TCoB & MEIS 2007

Shazia Sadiq, Manfred Reichert,
Karsten Schulz, Jos Trienekens,
Charles Moller and Rob J. Kusters (Eds.)

Technologies for Collaborative Business Processes and Management of Enterprise Information Systems

Proceedings of the
1st International Joint Workshop on
Technologies for Collaborative Business Processes and
Management of Enterprise Information Systems
TCoB & MEIS 2007
In conjunction with ICEIS 2007
Funchal, Portugal, June 2007

INSTICC Press

Shazia Sadiq,
Manfred Reichert,
Karsten Schulz,
Jos Trienekens,
Charles Moller and
Rob J. Kusters (Eds.)

Technologies for Collaborative Business Processes and Management of Enterprise Information Systems

Proceedings of the
1st International Joint Workshop on
Technologies for Collaborative Business Processes and
Management of Enterprise Information Systems
TCoB & MEIS 2007

In conjunction with ICEIS 2007
Funchal, Madeira, Portugal, June 2007

INSTICC PRESS
Portugal

Volume Editors

Shazia Sadiq, The University of Queensland, Australia

Manfred Reichert, University of Twente, The Netherlands

Karsten Schultz, SAP Research, Australia

Jos Trienekens, University of Technology Eindhoven/KEMA Quality, The Netherlands

Charles Møller, Aarhus School of Business, Denmark

and

Rob J. Kusters, University of Technology Eindhoven/Open University, The Netherlands

1st International Joint Workshop on
Technologies for Collaborative Business Processes
and Management of Enterprise Information Systems
Funchal, Madeira, Portugal, June 2007
Shazia Sadiq, Manfred Reichert, Karsten Schulz, Jos Trienekens,
Charles Moller and Rob J. Kusters (Eds.)

Printed in Portugal

ISBN: 978-972-8865-99-3
Depósito Legal: 258806/07

Foreword

Enterprise Information Systems (EIS) like ERP-, CRM- and SCM-systems are pervasive in today's organizations. EIS implementation projects ask for major financial investments of organizations. They impact many processes and employees of an organization and require significant organizational change. EIS are not only a pervasive phenomenon from a financial point of view, but they also embed a high level of complexity from a technical, an organizational and from a business perspective. As a consequence many heterogeneous managerial competences are required.

The MEIS workshop addresses the approaches, methods and techniques applied in order to analyze and manage Enterprise Information System. Areas of interest to this workshop included: Governance of EIS, Project management, Human and organizational factors, Critical success factors (CSF) and critical failure factors (CFF), Change management, Implementation characteristics in SMEs, Implementation scenarios; Risk analysis and management, IT service specification, ITIL applications, Service level agreements, Power and politics in EIS implementations, Maturity levels of EIS implementation, EIS maintenance and post-implementation issues, Case studies on successful & failed implementations, Role of the IS function during implementation, Business process management, Methodological aspects of EIS research, Innovative approaches to EIS management.

Collaborative business processes play a dominant role in enterprise information systems. In general process enforcement technologies are considered as one of the key success stories in providing process control and monitoring functions, and addressing complex integration requirements in enterprise systems. However, in current practice, process management often spans organizational and infrastructure boundaries. This is typically the outcome of business activity sharing, outsourcing, and trading partner collaborations. The resultant collaborative business processes pose a new set of challenges and warrant targeted attention from research and industry.

The intention of this workshop was to provide a forum wherein challenges in modeling and deployment of collaborative business processes can be debated. Areas of interest to this workshop included: Technologies for modeling and analysis of collaborative processes, E-service coordination and composition models, Cross-organizational process management, Event driven process management, Adaptive process management

Context-aware collaborative processes, Ontological aspects of collaborative processes, Knowledge management in collaborative processes, Middleware for collaborative process management, Architectures and implementations for collaborative processes, EAI and B2B technologies, Usability and technology adoption of BPM solutions, Business Process Scenarios: Description, Analysis, Classification.

We would like to acknowledge the support of the respective workshop program committees in the paper review process. We also thank Olivier Camp and Slimane Hammoudi as general workshops chairs. Finally, we would like to acknowledge the support of the ICEIS organization, in particular Vitor Pedrosa for the help in workshop organization and preparation of the workshop proceedings. We hope you will find the papers in this volume interesting and stimulating.

June, 2007

Shazia Sadiq, The University of Queensland, Australia

Manfred Reichert, University of Twente, The Netherlands

Karsten Schultz, SAP Research, Australia

Jos Trienekens, University of Technology Eindhoven/KEMA Quality, The Netherlands

Charles Møller, Aarhus School of Business, Denmark

Rob J. Kusters, University of Technology Eindhoven/Open University, The Netherlands

Workshop Chairs

Shazia Sadiq, The University of Queensland, Australia

Manfred Reichert, University of Twente, The Netherlands

Karsten Schultz, SAP Research, Australia

Jos Trienekens, University of Technology Eindhoven/KEMA Quality, The Netherlands

Charles Møller, Aarhus School of Business, Denmark

and

Rob J. Kusters, University of Technology Eindhoven/Open University, The Netherlands

Invited Speakers

Wil van der Aalst, Eindhoven University of Technology, The Netherlands
Charles Møller, Aarhus School of Business, Denmark

Program Committee

Technologies for Collaborative Business Processes
Boualem Benatallah, University of NSW, Australia
Giorgio Bruno, Politecnico di Torino, Italy
Dirk Werth, Business Integration Institute for Information Systems, Germany
Marlon Dumas, Queensland University of Technology, Australia
Marta Indulska, University of Queensland, Australia
Richard Lenz, University of Marburg, Germany
Olivera Marjanovic, University of NSW, Australia
Jan Mendling, University of Economics Vienna, Austria
Olivier Perrin, Nancy 2 University, France
Stefanie Rinderle, University of Ulm, Germany
Ruggaber Rainer, SAP Research

Julien Vayssiere, SAP Research
Barbara Weber, University of Innsbruck, Austria
Andreas Wombacher, University of Twente, The Netherlands

