

DEVELOPING A PRODUCT DERIVATION PROCESS FRAMEWORK FOR SOFTWARE PRODUCT LINE ORGANISATIONS

Padraig O’Leary, Dr. Ita Richardson, Dr. Steffen Thiel

LERO, The Irish Software Engineering Research Centre, University of Limerick, Limerick,
Ireland, {padraig.oleary, ita.richardson, steffen.thiel}@lero.ie

Abstract

Inefficient product derivation practices can greatly diminish the productivity gains expected from a software product line approach. As a foundation for systematic and efficient product derivation a better understanding of the underlying activities in industrial product line development is required.

This research has been developing a process framework that comprises important tasks product line stakeholders have to perform during product derivation. The framework is based on both literature and industrial practice. In this report, the status of the current research is presented along with the observations obtained thus far and how these observations were obtained.

Keywords: Software Product Lines, Product Derivation, process framework.

1 RESEARCH BACKGROUND

1.1 Research Question

Product derivation is a complex activity, involving a unique mesh of traditional software engineering approaches. However, product derivation practices are still at an early stage of evolution, and has primarily been driven by both industry and tool approaches to product derivation. In the absence of a theoretical approach to product derivation this research is focusing on the following research question:

“What are the basic activities in the product derivation process and to what extent can the development of a product derivation process framework guide organisations towards a “best-practice” approach to PD?”

1.2 Research Aims and Objectives

The main aims and objectives of this research project are to:

1. Develop a framework for the product derivation process that provides enhanced support for the derivation of products from a Software Product Line.
2. Validate the framework

1.3 Motivation for the Research

Product Derivation is the process of constructing a product from a Software Product Lines (SPL) core assets (Deelstra et al., 2005). An effective product derivation process can help to ensure that the effort required to develop the platform assets is less than the benefits delivered through using these shared

artefacts across the products within a product line. In fact, the underlying assumption in SPL that "the investments required for building the reusable assets during domain engineering are outweighed by the benefits of rapid derivation of individual products" (Deelstra et al., 2004) might not hold if inefficient derivation practices diminishes the expected gains.

In the literature a number of publications speak of the difficulties associated with product derivation. Hotz et al. (Hotz et al., 2003) describe the process as "slow and error prone even if no new development is involved". Griss (Griss, 2000) identifies the inherent complexity and the coordination required in the derivation process by stating that "...as a product is defined by selecting a group of features, a carefully coordinated and complicated mixture of parts of different components are involved". Therefore as Deelstra (Deelstra et al., 2005) points out, the derivation of individual products from shared software assets is still a time-consuming and expensive activity in many organisations. Despite this, there has been little work dedicated to the overall product derivation process. Rabiser et al. (Rabiser et al., 2007) claim that "guidance and support are needed to increase efficiency and to deal with complexity of product derivation". Therefore as Deelstra (Deelstra et al., 2005) states there "is a lack of methodological support for application engineering and, consequently, organizations fail to exploit the full benefits of software product families."

2 LITERATURE REVIEW

2.1 Software Development Process Models

As SPL has its roots in general software engineering, research that specialises on SPL process models has to consider work on software engineering process models. As an example of these software engineering process models, the researcher is focusing on the on the Rational Unified Process (RUP) and agile methodologies.

Rational Unified Process. The Rational Unified Process (RUP) (Kruchten, 2004) provides a disciplined approach to the assignment of tasks and responsibilities within a process. RUP is supported by tools, which automate parts of the process. These tools are used to create and maintain the various artefacts of the process. RUP is a configurable process. It acknowledges that no single process is suitable for all software development projects. Hence, it provides a generalised process architecture that provides commonality across a family of processes and can be adapted for the particular case. RUP claims to have captured many of the best practices in modern software development in a form that is useful and suitable for a wide range of projects and organizations.

Agile Methodologies. Agile practices have recently gained popularity as a mechanism for reducing costs and increasing ability to handle change in dynamic market conditions. Researchers and practitioners have proposed several software development approaches based on the principles of the Agile manifesto . The Agile manifesto allows for changing requirements throughout the development cycle and stresses collaboration between developers and customers, and early product delivery. People (coupled with effectiveness and manoeuvrability) are considered the primary drivers of project success (Highsmith et al., 2001). eXtreme Programming (XP) and Scrum are two well-known agile practices based on the agile manifesto. XP emphasises technical aspects; while Scrum concentrates on project management.

