

Automated Information Systems Generation for Process-Oriented Organizations

Francisco J. Duarte
Dep. de Sistemas de Informação
Universidade do Minho
francisco.duarte@dsi.uminho.pt

Ricardo J. Machado
Dep. de Sistemas de Informação
Universidade do Minho
rmac@dsi.uminho.pt

João M. Fernandes
Dep. de Informática
Universidade do Minho
jmf@di.uminho.pt

Abstract

Currently, the development of organizational information systems remains a complex task. Final software product quality often does not match expectations. The existence of organizational models is the first step to reduce complexity in the development of information systems.

Within the life cycle of an information system, activities are still very dependent in quality, time, and costs on the human resource skills that staff them. The existence of automated mechanisms to transform client requirements into characteristics of running systems would bring added value to the resulting software product, either in product quality and time perspectives.

In this proposal, the manipulation of requirements must be done using an understandable model for both software engineers and business process experts. This model should be used to automatically reshape the running organizational information system and be the basis for an automated information system generation. The usage of such mechanism can be done during a development project, but also after its implementation where standalone process experts could change the organization model, knowing that the changes, in an automated mode, would be transferred into the running system.

1. Introduction

It is considered that an organization has only one organizational information system, despite it can be constituted of several smaller information systems each one addressing distinct parts of the organization. Otherwise, it could be inferred that if an organization had several distinct information systems running, it was not a single organization but a constellation of smaller organizations, each one mapped into a different information system. Also, the one-to-one relation between organization and information system is relevant for process-oriented organizations due to the

interfaced nature of the processes. The effort to model the organizational information system is compensated with an easier manipulation and mapping between the desired organization characteristics and the information system implemented features.

An information system must embody and implement a model of the organization where it will be used [8]. The success of implementing an information system cannot only be measured by the quality and quantity of the implemented requirements presented to the development team. The (re)design of the organization's business processes should also be considered, as well as the ability to cope with future transformations in the process landscape, when evaluating the overall achievement of the project team. Defining the landscape and the contents of the business processes implemented in the information system should not be an exclusive task of the client. The possibility of the information system to incorporate new or redesigned business processes is also a key feature. World-class business processes should always be exploited to check the business value of the requirements of the information system.

Despite the decision to choose the final characteristics of the information system is taken by the client, he should be conscientious and base the decision on explicit and, wishfully, rigorous criteria. The existence of techniques allowing to automatically transform business process models into running information systems, starting with easy understandable models by clients' users will bring competitive advantages to the clients' organizations. The advantages will be expressed, for instance, in the reduction of the needed time to implement an information system, or in the improvement in the final software product quality, being the quality understood as the degree of fulfillment of presented requirements. Requirements specifications in natural language lead to subjectivity and thus propitiating misunderstandings and decreased final software product quality. The opposite approach, formal descriptions as CPN (Colored Petri Nets) [15], inhibits a proper perception by all requirement specification stakeholders. For these two reasons, the

usage of semi-formal notations as UML (Unified Modeling Language) [24] is a first step, although not sufficient, to reduce significantly misunderstandings and to avoid inefficiency transposing clients' wishes into running information system behavior. The implementation of business processes using workflow systems is also not a fully appropriated solution for process-oriented organizations due to the absence of business semantics. Workflow systems can be used to model sub-processes, but the missing business semantics can be an escape path in an organization to implement sub-processes completely misaligned with the strategy of the organization. The existence of process models for specific types of organizations (process reference models) allows the adequate choice of a set of processes to implement in an information system. At present, COTS (Commercial-Off-The-Shelf) systems, like ERP (Enterprise Resource Planning), require significant parametrization and less programming efforts, nevertheless demanding a specific technical knowledge from software engineers. Even for skilled software engineers, ERP customization tasks are not straightforward, mainly due to the complexity and interdependency of clients' business processes.

2. State of the Art

2.1. Process-Oriented Organizations

The idea of defining processes inside organizations [13], aims to focus the organization in activities that can bring value for its clients. A traditional hierarchical organization's profile tends to focus on internal departmental concerns, frequently ignoring clients' best interests. The process-oriented organization profile (Figure 1) is designed to eliminate internal barriers (normally departments).

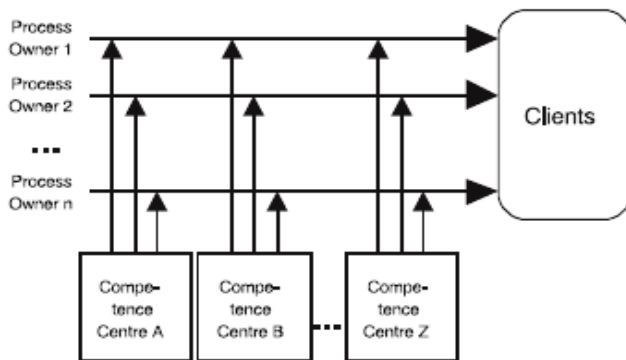


Figure 1. Generic model for a process-oriented organization [13].