Management of Enterprise Information Systems
Jos Trienekens, Eindhoven University of Technology/KEMA Quality, The Netherlands
Charles Møller, Aarhus School of Business, Denmark
Paolo Faverio, Carlo Cattaneo – LIUC University, Italy
Donatella Sciuto, Politecnico di Milano and Carlo Cattaneo – LIUC University, Italy
Rob Kusters, Eindhoven University of Technology/Open University, The Netherlands
Fred Heemstra, Open University, The Netherlands
Michiel van Genuchten, Philips, The Netherlands
Katalin Balla, Budapest Technical University, Hungary
Hans Moonen, Erasmus University, The Netherlands
Gerico de Vries, KEMA Quality, The Netherlands
Wouter Kuijpers, ACA Retail BV, The Netherlands
Ruud Hendriks, OCE, Germany
Jette Lundin, Aarhus School of Business, University of Aarhus, Denmark
Bjarne Rerup Schlichter, Aarhus School of Business, University of Aarhus, Denmark
Ton Weijters, Eindhoven University of Technology, The Netherlands

Table of Contents

Papers

PAPERS

Binary Collaboration Models related to Manual Activities

Giorgio Bruno

Dip. Automatica e Informatica, Politecnico di Torino, Torino, Italy
giorgio.bruno@polito.it

Abstract. The operational implications of manual activities in business processes lead to an interesting case study on binary collaborations in which situations of clashing interactions appear. This paper analyzes three situations involving manual activities and proposes three collaboration patterns to deal with them. These patterns are presented by means of a notation based on colored Petri nets in which most transitions are related to interactions. A solution to the problem of clashing interactions is illustrated as well.

1 Introduction

Collaboration models are meant to specify how a number of participants have to interact in order to achieve a common goal. Binary collaboration models are concerned with the interactions taking place between two participants and they can be the starting point for building multi-party collaboration models, as discussed in [1].

Various approaches and notations have been proposed, but they basically fall into two categories: 1) a collaboration model can be presented from the point of view of one participant (the point of view of the other participant being complementary), and hence it is made up of communication actions (i.e. sending actions and receiving ones); 2) or it can be formalized from a neutral perspective based on interactions. An interaction is a kind of abstract entity as in reality it encompasses a sending action on one side and a receiving action on the other side. Behavioral interfaces [2] belong to the first category, while UMM [3] and WS-CDL [4] fall in the second one.

In both cases a collaboration models consists of a number of communication entities (communication actions or interactions) placed in a suitable control structure. For this reason, in principle, the structure of collaboration models is similar to that of business processes and the results, such as the workflow patterns [5], obtained from the study of the latter can be applied to the former as well.

This paper analyzes the collaboration taking place between two major components of a process-driven information system, i.e. the Process Manager (PM) and the Task Manager (TM). A process-driven information system, also referred to as a PAIS (Process-Aware Information System [6]), is basically an information system equipped with business processes, which specify the order in which manual activities and automatic ones have to be performed. Business processes are formal descriptions to be interpreted by PM which, in fact, orchestrates their execution by calling the services of the underlying information system (for the automatic activities) and by interacting with TM for the manual activities. TM is in charge of mapping the manual activities, as seen by PM, to actual tasks, as seen by the users, and of notifying PM of the results provided by such tasks.

This paper analyzes three situations involving manual activities, from the point of view of the collaboration between PM and TM, and proposes three collaboration patterns addressing these situations. Such patterns are presented by means of a notation based on colored Petri nets [7], in which most transitions are related to interactions.

These patterns show circumstances in which two or more transitions are enabled and they refer to interactions in opposite directions; this means that the participants may take different paths and, as a consequence, they will produce clashing interactions. This paper presents an approach, based on priorities, which enables the participants to reconcile.

The organization of this paper is as follows: section 2 introduces the notation used for collaboration models and explains how to deal with clashing interactions; section 3 presents the collaboration patterns between PM and TM; section 4 makes a comparison with related work; section 5 presents the conclusion.

2 Modeling Binary Collaborations

This section introduces the notation used in this paper to represent binary collaborations.

A binary collaboration takes place between two participants (i.e. two business processes), which are abstractly referred to as the left participant (LP) and the right one (RP). The reason for these terms is that binary collaborations are usually exemplified by means of UML sequence diagrams having exactly two roles, one on the left and the other on the right.

The notation is based on a special kind of colored Petri nets [7], called i-nets, as shown in the examples presented in Figure 1. Transitions fall into two categories: regular transitions and interaction-oriented ones. In Figure 1, only interaction-oriented transitions appear. Interactions can be one-way or two-ways, although only the first category appears in this paper. A one-way interaction indicates the name of the message and its direction. For example, transition t1 contains interaction *a* followed by a right arrow (→): this indicates that the interaction is initiated by LP, which sends message *a* to RP. The left arrow (←) indicates that the interaction is initiated by RP. The description of messages is omitted for the sake of simplicity. The initial node is connected to the first transition.

Binary collaboration models can be interpreted as abstract behavioral models of the participants as far as the interactions are concerned. In fact, in the model shown in Figure 1.a, LP is assumed to send *a* and then wait for either *b* or *c*, while RP is assumed to receive *a* and then send either *b* or *c*. Therefore LP makes the first move (by sending *a*) and RP makes the second one (by sending *b* or *c*). Both participants follow the same path in the net and take the lead in turn.

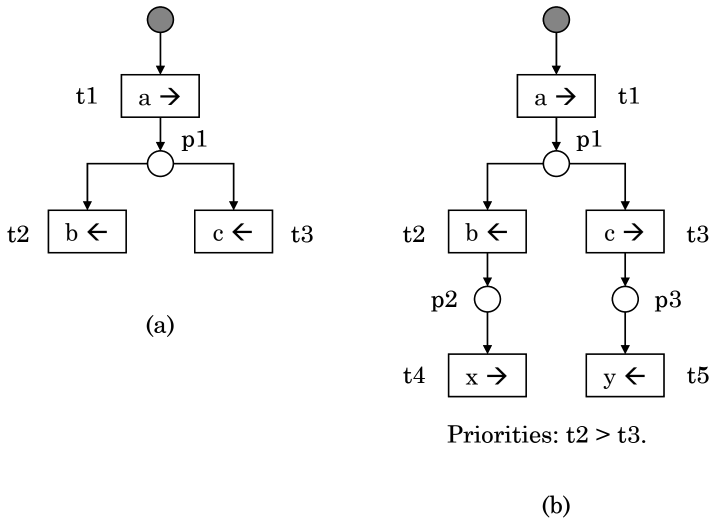


Fig. 1. Examples of collaboration models.

There are situations, however, in which the participants may take different paths at a fork and, as a consequence, they will produce clashing interactions.

