *International Journal of Quality Assurance in Engineering and Technology Education (IJQAETE) 2*, 2, april 2012, p. 80 – 92.

[8] J. Simonin, A. Beugnard, R. Nedelec, "Processus de développement de système fondé sur l'alignement de modèles", *Ingénierie des systèmes d'information 17*, 3, octobre 2012, p. 119 – 142.

Publications in Conferences and Workshops

[9] I. Alloush, V. Chiprianov, Y. Kermarrec, S. Rouvrais, "Linking Telecom Service High-level Abstract Models to Simulators based on Model Transformations: The IMS Case Study", *in: Information and Communication Technologies.18th EUNICE/ IFIP WG 6.2, 6.6 International Conference, EUNICE 2012, Proceedings*, S. B. Heidelberg (editor), *7479*, p. 100 – 111, Heidelberg Dordrecht London New York, 2012.

[10] V. Chiprianov, Y. Kermarrec, S. Rouvrais, "Extending Enterprise Architecture Modeling Languages: Application to Telecommunications Service Creation", *in: SAC 2012: 27th symposium on applied computing*, *2*, p. 810 – 816, 2012.

[11] F. R. Golra, F. Dagnat, "Generation of Dynamic Process Models for Multi-metamodel Applications", *in: ICSSP 2012: International Conference on Software and System Process*, 2012.

[12] F. R. Golra, F. Dagnat, "Specifying the Interaction Control Behavior of a Process Model using Hierarchical Petri Net", *in: PMDE 2012: 2nd Workshop on Process-based approaches for Model-Driven Engineering*, 2012.

[13] Q. T. Pham, A. Beugnard, "Automatic Adaptation of Transformations based on Type Graph with Multiplicity", *in: 38th Euromicro Conference on Software Engineering and Advanced Applications*, 2012.

[14] S. Rouvrais, N. Chelin, "Introducing personality tests to clarify engineering student self-perception and demystify recruitment procedures: quantitative and qualitative results", *in: 8th International CDIO Conference*, 2012.

[15] S. Rouvrais, G. Landrac, "Resistance to Change in Institutionalizing the CDIO Standards: From a Waterfall to and Agile Improvement Model", *in: 8th International CDIO Conference*, 2012.

[16] S. Rouvrais, "Recognizing non Formal Learning Experiences: Top-down or Bottom-up Approaches for Skills Alignment", *in: EDUCON 2012: 3rd IEEE Global Engineering Education Conference, Collaborative Learning & New Pedagogic Approaches in Engineering Education*, IEEE (editor), p. 1 – 5, 2012.