2.2 Product Derivation Approaches and Tools

To date, several approaches and tools that support or partly automate product derivation activities in SPL have been proposed. Krebs et al. (Krebs et al., 2005) outline a derivation methodology that uses a configuration model to represent functionality and variability in a product line. The configuration model supports two types of artefacts: components and features. The components are derived from the physical architecture of the product line. Features represent a customer's view of the functionality in

the components. A mapping between features and components allows for automated inferring of the components required for a given selection of features.

Asikainen et al. (Asikainen et al., 2004) provide a product configuration modelling language (PCML) and configuration tool (WeCoTin). PCML supports the creation of feature models for a software product line. WeCoTin is used to derive valid feature models for particular products of the product line.

The ConIPF Methodology (Hotz et al., 2003) proposed by Hotz et al. tackles the challenges of product derivation by combining concepts from product line engineering and knowledge-based configuration.

Rabiser et al. (Rabiser et al., 2007) present an approach for supporting product derivation using feature specifications. The approach introduces business decision-making into product derivation through a combination of modelling stakeholder needs, product features, architectural elements, and variability. The approach emphasises supporting the requirements acquisition and management mechanism through the use of variability models.

McGregor (Chastek et al., 2002) introduces the *production plan*, which prescribes how products are produced from platform assets. It contains the attached processes of the platform assets as well as an overall scheme of how the processes are combined to build products. The product plan facilitates the passing of knowledge between the platform developers and the product developers. An example of the production plan in use is given in (Chastek et al., 2002). McGregor (McGregor, 2005) also provides an overview of technologies and approaches to automate product derivation.

Deelstra et al. (Deelstra et al., 2005) present a product derivation approach developed based on two industrial case studies. The framework consists of two phases: an initial and an iteration phase. During the initial phase, a first product configuration is derived from the product line artefacts. The initial configuration is modified in a number of subsequent iterations during the iteration phase until the product sufficiently implements the imposed requirements. Requirements that cannot be accommodated by existing assets are handled by product-specific adaptation or reactive evolution. Parts of the derivation framework have been implemented in a research tool called COVAMOF (Sinnema et al., 2006), a variability modelling framework which purports to solve the product derivation problems associated with dependencies.

The work by Deelstra et al. presents a framework of terminology and concepts for product derivation. The framework focuses on product configuration and is a high level attempt at providing the methodological support that Deelstra et al. (Deelstra et al., 2004) agree is required for product derivation.

3 THEORITICAL BASIS FOR WORK

This research aspires to develop a Product Derivation Process Framework (PDPF). The framework builds on work done by Deelstra et al. (Deelstra et al., 2005). It identifies tasks, roles, artefacts and process flows within product derivation. A framework, as proposed here, provides benefits to both academia and industry.

For academia, the framework provides structure to the area under concern. Hence, it becomes easier place a particular research topic in context. There is historical evidence of certain fields achieving progress at the expense of others, through the establishment of a core, theoretical structure (Latour, 1988). As a roadmap, the framework points to areas of uncertainty and helps identify remaining challenges. Such a roadmap encourages the insertion of those pieces that may be missing, or the extra detail that may be needed for a particular purpose or group.

For industry, it is envisaged that the development of the framework will help the advancement of product derivation practices. It will assist organisations by providing a structured approach to product derivation, making the process more predictable and manageable. Moreover, this enables the

integration of non-standard techniques such as agile practices, at appropriate times of the development process (O’Leary et al., 2007; O’Leary et al., 2007).

It can be argued that without some established process framework for product derivation, the applied practices will be motivated by technological solutions (e.g., available tools), rather than good practice. Standardisation of the development process around a methodology rather than a tool facilitates technology change. Tools can be interchanged more easily once seen to work within the bounds of the methodology. The framework can also support the development of these tool environments to make the derivation of products more efficient and effective.

4 RESEARCH DESIGN

This section presents the research approach used in this study.