A fork is a situation (more specifically, a marking in the net) in which two or more transitions are enabled. If they refer to interactions in the same direction (such as t2 and t3 in Figure 1.a), all of them are initiated by the same participant: this participant makes the choice, and the other one follows.

On the contrary, if there are two or more transitions enabled and they refer to interactions in different directions (such as t2 and t3 in Figure 1.b), both participants may initiate an interaction at the same time, thus taking different paths.

A way of reconciling the participants is needed; this is based on the priorities given to the clashing transitions, as discussed with the help of Figure 1.b. From place p1, the two participants may follow different paths: RP may decide to send *b* at the same time as LP is sending *c*.

The clash can be detected in several ways: for example, if each message is followed by an ack from the receiver, the clash is detected by RP, if it receives *c* instead of *ack(b)* after sending *b*, and by LP, if it receives *b* instead of *ack(c)* after sending *c*. If the priority is given to message *b*, as shown by means of annotation "t2 > t3", the right path is the one taken by RP, so LP is to abandon its path and accept interaction *b*. In this example, RP turns out to be the winner, while LP is the loser.

As to message *c*, which has actually been sent by LP and received by RP, there are two ways of handling it: it can be accepted anyway or it can be rejected. In the first case, RP is meant to process message *c*, while in the second case, RP will ignore message *c* and LP will undo the actions related to the production of *c*. The first behavior is the standard one; the second behavior is indicated by a "-" following the lower-priority transition. Therefore if the annotation in Figure 1.b were "t2 > t3-", message *c*, in case of clash, would be completely ignored.

The annotation must establish the priorities for all the clashing transitions in the net.

A particular function of colored tokens in i-nets is to manage the correlations among the messages, as follows. A collaboration model, such as the one in Figure 1.a, exists in several instances, in the sense that several collaborations of that kind can take place simultaneously between the same pair of parties. For example, at a given instant, party A has sent a number of messages *a* to party B, while no message *b* has been sent yet; this means that several transitions t1 have been performed and several tokens are in place p1. When party B replies with a message *b*, which *a* is this *b* a reply to?

The solution adopted in i-nets is based on correlation values carried with the messages. When a transition receives a message, it reads the correlation value from the message and sets the correlation attributes in the outgoing tokens. Therefore, when transition t1 receives a message *a*, it delivers a token to place p1 having its correlation attribute equal to the value read from message *a*. When transition t2 receives a message *b*, it matches the message with the proper input token on the basis of the correlation value of *b* and the correlation attributes of the tokens in p1, and removes that token from p1.

The standard color is called CC (correlation color): it has no additional attributes and is omitted from the models for the sake of simplicity.

3 Collaboration Patterns

A business process is a collection of interrelated activities which fall into three major categories: manual activities, automatic activities and control-flow ones.

A manual activity is carried out by a specific (human) user with the help of a suitable GUI (graphical user interface). The “task” term is often used to denote a manual activity as perceived by the user in charge of it. A task encompasses all the low-level interactions a user has to carry out through the GUI (e.g. entering textual information, selecting items from lists, giving commands) in order to achieve a particular purpose, such as placing an order.

Automatic activities are carried out without human intervention. Control-flow activities are special automatic activities which control the flow of action; decision nodes, merge nodes, fork nodes and join ones are common examples.

There are several notations and XML-based languages addressing business processes: the former include BPMN [8], while the latter include XPDL [9].

In traditional information systems (i.e. those in which processes are not explicitly defined) a user can select the task to carry out through a menu showing all the possible tasks compatible with his/her role. In a process-driven information system, there is a new category of tasks, called process-driven tasks, corresponding to the manual activities defined in the business processes. Such tasks are usually presented to the user in a list, called to-do list showing the name of the task as well as some specific information.

From a logical point of view, three major software components are involved in the operation of a process-driven information system: the Process Manager (PM), the Task Manager (TM) and the Enterprise Information System (EIS). PM interprets the business processes (e.g. their XML representations) and orchestrates them by interacting with TM and EIS. EIS is a conceptual entity encompassing all the software components needed to operate on the underlying business objects. In order to perform an automatic activity, PM requests a service of EIS, since a business process has limited processing capabilities. When a manual activity has to be carried out, PM requires a service of TM, so as to include the task description in the performer's to-do list.

Manual activities indicate the rule for selecting the performer (i.e. the user appointed to fulfil the task). In most cases, the performer is selected among the users playing a certain role, often on the basis of a load-balancing criterion. However, there are situations in which the selection depends on the information flow: for example, the performer of task "accept order" is the account manager of the client organization that issued the order to be examined. The resource patterns presented in [10] provide a number of techniques for the selection of the performer.

This section analyzes three situations involving manual activities, from the point of view of the collaboration between PM and TM, and proposes three collaboration patterns addressing these situations.

In the models below, PM is the left participant and TM the right one.

Pattern SASR (Single Activity – Single Result)

This is the most frequent case, in which a manual activity corresponds to a single task. For example, activity "accept order" brings about a task for a single performer (i.e. the account manager of the client organization that issued the order).

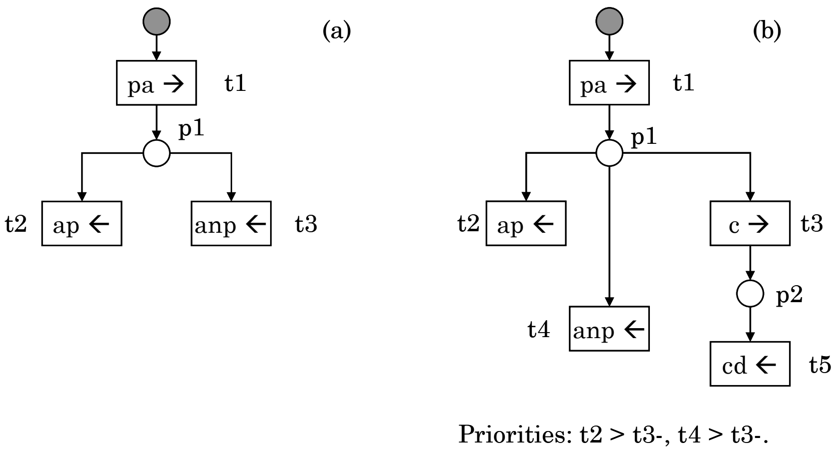


Fig. 2. Collaboration pattern SASR.