Competence centers are hubs where different specific skilled human resources are trained. Processes are a co-

ordinated set of activities where needed skilled human resources, managed by a process owner, cooperate in order to bring value for clients. The design, control, and improvement in business processes can only be acceptable if a defined, manageable, and measurable process model exists. Its structure is a key element to the organization success. Consequently, process models for organizations are the most important driver for next generation ERP designs [4].

2.2. Process Reference Models

The existence of process reference models for organizations facilitates software engineering tasks, mainly in requirements and design tasks. Organizations models, like the one presented in [9], can be the common platform where software engineers and remaining stakeholders discuss organization shape. When performing requirement analysis, comparing desirable and stated characteristics for the organization model [10] is a needed activity in order to check if the model is complete enough to start an information system project. In an information system development project, namely in software engineering tasks, the usage of a meta-model for a development process allows a proper process customization into the needs of a particular project. This can be seen in RMC (Rational Method Composer) [18] where a set of pre-configurations to RUP (Rational Unified Process) [14] exists. Thus, client organization model and development process meta-model should be inputs for information systems development.

2.3. Automated Information Systems Generation

An organization model can be seen in functional, organizational, information, and resources perspectives [28]. The key for the holistic comprehension of an organization can be an integrated model [5] presenting the challenge to transform a generic description into a requirement. The adoption of an MDA (Model Driven Architecture) [21, 22] approach together with UML can bring significant savings in software development, as shown in [25]. Also, MDA, as proposed by OMG, demands the transformation of platform independent models into platform dependent models or into executable code. Since UML models are not yet semantically defined, other models, like Petri Nets, are necessary for automated code generation [23]. The existence of an organization reference model, embodying clients' requirements together with a proper language to express them, is crucial to transform the requirements model into running code. For that, a possible approach to create a transformable, and at the end executable, process reference model is to formalize it with UML [11], eventually complemented with xUML

(eXecutable UML) [3]. Similar approaches were used with success in [12, 27]. In contexts where software is to be reused, two steps are needed: First, a representation of the implemented business processes must be drawn and incorporated into the business processes model; Second, a suitable application driver must be developed, eventually using SOA [20], in order to tie the business processes model to the running software. In this last step, the necessary precautions on the introduction of new software technologies must be attained, as stated in [30].

The adoption of a formal language, like CPN, allows modeling, animation, simulation, and formal properties verification of each process inside the reference model. Also, processes inter-dependencies can be modeled, allowing the analysis, verification, and validation of the complete business process reference model. By the end, CPN skeletons can be derived from business requirements models. A semantic layer in Arena [16], able to accept business specifications based on CPN, would allow the stochastic execution of business processes as a complement to current validations based on CPN/Tools [2]. After definition, the reference model could be used in any organization to be compared with its current business process model. This assessment, based on CPN models, would be the basis for a quick audit of current organization status, comparing it with world-class reference model. This way, the reference model acts as the To-Be model and the current organization process model as the As-Is, allowing the differences to be explicitly managed. Software requirements express needs and restrictions of a software product [17] and should be understood as properties shown by the software to solve a particular problem [1]. Thus, in the development of an information system impacts from the organization model, the set of business processes, and also the desired business operational quality, are among the inputs to define the requirements for a software product. Functional requirements validation using only UML is not sufficient but, as stated in [19], executable and interactive prototypes contribute to a more rigorous and a better understanding of functional requirements by the client.

The needed confrontation between the reference model and the client model provides also a violation check of stated requirements and clients' strategy for the business. Requirement engineers should not only restrict their analysis to expressed requirements. They should also express other needs and restrictions coming from the organizational reality, restrictions from a particular software solution, or best practices from similar organizations. To achieve it, a new stakeholder ("the organizational model") must be introduced every time process requirements are expressed. Commonly it is accepted that the best interest of an organization is expressed via their managers or owners, but this does not necessarily happens, namely when overall best interest col-

lides with personal interests, or when some requirement is forgotten. With this new immaterial actor for the requirement analysis process, organization models based in best practices, re-engineering the corporation towards the company sake, and also characteristics derived from vision or mission, must always be considered.

3. Research Objectives and Approach

3.1. Research Objectives

The main research question is to define and test how much automation can be brought into information systems generation for process-oriented organizations, improving at the same time the management of project restrictions: quality, time, costs, and requirements. Some study should also be carried out to search for proper modeling languages to cope with model transformations, users perceptions, and desired formality. The motivation for the research is also triggered from the considerable amount of needed effort, and the almost complete dependency on the quality of software engineers, to complete with success an implementation of an information system, not forgetting the benefits of using that opportunity to improve the performance of an organization and its supporting information system.