4.1 Steps in achieving research objectives

The steps required to complete this research are as follows:

1. Create an initial PD Process framework (COMPLETE)
 - i) Perform an extensive literature review of PD practices within SPL.
 - ii) Extrapolate from the literature an initial process framework for PD.
 - iii) Through a series of organised iterative workshops with SPL experts and practitioners, the framework is refined and assessed.
2. Iterative Workshop Series (COMPLETE)
 - i) Series of workshops organised over four month period, with five participants attending twice a month
 - ii) The framework was iteratively developed based on comments and feedback from participants
3. Perform Case Study Research (COMPLETE)
 - i) Review Case Study Company PD practices based on company documentation
 - ii) Perform two day in-dept facilitated workshop study
 - iii) Create a technical report on documented Case Study Company practices
 - iv) Validate documented practices by emailing technical report to the Case Study Company.
 - v) Revise initial version of framework based on technical report.
4. Perform Expert Opinion Analysis (NOT-STARTED)
 - i) Design research instrument construction
 - ii) Perform pilot study of research instrument
 - iii) Establish selection of key SPL experts who are willing to participate in the analysis
 - iv) Refine and assess framework based on feedback
 - v) Iteratively develop framework, until it meets certain pre-defined stop criteria.

4.2 Objective 1 – Development of the Framework

The framework is being developed in response to the absence of a defined “best-practice” approach to PD within current literature (Section 1.2). This research intends to rectify this gap in knowledge. It is a requirement that the framework would take into account the particular domain of operation. It is envisaged that this would lead to a more efficient productive derivation process and one which reflects better the investment made by the SPL organisation in the development of platform assets.

The framework described in this report (Section 5) is an output of the first three steps. In the development of this version of the framework an extensive literature review of, existing SPL whitepapers, PD papers and software process improvement (SPI) practices, was conducted. Evidence and feedback from SPL practitioners and researchers was collected from organised workshops and case study research into industrial product derivation practices was conducted.

The output of this research will be a PD process framework that supports an organisation engaged in PD. The framework will provide a structured approach to the development of a situational PD process, i.e. a process that takes into account the particular domain of operation, for instance systems development will have additional cross discipline coordination tasks. It is envisaged that the framework will improve the effectiveness and efficiency of the PD process within SPL organisations by forcing the adoption of the most essential PD practices for a particular domain

4.3 Objective 2 – Validation of the Framework

The researcher is looking at the practices and processes involved in the derivation of products from a SPL rather than any end product of the derivation process. Accordingly, qualitative methods are appropriate when the process is being studied rather than the product (Patton, 1987).

The study itself can be classified as applied research and more specifically, as constructive research. Järvinen (Järvinen, 2001) defines constructive research as typically involving the building of a new innovation based on existing (research) knowledge and new technical or organisational advancements. Furthermore, Järvinen suggests that constructive research also involves an evaluation of the innovation. According to Järvinen, in constructive research it is possible to accept a prototype or even a plan as a research outcome instead of a final product.

According to Hakim (Hakim, 1987) small samples of experts can be used to develop and test explanations. Previous studies have used small samples to gain expert feedback to evaluate and support model development. There are many successful examples where expert opinion has been used to validate models. For example, Lauesen and Vinter used expert opinion to identify techniques for requirement defects prevention (Lauesen et al., 2001). This research plans to emulate such studies in its research design.

5 RESEARCH METHODS

5.1 Literature Review

The preparatory stage of this research was conducted as a review of SPL literature focusing in particular on PD. As suggested by Cooper (Cooper, 1984), the literature review is aimed at defining the research topic’s current state of knowledge. More specifically the research aimed to identify the fundamental practices of PD, to study the existing PD practices as well as to chart the available empirical evidence on the topic – scientific as well as anecdotal.

The literature study undertaken in this research revealed, among other issues, the lack of any defining methodological process for PD, and that PD approaches to date were centred on technological

solutions that supported or partly automated the PD process. The output of the literature review was a first version of the PD process framework. The researcher organised the existing literature in PD and synthesised it to form the below framework (see Figure 1). This framework is validated theoretically through the use of existing literature from the SPL field. This theoretical validation can be seen in Chapter four through the use of theoretical arguments to support the various elements of the framework.