Two models are shown in Figure 2, the first is the basic one, and the second exhibits clashing transitions due to cancellation message *c*.

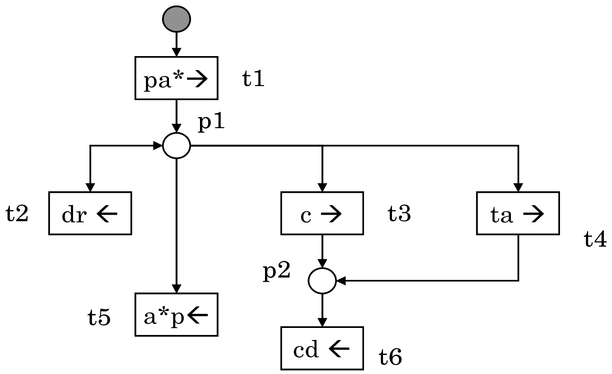
In the first model, shown in Figure 2.a, PM starts the collaboration by sending message *pa* (perform activity), then waits for either message *ap* (activity performed) or message *anp* (activity not performed) to come.

The parameters of message *pa* include: the name of the task, the performer, the business objects to be acted on, and the deadline. The parameters of the other messages are omitted for the sake of simplicity. The meaning of message *ap* is that the activity required has been successfully performed and a result has been produced: the information on the result is returned in the message. Message *anp* returns the reason why the task has not been performed (e.g. deadline elapsed or user's refusal).

In the second model, shown in Figure 2.b, PM can send cancellation message *c* so as to make TM cancel the task, and TM replies with message *cd* (cancellation done). However TM may be sending message *ap* (or message *anp*) at the same time as PM is sending message *c*, so a clash may happen. In this case, as the annotation gives priority to messages *ap* and *anp*, the cancellation message is ignored and the collaboration ends normally.

Pattern SAMR (Single Activity – Multiple Results)

As an example, manual activity "review paper" is meant to make a conference paper reviewed by a number of reviewers (those associated with the paper). TM will produce several actual tasks, one for each reviewer involved, and will notify PM of each review produced.



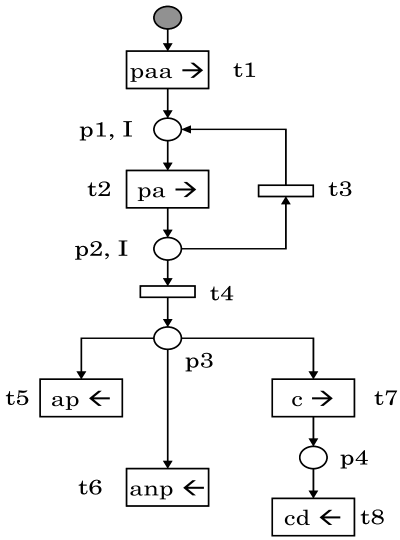
Priorities: t3 > t2, t4 > t2, t5 > t3-, t5 > t4-.

Fig. 3. Collaboration pattern SAMR.

The model presented in Figure 3 shows that the collaboration is started by message *pa** (perform activity with multiple results) and is normally followed by a number of messages *dr* (deliver result) and one final message *a*p* (activity performed). However, at any time, PM can decide to cancel the whole activity or only the remaining tasks (if it does not want to receive any more results): in the first case it will send message *c*, in the second case message *ta* (terminate activity). On the basis of the priorities presented in Figure 3, if a clash occurs between message *dr* and messages *c* or *ta*, *dr* is accepted by PM, but TM follows the path chosen by PM and replies with message *cd*; on the other hand, if a clash occurs between message *a*p* and messages *c* or *ta*, message *a*p* prevails and the collaboration ends normally.

Pattern AASR (Alternative Activities – Single Result)

This pattern addresses situations in which several activities are possible, but as soon as one is started, the others are disabled. For example, if the activity of charging an expense, such as a travel expense, to a project is being performed, the activity of closing the project has to be disabled and vice versa. Such situations are also known as deferred choices [5].



Priorities: t5 > t7-, t6 > t7-.

Fig. 4. Collaboration pattern AASR.

The collaboration is started by message *paa* (perform alternative activities) and is followed by a number of messages *pa*, one for each alternative activity. Places p1 and p2 are colored and color I includes an integer value counting the number of messages *pa* to be transmitted. Transitions t3 and t4 are regular transitions and the inscriptions on their arcs are omitted. The subnet originating from place p3 is similar to the one originating from place p1 in Figure 2 and therefore they have similar interpretations.

4 Comparison with related Work

Collaborations are based on interactions (either one-way interactions or two-way ones). The RosettaNet consortium [11] provides a rich catalog of (business) interactions including the specification of the business documents to be exchanged and of the quality of service requirements.

Collaboration models include control-flow elements, besides the interactions. The UMM [3] modeling methodology represent collaboration models, called business collaboration protocols, as UML activity diagrams and provides four control-flow elements: decision, merge, fork and join. Well known XML representations based on interactions are the Business Process Specification Schema (BPSS) [12] and WS-CDL [4].

A recent notation, "Let's Dance" [13], proposes an original control-flow structure based on three binary relationships, called precedence, weak precedence and inhibition.

None of the above mentioned notations and languages tackles the issue of clashing interactions.

The Language Action perspective [14] criticizes the interactions, as conceived by the above mentioned notations and languages, for being too rigid and adds a conversational structure enabling the parties to communicate in search of mutual agreement.

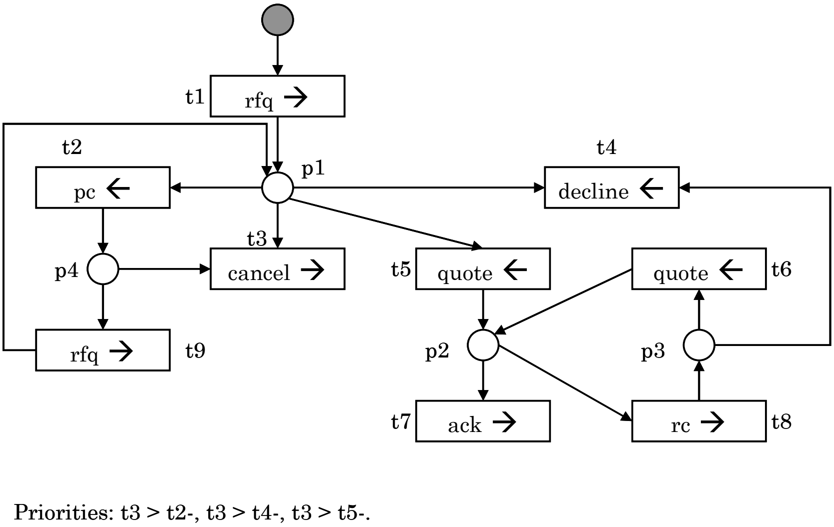


Fig.5. A conversational model based on i-nets.