The primary contribution of this proposal is:

To suggest, for process-oriented organizations, an automated mechanism for generating information systems, which should be immediately executable within the organization's software landscape.

Other contributions, understood as success criteria for the thesis, can also be stated:

- Propose the outline and the contents for correct, complete, and executable information system models;
- Use the information system model as the shape for the client organization model and as the crucial valuator for functional requirements;
- Define pre-conditions to start the development project formation system for process-oriented organizations; Suggest precise and automated methods for information system requirements evaluation.

3.2. Research Approach

The classical scientific method depends upon theory formation followed by experimentation and observation to provide a feedback loop to validate, modify and improve the theory. This procedure can be followed, and it is appropriated, also for software engineering research [29].

Several methods can be used to conduct software engineering research [26]:

- The scientific method: observe the world, propose a model or theory of behavior, measure and analyze, validate hypotheses of the model or theory, and if possible repeat;
- the engineering method (evolutionary paradigm): observe existing solutions, propose better solutions, build or develop, measure and analyze, repeat until no further improvements are possible;
- the empirical method (revolutionary paradigm): propose a model, develop statistical or other methods, apply to case studies, measure and analyze, validate the model, repeat;
- the analytical method: propose a formal theory or set of axioms, develop a theory, derive results and if possible compare with empirical observations.

For this thesis, the proposed research method (incorporating aspects of distinct methods) is to first evaluate the state of the art and use an organization to act as the case study, followed by a complete definition of the “problem”. Afterward, the proposals in the thesis will be verified, quantified, and validated with the help of a demonstration case, and the relevant lessons learned embedded, eventually starting a new small cycle, in the final thesis proposal. This procedure is also based on Shewhart and Deming’s PDCA (Plan, Do, Check, Act) [6] methodology.

4. Current Work and Preliminary Results

Until the present, the main research activities were directed towards the definition of a reference framework for process-oriented organizations, as seen in the published work [7, 8, 9] where general organizational reference frameworks are presented and, afterward, materialized for specific types of business organizations (a software development organization whose processes were based on RUP disciplines, when available). Only after defining the business semantics, the transformation of a process-oriented business model into a running information system can be properly accomplished.

5. Work Plan and Implications

5.1. Step 1: Preparation

Start: November, 2006. End: August, 2007.

This step is where the main activities for literature review are carried on. Main areas of investigation are: organizations’ models (namely for process-oriented ones), languages to describe and automate business processes, business processes modeling, and methods of software engi-

neering (according to the SWEBOK). The empirical knowledge is obtained through a case study observation and analysis.

Goal: Theoretical and practical preparation.

Success Criteria: State of the art knowledge is acquired. Case study analysis derives in a clear perception of the main issues for information systems generation in process-oriented organizations.

5.2. Step 2: Thesis Proposals

Start: September, 2007. End: September, 2008.

The knowledge body and methodologies to accomplish the thesis proposal will be defined in this step. It is supposed to develop and validate the principle that a running information system can be generated and modified in a quick, automated, and quality manner. This manipulation can be done with business process experts and without the intervention on software experts.

Goal: Based on Step 1, define and create a ground basis for the thesis arguments.

Success Criteria: All the knowledge body and methodologies (probably with software tools) is finished.

5.3. Step 3: Demonstration Case

Start: October 2008. End: April 2009.

Step 2 proposals are implemented in a real scenario of a process-oriented organization.

Goal: Use knowledge and tools defined in Step 2, in order to be possible to extract conclusions about their validity.

Success Criteria: At least one subset of the information system was generated in an automated way and based on the organization’s business processes.

5.4. Step 4: Lessons Learned and Text Revision

Start: May 2009. End: September 2009.

Lessons learned from demonstration case are extracted and, eventually, a revision on proposals is done and implemented in the demonstrations case, fulfilling a new small PDCA cycle. All thesis text is revised.

Goal: With a critical perspective, the demonstration case results are analyzed and incorporated into the thesis text and also into the running information system.

Success Criteria: The textual document to obtain the PhD is finalized.

5.5. Step 5: PhD Defense

Start: October 2009.

6. Conclusions

This proposal intends to validate some general principles related to the creation and management of information systems for process-oriented organizations, namely: Defined and explicitly managed business processes are an asset for the organization; The shape and the contents of their processes must be aligned with their mission, their vision, and their strategic goals; The running software information system must properly implement business processes. The usage of a case study should provide an empirical laboratory to extract and consolidate experiences. Subsequently, a demonstration case will be used to provide an environment for a minimal validation of the proposals. It is planned to apply some automated mechanisms to transform the business process model of a process-oriented organization (Blaupunkt Auto-Rádio Portugal, Lda.) into its information system.

Additionally, a suitable language for representing business processes models and a transformation process between the running software system and the business processes model must be chosen.