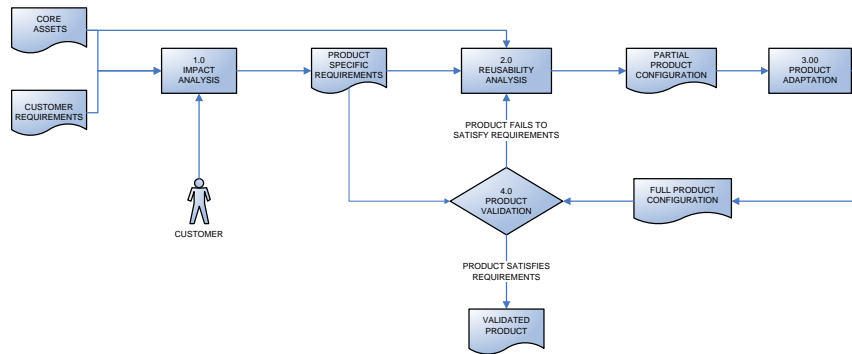


Figure 1 - Overview of initial version of PD process framework

5.2 Iterative Workshop Series

The initial framework was further developed and assessed following organised workshops with SPL and SPI experts.

The participants were:

- An academic SPL expert
- An industrial SPL expert
- An academic software architecture expert
- An academic SPI and Agile expert
- The researcher of this study

This research defines ‘expert’ as a person who has either (a) published widely in recognised journals in the field of SPL and in particular PD; or (b) has practical experience of SPL – working in the field for several years and holding positions of responsibility.

The researcher organised a series of iterative workshops over a four months period, with participants meeting twice a month. Of the five participants, a minimum of three were present at each workshop. Prior to each workshop, the framework was sent to each participant, allowing them to familiarise themselves with the version being analysed. At each workshop an interactive presentation on the framework was given by the researcher. Issues requiring clarification or which were disputed among participants were discussed. Clarification and agreement was sought on varying issues through which the framework was further developed.

5.3 Case Study on Industrial Product Derivation Practices

The case study approach has always been one of the most popular research strategies (Orlikowski et al., 1991). The case study method is a powerful and flexible technique, considered suitable for exploratory research both prospectively and retrospectively (Perry et al., 2004). A case study is especially helpful in situations where researchers are seeking to develop understandings of the dynamics of a phenomenon in its natural context (Yin, 2003). As this study wished to investigate PD practices in industry the case study methodology seemed very applicable. In addition, previously, the

researcher identified this study as an interpretive study (Section 3.6.1). Walsham (Walsham, 1993) stated that “case studies provide the main vehicle for research in the interpretive tradition”.

For the case study, the researcher collected data on the PD process of a major supplier of automotive software. The company was implementing an SPL approach to software development. The researcher met the company where he presented a short presentation on the research project and the nature of the commitment required for any participant company.

For the case study, the researcher collected data on the product derivation practices of a major supplier of automotive systems. The systems produced consisted of both hardware (such as processors, sensors, connectors, and housing) and software. Many of the requirements were derived from market segments, such as low cost or high cost customers or from regulatory requirements.

According to Yin (Yin, 2003), the use of multiple sources of evidence in a case study allows an investigator to address a broader range of historical, attitudinal, and behavioural issues. Furthermore, the multiple sources of empirical evidence provide an opportunity for triangulation in order to make any finding or conclusion of the study more convincing and accurate (Yin, 2003). Yin identifies six sources of evidence: documentation, archival records, interviews, direct-observation, participant-observation, and physical artefacts (Yin, 2003).

While conducting the case study the researcher had access the following sources of evidence:

- Company documentation
- Collective group notes from workshop
- Whiteboard drawings of company practices collected from workshop
- Researcher notes of workshop discussion

After the workshop the collected data is used to create a technical report on the company derivation practices. This technical report is currently being compiled. On completion the report will be sent to the company for validation.

5.4 Expert Opinion Assessment

The researcher plans to target key SPL experts' world wide to participate in the study. The researcher will choose experts from different backgrounds and audiences groups as recommended by Lauesen and Vinter (Lauesen et al., 2001) and Kitchenham et al. (Kitchenham et al., 2002). The researcher will aim to draw experts from both research and industry. Industry participants will be selected based on relativity, in terms of area, and their domain of expertise.

Expert opinion based assessments will be conducted through Delphi analysis, focus groups or questionnaires. With the arrival of SPLC'08 at the University of Limerick in September 08, the researcher would have a unique opportunity as the main SPL experts in the world would be visiting the university. This provides an opportunity to engage SPL experts face to face who, perhaps for geographical reasons, would typically be inaccessible for this study.

The researcher is currently constructing a list of willing participants for the assessment. Once confirmation and final numbers of willing participants is received, meeting schedules will be organised. Parallel to this, the researcher is investigating expert opinion assessment techniques for the purpose of conducting the assessment.