The "conversation for action" approach [15], based on speech act theory [16], represents a conversation as a state model between two roles, referred to as the requester and the provider.

An example of conversational state model is shown in Figure 5 with i-nets.

The requester initially sends a request for quote (*rfq*) and then the provider may decline, or reply with a quote or request a clarification (*pc*). In the last case, the requester can provide a new *rfq* or terminate the conversation (*cancel*); the new *rfq* is assumed to contain the reply to the clarification request. After receiving a quote, the requester can acknowledge it (*ack*), or request a clarification (*rc*); in the latter case, the provider may send a new quote or decline; the new quote is assumed to contain the reply to the clarification request. Before receiving the first quote, the requester may terminate the conversation (*cancel*), and this implies a situation of clashing interactions as shown in Figure 5.

The annotation gives priority to message *cancel* which prevails over messages *pc* and *decline* in case of a clash.

5 Conclusion

This paper has presented a notation (i-nets) based on colored Petri nets in order to cope with the collaboration needed between two major components of a process-driven information system, i.e. the Process Manager (PM) and the Task Manager (TM).

Three collaboration patterns, related to the implementation of manual activities, have been presented. These patterns show circumstances in which two or more transitions are enabled and they refer to interactions in opposite directions; this means that the participants may take different paths and, as a consequence, they will produce clashing interactions. This paper has presented an approach, based on priorities, which enables the participants to reconcile.

Current work is being devoted to taking advantage of the expressive power of i-nets in order to extend the interactions presented in this paper with conversational features, on the basis of the Language Action perspective.

Acknowledgements

The author would like to thank the reviewers for their helpful suggestions.

References

1. Bruno Giorgio, Bruno Giulia, La Rosa, M.: On Collaborations and Choreographies. Proc. 1st Int. Workshop on Technologies for Collaborative Business Process Management. INSTICC Press (2006) 3-12
2. Barros, A., Dumas, M., Oaks, P.: Standards for Web Service Choreography and Orchestration: Status and Perspectives. Lecture Notes in Computer Science, Vol. 3812. Springer, Berlin (2006) 61–74
3. UN/CEFACT: Modeling Methodology (UMM) User Guide. Retrieved April 10, 2007, from http://www.unece.org/cefact/umm/UMM_userguide_V20030922.pdf
4. Kavantzas, N., Burdett, D., Ritzinger, G. : Web Services Choreography Description Language (WS-CDL), Version 1.0. Retrieved April 10, 2007, from http://www.w3.org/TR/ws-cdl-10/
5. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases, Vol. 14(2003) 5-51
6. Dumas, M., van der Aalst, W.M.P., ter Hofstede: Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley (2005)
7. Kristensen, L.M., Christensen, S., Jensen, K.: The Practitioner's Guide to Coloured Petri Nets. International Journal on Software Tools for Technology Transfer, Vol. 2. Springer, Berlin (1998) 98-132
8. OMG: Business Process Modeling Notation (BPMN), Final Adopted Specification, February 2006. Retrieved April 10, 2007, from http: /www.bpmn.org/
9. Workflow Management Coalition: XML Process Definition Language, Version 2.00, 2005. Retrieved April 10, 2007, from http://www.wfmc.org

10. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow Resource Patterns: Identification, Representation and Tool Support. Lecture Notes in Computer Science, Vol. 3520. Springer, Berlin (2005) 216–232
11. RosettaNet, HomePage, Retrieved April 10, 2007, from http://www.rosettanet.org
12. ebXML Business Process Specification Schema (BPSS), version 1.01. Retrieved April 10, 2007, from http://www.ebxml.org/specs/ebBPSS.pdf
13. Zaha, J.M., Barros, A., Dumas, M., ter Hofstede, A.H.M.: Let's Dance: A Language for Service Behavior Modeling. Proceedings of the 14th International Conference on Cooperative Information Systems (CoopIS), Montpellier, France, October 2006
14. Te'eni, D.: The language-action perspective as a basis for communication support systems, Comm. of the ACM, Vol. 49 (2006), 65-70
15. Winograd, T.: A Language/Action Perspective on the Design of Cooperative Work. Human-Computer Interaction, Vol. 3 (1987-1988) 3-30
16. Searle, J.R.: Speech acts. Cambridge University Press (1969)

KAT based CAD Model of Process Elements for Effective Management of Process Evolution

Jeewani Anupama Ginige, Athula Ginige and Uma Sirinivasan

School of Computing and Mathematics, University of Western Sydney, Locked Bag 1797
Penrith South DC NSW 1797, Australia
{j.ginige, a.ginige, u.sirinivasan}@uws.edu.au

Abstract. Processes consist of actions, participants, object and rules, known as elements. In a process, these elements are inter-woven together to achieve desired business goals. When managing process evolutions and changes, it is imperative to understand constraints, associations and dependencies (CAD) among process elements. Use of high-level graphical model that encapsulate CAD among process elements, as given in [1], is limited in practice. Therefore, here we present formalism, to model CAD among process elements. This formalism is based on constraint modeling algebra named Kleene Algebra with Tests (KAT) [2]. This paper gives a set of definitions to capture CAD among process elements based on KAT. These definitions are used to create a single compact KAT expression that captures all possible CAD among process elements. The holistic and cohesive nature in capturing CAD among process elements and deploying KAT to model them into a single expression are the unique contributions of this research.

1 Introduction and Background

Business processes evolve and change to cater diverse needs. When processes are automated, changes and evolutions need to be reflected in the automated systems (workflows [3]). Changing a process means changing process elements - *actions*, *participants*, *object* and *rules*. In a process, these elements do not exist in isolation. To satisfy various business goals, elements are inter-woven and linked. Thus, making a change (modifying, adding or removing) one element may result in propagating impact on other elements. Therefore, it is important to have a mechanism to capture various constraints, associations and dependencies (CAD) among process elements.