References

- [1] A. Abran and J. Moore. *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.
- [2] M. Beaudouin-Lafon, W. Mackay, P. Andersen, P. Janecek, M. Jensen, M. Lassen, K. Lund, K. Mortensen, S. Munck, A. Ratzler, et al. CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets. *Application and Theory of Petri Nets*, 2075:71–80, 2001.
- [3] K. Carter. Supporting Model Driven Architecture with eExecutable UML. *Ref: CTN*, 80.
- [4] N. Dalal, M. Kamath, W. Kolarik, and E. Sivaraman. Toward an integrated framework for modeling enterprise processes. *Communications of the ACM*, 47(3):83–87, 2004.
- [5] D. Delen, N. Dalal, and P. Benjamin. Integrated modeling: the key to holistic understanding of the enterprise. *Communications of the ACM*, 48(4):107–112, 2005.
- [6] W. Deming. *Out of the Crisis*. MIT Press, 2000.
- [7] F. Duarte, J. Fernandes, and R. Machado. *Reference Modeling for Business Systems Analysis*, chapter Business modeling in process-oriented organizations for RUP-based software development. Idea Group Pub., an imprint of Idea Group Inc., 2007.
- [8] J. Fernandes and F. Duarte. Using RUP for Process-Oriented Organisations. *5th Int. Conf. on Product Focused Software Process Improvement (PROFES 2004)*, *Lecture Notes in Computer Science*, 3009:348–62, 2004.
- [9] J. Fernandes and F. Duarte. A reference framework for process-oriented software development organizations. *Software and Systems Modeling (SoSym)*, 4(1):94–105, 2005.
- [10] P. Fettke, P. Loos, and J. Zwicker. Business Process Reference Models: Survey and Classification. *Workshop on Business Process Reference Models (BPMR 2005)*, pages 1–15, 2005.
- [11] P. Fettke, P. Loos, and J. Zwicker. Using UML for Reference Modeling. *Enterprise Modeling and Computing with UML. Hershey, PA, USA, et al*, pages 174–205, 2007.
- [12] V. Gruhn and U. Wellen. Process Landscaping: Modelling Distributed Processes and Proving Properties of Distributed Process Models. *Lecture Notes in Computer Science*, 2128(S 103), 2001.
- [13] M. Hammer. Beyond reengineering: how the process-centred organization is changing our work and our lives. 1996.
- [14] I. Jacobson, G. Booch, and J. Rumbaugh. *The unified software development process*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1999.
- [15] K. Jensen. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. *EATCS Monographs in Computer Science*, 26, 1992.
- [16] W. Kelton, R. Sadowski, and D. Sadowski. *Simulation with Arena*, WCB, 2002.
- [17] G. Kotonya and I. Sommerville. *Requirements engineering: processes and techniques*. J. Wiley & Sons, 1998.
- [18] P. Kroll. Introducing IBM Rational Method Composer. *The Rational Edge*, 2005.
- [19] R. Machado, K. Lassen, S. Oliveira, M. Couto, and P. Pinto. Requirements Validation: Execution of UML Models with CPN Tools. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(3):353–369, 2007.
- [20] C. MacKenzie, K. Laskey, F. McCabe, P. Brown, and R. Metz. Reference Model for Service Oriented Architecture. *OASIS Committee Draft*, 1, 2006.
- [21] T. Meservy and K. Fenstermacher. Transforming software development: an MDA road map. *Computer*, 38(9):52–58, 2005.
- [22] J. Miller and J. Mukerji, editors. *Technical Guide to Model Driven Architecture: The MDA Guide v1.0.1*. June 2003.
- [23] S. Philippi. Automatic code generation from high-level Petri-Nets for model driven systems engineering. *The Journal of Systems & Software*, 79(10):1444–1455, 2006.
- [24] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language reference manual*. Addison-Wesley Longman Ltd. Essex, UK, UK, 1998.
- [25] S. J. Sewall. Executive Justification for Adopting Model Driven Architecture (MDA), November 2003.
- [26] W. Tichy, N. Harbermann, and L. Prechelt. Future directions in software engineering. *ACM SIGSOFT, Software Engineering Notes*, 18(1):35–48, 1993.
- [27] W. van der Aalst. Challenges in business process management: Verification of business processes using Petri nets. *Bulletin of the EATCS*, 80:174–199, 2003.
- [28] F. Vernadat. *Enterprise Modeling and Intergration: principles and applications*. Springer, 1996.
- [29] M. Zelkowitz and D. Wallace. Experimental validation in software engineering. *INF SOFTWARE TECHNOL*, 39(11):735–743, 1997.
- [30] M. Zelkowitz, D. Wallace, and D. Binkley. Experimental validation of new software technology. *as a chapter in SE Empirical Validation (Natalia Juristo and Ana Moreno eds.)*, 2003.