Feedback from the assessment would be used to refine and assess the framework.

6 RESULTS TO DATE

6.1 A Product Derivation Process Framework

The PDPF is structured into four main steps:

1. **Impact Analysis** is aimed at gathering product-specific requirements based on customer requirements and negotiation with the platform team.
2. **Reusability Analysis** purports to create a partial product configuration based on the product specific requirements and by using the available core assets.
3. During **Component Development and Adaptation**, new components are developed (if required) and existing components are adapted to satisfy requirements which could not be satisfied by configuring existing core assets.
4. Finally, **Product Integration and Validation** aims to integrate the core asset configuration and newly developed components. The integrated product is then validated by performing appropriate testing procedures.

The researcher will now discuss these product derivation steps in more detail.

6.2 Formalising the framework

The researcher is using the Eclipse Process Framework (EPF) to model the product derivation process and create a more formal version of the PDPF. EPF allows the development, maintenance and deployment of process content. It also allows the development of situational method content. By enabling inbuilt process variability within EPF you can select, tailor and or remove content from a process in order to strike the right balance for a particular situation. This has the potential for making the PDPF as applicable to a small software development team working on a mobile application as it is for large aerospace and defence contractor building a system of systems. For instance, in the case study company the researcher saw that embedded software development is a cross discipline activity. In this context, discipline mapping where requirements are allocated to software, hardware or mechanical disciplines, would be a relevant task. However discipline mapping might be different when developing a product line in another domain. This process flexibility allows the researcher to model generic product derivation practices and domain specific practices.

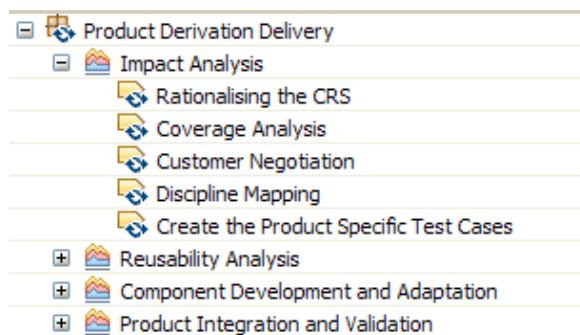


Figure 2. New Tasks in Impact Analysis

7 CONCLUSION AND FUTURE WORK

This research is motivated by the assumption that despite the adoption of SPL within industry, product derivation remains an expensive and error-prone activity.

The researcher has presented a product derivation framework which is based on an extensive literature review and discussions with SPL practitioners and researchers. The researcher studied the product derivation practices of a large automotive supplier. The observations from this case study research resulted in augmentations for our framework, especially for industries which focus on software-intensive systems (which include other aspects besides software).

The researcher hope the framework will assist organisations in applying a structured approach to product derivation activities. To progress towards that aspiration the researcher sees a multitude of potential research avenues.

The researcher's goal for the near future is to provide a version of the PDPF which is completely described electronically, i.e. in models used by the Eclipse Process Framework (EPF). First, this would allow easy access to documentation and guidelines, since these models can be published as a website. Second, it allows the researcher to employ variability in the process framework. Consequently, the models can be adapted and customized for a particular industry and organization. This is adaptation is already tool- supported by EPF Composer (The Eclipse Foundation, 2007).

In the long run, the framework itself could be used as a foundation for tools, which support the process side of product derivation. This could include support for process management, for instance, by increasing the visibility of status information on tasks, milestones and key deliverables.

To gather meaningful information with minimal additional effort, it would be desirable to integrate the PDPF tools with existing practices. For instance, the tool could detect the current tasks and artefacts the developer is working on, by interaction (hotkeys) or by analysing access operations to a version control server. Integrated with the status overview (of tasks and artefacts) the tool could provide functionality to start task-specific activities or jump directly to an artefact.

As a further aspect of the product derivation processes, the researcher is interested in the integration of agile practices with plan-driven approaches. The researcher believes that in many contexts the adoption of agile practices can improve the product derivation process. The researchers work to date in this area has identified a set of agile practices that have potential for integration into the product derivation process (O'Leary et al., 2007; O'Leary et al., 2007). The researcher believes that our framework can provide a means to balancing between agility and formalism during the product derivation process.