In this research, the focus is on exploring a formalism to capture CAD among process elements. This work is presented as part of the solution of *evolution meta-model,* which support the effective management process evolution in web-based workflows systems (WWS) [4]. In this work, process evolution is discussed in relation to the *framework of process automation* [4]. Based on this framework, previously we developed a *graphical CAD model among process elements* [1]. The work presented in this paper, extends our previous work by proposing a formalism to capture CAD among process elements.

The issue of reflecting process evolutions and changes in WWS is possibility of making errors and inconsistencies, in implemented systems. The reasons for this are twofold. First is the flexibility of the implementation of WWS. Second is the understating and cohesive capturing CAD among process elements.

There are various process modeling tools such as Petri-nets [6], UML activity diagrams [7] and Event-driven Process Chains (EPC) [8]. However, most of these modeling tools are geared to capture process for automation. Thus fails to capture some of the CAD among process elements such as actions and data object [9, 10]. Apart form Petri-nets, other modeling tools are not equipped to model constraints adequately. Therefore, main purpose of this work is to find constraint modeling formalism, which facilitates encapsulating CAD among process elements. Most importantly, such mechanism should capture CAD among process elements, in a complete and cohesive manner. Further, the possibility of presenting the process and associated CAD into a single expression is preferred. Then this expression can be analyzed to understand the impact of one element change to the rest of the process.

The rest of the paper is organized as follows. In section 2, we briefly recapture the CAD model of process elements previously presented in [1]. In section 3, formalisms are explored to find a suitable constraint modeling language, which leads to the introduction Kleene Algebra (KA) and Kleene Algebra with Tests (KAT) [2]. Section 4 introduces a series of definitions based on KAT, to model CAD among process elements. In section 5, the use of of KAT expressions to find impact of process evolutions is discussed. The section 6 identifies the significance of the work compared to previous similar work. In conclusion, section 7 highlights possible future research directions, arising from this work.

2 CAD Among Process Elements

The CAD among process elements have two facets; i) types of associations and ii) constraints and conditions that affect the associations [1]. These are summarized below and drawn upon from previous landmark research such as [11-19].

Types of Associations among Process elements are; i) among actions (A,A) – *sequential, parallel, conditional split, simple merge, multi merge, compensation and skip*. ii) between actions and participants (A,P) *–obligation, permission and forbiddance.* iii) between actions and data object (A,D) *–visibility, interaction and routing.* iv) among data object elements (D,D) – *integrity and computational relationships.*

Constraints and conditions imposed on the associations among process elements are; i) on the associations among actions C(A,A) *–work item failures, deadline expiry, external trigger, constraints violation, time constraints and external events.* ii) on the associations between actions and participants C(A,P)- *location, experience, expertise or skills, availability, workload and level of the organization structure, same or different sub structure, interpersonal relationships and reporting or delegation authorities and history.* and iii) on the associations between actions and data object C(A,D) - *personalization or ownership of information and specialization or extensions.*

3 Kleene Algebra with Tests (KAT)

As mentioned previously the requirement is to find a formalism that allows capturing, CAD among elements into a single and analytical expression. Therefore, below we have analyzed to suitability of some of the prominent process modeling formalisms.

There are various formal methods used for process modeling purposes, for example Process Algebra [20, 21] and Petri-Nets[22, 6]. While, use of Petri-nets for process modeling is advocated, the inability to link to or refer to process data is considered to be one of the disadvantages of Petri nets [9]. In addition, Petri-nets alone do not provide an algebraic structure that allows capturing CAD among process elements into a single and analytical expression. The combination of Petri Nets with Process Algebra, named to be Petri Nets Calculus (or Petri Box Calculus – PBC) attempts to bring together the advantages both of Petri Nets and Process Algebra [23, 24]. However, the PBC does not provide a mechanism to capture the negation of a constraint, which is applicable with the constraints identified above. On the other hand, the use of *Kleene Algebra (KA)* and *Kleene Algebra with Tests (KAT)* [2] is promoted for constraint based program modeling [25] and other application modeling purposes [26]. Thus, here we experiment the use of KAT to model CAD among process elements.

The axioms KA and KAT as they are presented below. KA is an algebraic structure (K, +, · , * , 0 , 1) that satisfies the following axioms [2];

- + and · operators are associative → **a + (b + c) = (a + b) + c** and **a(bc) = (ab)c** for all **a, b, c in K**
- + is commutative → **a + b = b + a** for all **a, b in K**
- + and · are distributive → **a(b + c) = (ab) + (ac)** and **(b + c)a = (ba) + (ca)** for all **a, b, c in K**
- for + and · there exists an element **0** in **K** such that for all **a** in **K: a + 0 = 0 + a = a** and **a0 = 0a = 0**
- for + and · there exists an element **1** in **K** such that for all a in **K: a1 = 1a = a**
- for * there exists an elements **1** and **a** in **K** such **1+aa* = a** and **1+ a*a = a**. In other words * behaves like the Kleene Star operator in formal language theory.

KAT is a two-sorted algebraic structure (B, K, +, · , * , 0 , 1, ¬), where B is a subset in K and ¬ is a unary operator, similar to negation, defined only on B such that (K, +, · , * , 0 , 1) is a Kleene algebra and (B, +, · , ¬, 0 , 1) is a Boolean algebra [2].

Process actions are an integral part of a process. There are special characteristics associated with process actions. For example, actions cannot be negative as compared to conditions, in which the negation is valid. Also there is the possibility of certain actions to be repeated [14]. The Kleene star operator * allows this iteration to be modeled in combination with a guard condition to control the merging of parallel branches. In KA dot operator is not commutative (only + is), thus could be used to support two actions (say a_1 and a_2) in sequence. This is written as a_1a_2 since $a_1a_2 \neq a_2a_1$ according to KA axioms. Usually 'dot' operator is omitted in KAT expressions.

Looking at these characteristics, we define the following to present actions.

Definition 1: Process actions A $\{a_1, a_2, \ldots, a_x\}$ is a subset of K with an algebraic structure (K, A, +, · , * , 0 , 1) that satisfy the axioms of KA (defined above).

Usually a constraint is represented as a condition [26]. In conditions, it should be possible to represent the negation. Thus the subset B of K, which is a defined to be a Boolean algebra with the algebraic constructs (B, +, · , ¬, 0 , 1) can be utilized to define these different constraints on the association among process elements. Therefore, the definition below presents a mechanism to indicate constraints;

Definition 2: Constraints and conditions C $\{c_1, c_2, \ldots, c_y\}$ is a subset of B with an algebraic structure (B, C, +, · , ¬, 0 , 1) that satisfy the axioms of KA (defined before).

For this point onwards to denote actions and constraints, we use notations introduced in definitions 1 and 2.

Let us consider an example to demonstrate the use of KAT in representing process flow and various associations among process elements. Consider a typical *course creation* example in a university environment, in which a *course* is proposed to be offered (refer figure 1). In such a process, various academics would need to get involved in filling in various types of data. This data would include *basic course information*, *information about subjects offered in this particular course*, *business case details* (to assess the financial viability of the course) and *marketing information* (in order to publicize the course among potential students). In addition, various types of checking, assessing and approval would be required. The figure 1 gives an illustration of the process described above.

We have deliberately not used a specific standard modeling technique such as UML, Petri nets or EPC to represent the example process in figure 1. The rationale for this is to demonstrate the applicability of the KAT expressions to any process irrespective of the modeling technique used.

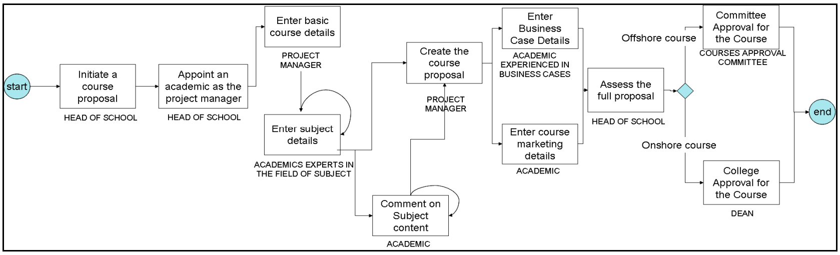


Fig. 1. Example of a course creations and approval process.

The notations are simple and denote the following; *boxes* represent *actions*, the *role name* below the box shows participants who perform the action. Arrow and diamond shape respectively denote process flow and condition.

Actions related to figure 1, are defined in table 1 according to our Definition 1.

Table 1. Actions identified according to example in figure 2.

a_0 → Initiate a course proposal	a_5 → Create the full course proposal
a_1 → Appoint an academic as the project manager	a_6 → Enter business case details
a_2→ Enter basic course details	a_7 → Enter course marketing details
a_3 → Enter Subject details	a_8 → Assess the full course proposal
a_4 → Comment on Subject content	a_9 → Committee Approval for the course
	a_{10} → College Approval for the Course

First, we show the method to embed elements from K and B in a single expression. The definition below represents performing an action under a guard condition.

Definition 3: If $\mathbf{a_x}$ represents an action and $\mathbf{c_m}$ denotes a constraint the expression $\mathbf{c_m a_x}$ indicates that a_x can be performed iff the c_m constraint is satisfied.

Next, more definitions are built on the foundation of Definitions 1 to 3 above.

4 KAT to Capture CAD among Process Elements

The definitions 4-7 below, demonstrate mechanisms to capture, various types of associations that exist among process actions, identified as (A, A) in section 2.

The definition below represents the sequential representation between two actions

Definition 4: The sequential association between two action a_x follows a_y is represented as $\mathbf{a_x a_y}$; denoting that action a_y can take place after a_x is completed.

The following definition shows the conditional split between two actions.

Definition 5: The conditional choice between two-action a_x or a_y is represented as $\mathbf{a_x + a_y}$. This denotes either action a_x OR a_x can be performed

The parallelism between two actions and multi merge is as follows.

Definition 6: The parallelism between two action a_x follows a_y is represented as $\mathbf{(c_m (a_x + a_y))^*}$. In this c_m presents the merging condition.

With the special element 1 ϵ K skipping of a certain action is denoted as follows;

Definition 7: The construct to denote that action a_x can be performed optionally is written as; $\mathbf{(a_x + 1)}$. This means that either action a_x can be performed or skipped as required.

Based on the definitions 4 to 7 we will write the basic process model in KAT of the above example in figure 1 (KAT expression 1). Note in this, the composite constraints are denoted using C_1 to C_{14}. These guard conditions are progressively expanded according to a new definitions introduced below.

$$(C_1\mathbf{a}_0)\ (C_2\mathbf{a}_1)(C_3\mathbf{a}_2\ )\ C_4(C_5\mathbf{a}_3)^*(C_6(C_7\mathbf{a}_4)^*+1)\ (C_8\mathbf{a}_5)\ (C_9(C_{10}\mathbf{a}_6 + C_{11}\mathbf{a}_7))\ ^*\ (C_{12}\mathbf{a}_8)\ (C_{13}\mathbf{a}_9 + C_{14}\mathbf{a}_{10})$$

KAT Expression 1

Due to space limitations the rest of the types of associations among process elements and constraints on these associations are presented in the definitions 8-20, in the table 2 below. The table also exemplifies the usage of this definition in relation to the example given in figure 1.

The notations c_1 to c_{32} captures CAD among process elements in relation to the example in figure 1. In table 3, we demonstrate how these notations are used to create composite conditions C_1 to C_{14}, which were identified in the KAT Expression 1.

Table 2. More KAT definitions for CAD of process elements and examples from figure 1.