References

- The Eclipse Foundation. Eclipse Process Framework Project (EPF).
<http://www.eclipse.org/epf>, Accessed November 29th 2007.
- Asikainen, T., T. Männistö, et al. (2004). Using a Configurator for Modelling and Configuring Software Product Lines Based on Feature Models. 3rd Software Product Line Conference (SPLC 2004). Boston, MA,.
- Chastek, G., P. Donohoe, et al. (2002). Product Line Production Planning for the Home Integration System Example. Product Line Practice Initiative. Pittsburgh, Software Engineering Institute.
- Chastek, G. and J. D. McGregor (2002). Guidelines for Developing a Product Line Production Plan. Product Line Practice Initiative. Pittsburgh, PA, Software Engineering Institute.
- Cooper, H. M. (1984). The Integrative Research Review: A Systematic Approach. Beverly Hills, Sage Publications, Inc.
- Deelstra, S., M. Sinnema, et al. (2004). Experiences in Software Product Families: Problems and Issues During Product Derivation. Software Product Lines, Third International Conference. Boston, MA, USA, Springer.
- Deelstra, S., M. Sinnema, et al. (2005). Product Derivation in Software Product Families: A Case Study. J. Syst. Softw. New York, NY, USA, Elsevier Science Inc. **74**: 173-194.
- Griss, M. L. (2000). Implementing Product-Line Features with Component Reuse. London, UK, Springer-Verlag.

- Hakim, C. (1987). Research design : strategies and choices in the design of social research. London, Allen & Unwin.
- Highsmith, J. and A. Cockburn (2001). Agile Software Development: The Business of Innovation. IEEE Computer. **34**: 120-122.
- Hotz, L., A. Gunter, et al. (2003). A Knowledge-based Product Derivation Process and some Ideas how to Integrate Product Development. Proc. of Software Variability Management Workshop. Groningen, The Netherlands.
- Järvinen, P. (2001). On Research Methods. Tampere, Finland, Juvenes-Print.
- Kitchenham, B., S. Pfleeger, et al. (2002). "Preliminary guidelines for empirical research in software engineering." IEEE Transactions on Software Engineering **28**(8): 721 - 734.
- Krebs, T., K. Wolter, et al. (2005). Model-based Configuration Support for Product Derivation in Software Product Families. Koblenz, Germany.
- Kruchten, P. (2004). The Rational Unified Process: An Introduction, Addison-Wesley.
- Latour, B. (1988). The Pasteurization of France. Cambridge, MA, Harvard University Press.
- Lauesen, S. and O. Vinter (2001). "Preventing Requirement Defects: an Experiment in Process Improvement." Requirements Engineering Journal **6**(1): 37-50.
- McGregor, J. D. (2005). Preparing for Automated Derivation of Products in a Software Product Line, Software Engineering Institute,.
- O'Leary, P., M. Ali Babar, et al. (2007). Product Derivation Process and Agile Approaches: Exploring the Integration Potential. Proceedings of 2nd IFIP Central and East European Conference on Software Engineering Techniques, Poznań, Poland, Wydawnictwo NAKOM.
- O'Leary, P., M. Ali Babar, et al. (2007). Towards Agile Product Derivation in Software Product Line Engineering. RISE 2007, 4th International Workshop on Rapid Integration of Software Engineering techniques, Luxembourg, LUXEMBOURG.
- Orlikowski, W. and J. Baroudi (1991). "Studying Information Technology in Organizations: Research Approaches and Assumptions." Information Systems Research.
- Patton, M. Q. (1987). How to Use Qualitative Methods in Evaluation. California, Sage Publications Inc.
- Perry, D., S. E. Sim, et al. (2004). Case Studies for Software Engineers. Proceedings of the 26th International Conference on Software Engineering.
- Rabiser, R., P. Grunbacher, et al. (2007). Supporting Product Derivation by Adapting and Augmenting Variability Models. Software Product Line Conference, 2007. SPLC 2007. 11th International. Kyoto, Japan.
- Sinnema, M., S. Deelstra, et al. (2006). Modeling Dependencies in Product Families with COVAMOF. 13th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2006). Potsdam, Germany.
- Walsham, G. (1993). Interpreting Information Systems in Organizations. New York, NY, USA, John Wiley & Sons, Inc.
- Yin, R. K. (2003). Case Study Research: Design and Methods. Thousand Oaks, CA, Sage Publications Inc.