Type of CAD	KAT based Definitions	CAD among process elements identified in relation to example in figure 1
(A, P)	**Definition 8:** $c_n \rightarrow$ P do (role, action) **Definition 9:** $c_n \rightarrow$ O do (role, action) **Definition 10:** $c_n \rightarrow$ F do (role, action)	**c_1** $\rightarrow$ P do (head of school, a_0) **c_2** $\rightarrow$ O do (head of school, a_1) **c_3** $\rightarrow$ O do (project manager, a_2) **c_4** $\rightarrow$ O do (expert academic, a_3) **c_5** $\rightarrow$ P do (academic, a_4) **c_6** $\rightarrow$ O do (project manager, a_5) **c_7** $\rightarrow$ O do (business case expert, a_6) **c_8** $\rightarrow$ O do (academic, a_7) **c_9** $\rightarrow$ O do (head of school, a_8) **c_{10}** $\rightarrow$ P do (dean, a_9) **c_{11}** $\rightarrow$ P do (course approval committee, a_9)
(A, D)	**Definition 11:** $c_n \rightarrow$ V ((**data** => {name =>(reference), location => (database.table.attribute | folder.document | database.table | class.object), display=> (view| edit| hidden), format => (textf | texta| selection |check box |radio | label | report| default)}), **action**) **Definition 12:** $c_m \rightarrow$I ((**data** {reference1, reference2, …}), **action x**, **action y**) **Definition 13:** $c_m \rightarrow$ RC (variable =>{reference}, operator =>{eq| gt| lt| el| eg}, value=> {any| 1|0| true| false| yes | no})	**c_{12}**$\rightarrow$V((data=>{name=>{basic course details}, location => {course_approval.basic_course}, display=> {view}, format=>{report }, name=>{ responsibility }, location=> {course_approval.staff_detail}, display=> {edit}, format =>{selection}), a_1) *Similar to above constraint notation c_{12}, assume that there are constraints c_{13} to c_{2}, showing the visibility constraints for actions a_2 to a_9 respectively.* **c_{22}**$\rightarrow$ I((data{number of subjects, subject names=>{}, key area=>{}), a_2 , a_3) **c_{23}** $\rightarrow$ I ((data{first offering year}), a_1 , a_2) **c_{24}** $\rightarrow$ RC (condition (variable =>{offshore }, operator =>{eq}, value=> {1})) **c_{25}** $\rightarrow$ RC (condition (variable =>{ number of loops }, operator =>{eq}, value=> {number of subjects}))
(D, D)	**Definition 14:** $c_m \rightarrow$ IC (subject component => {reference}, related components => { reference1, reference2 , ...}, connector => {reference, reference, ..}, type of link => {foreign key | hyperlink | document link }) **Definition 15:** $c_m \rightarrow$ CC (subject component => {reference}, related components => { reference1, reference2 , ...}, type of computation=> {summation | average | other formula })	**c_{26}**$\rightarrow$ IC (subject component => { basic course details }, related components => { subject information }, connector => {course_name, course_code}, type of link => { foreign key }) **c_{27}**$\rightarrow$ CC (subject component => { basic course information .number of subjects }, related components => { subject information }, type of computation=> {summation of number of subjects}) **c_{28}**$\rightarrow$ CC (subject component => { business case. Total cost }, related components => { staff costs, overheads, other costs, tax, levy }, type of computation=> {summation })
C(A,A)	**Definition 16:** $c_m \rightarrow$ TC (time reference => {absolute | relative}, start time => {seconds: minutes: hour: day: month: year| relative time}, end time => { seconds: minutes: hour: day: month: year | time period}) **Definition 17:** $c_m \rightarrow$ XC (entity => {reference}, trigger => {reference})	*This example does not contain any constraints under these definitions*
C(A,P)	**Definition 18:** $c_m \rightarrow$ PC (role, characteristic (factor => {location| experience| expertise| skills| availability| workload| level}, operator => {eq| gt| lt| el| eg | in | not in | yes | no | is | is not}, value =>{ figure})) **Definition 19:** $c_m \rightarrow$ PC (role 1, role 2, (characteristic (factor => {department| organizational unit| level | personal relationships| delegate | report}, correlation => {same | different | yes | no | higher | lower}))	**c_{29}**$\rightarrow$PC(academic, characteristic (factor => {skill}, operator=> {in}, value => {subject_details.key_area})) **c_{30}**$\rightarrow$PC(academic, characteristic (factor => { experience }, operator=>{in},value=>{ business case })) **c_{31}** $\rightarrow$ PC (academic, head of school, (characteristic (factor => {organizational unit }, correlation => {same})) **c_{32}** $\rightarrow$ PC (academic, dean, (characteristic (factor => { organizational unit }, correlation => {same}))
C(A,D)	**Definition 20:** $c_m \rightarrow$ OC (characteristic => {ownership | specialization}, (identification => (object => {attribute reference}, specific value => {identification of the individual | specialization details })))	*This example does not contain any constraints under these definitions*

Table 3. Guard conditions of KAT Expression 1 according to the definitions given in Table 2.

Composite guard element	Description According to the Dependencies Related to Example in Figure 2.
$C_1 = c_1$	Head of school is *permitted* to perform the subsequent action
$C_2 = c_2c_{12}$	Head of school is *obliged* to perform the subsequent action AND use the action interface definition referred by c_{12}
$C_3 = (c_{23})(c_3c_{31}c_{13})$	Academic appointed as the project manager in the same school as the head of school is *obliged* perform the subsequent action AND use the interface definition defined by c_{13}
$C_4 = c_{22}c_{25}$	A looping condition-based on the number_of_subjects captured in a_2, the subsequent action requires to be looped until data are filled for all subjects
$C_5 = (c_4c_{31})(c_{29})(c_{26}c_{14})$	An academic in the same school as head of school and has got the special expertise in the subject area can perform the subsequent action, using the interface definition in c_{14}
$C_6 = c_{22}c_{25}$	A looping condition - based on the number_of_subjects captured in a_2, the subsequent action requires to be looped until data are filled for all subjects
$C_7 = (c_5c_{31})(c_{15})$	An academic who is in the same school as head of school and has the special expertise in the subject area can perform the subsequent action using the interface definition c_{15}.
$C_8 = (c_6c_{31})(c_{26}c_{16})$	An academic appointed as project manager AND in the same school as head of school is allowed to perform the subsequent action, using the action interface definition c_{16}
C_9	Internal merge condition that looks whether both parallel actions are completed
$C_{10} = (c_7c_{31})(c_{30})(c_{26}c_{29}c_{17})$	An academic who is in the same school as the head of school AND has the expertise in the area of business cases is allowed to perform the subsequent action using data set defined by c_{27} and c_{28} using the interface definition c_{17}
$C_{11} = (c_8c_{31})(c_{26}c_{18})$	An academic who is in the same school as the head of school allowed to perform the subsequent action using data set defined by c_{26} AND using the interface definition c_{18}
$C_{12} = c_9\,c_{19}$	Head of school is obliged to perform the action using the interface definition c_{19}
$C_{13} = (\neg c_{24}(c_{10}c_{32}))(c_{21})$	If the course is NOT off-shore a dean is in the same college as head of school is allowed to perform the subsequent action using the interface definition c_{21}
$C_{14} = (c_{24}(c_{10})(c_{21}))$	If the course is offshore the chair of the courses approval committee is allowed to perform the subsequent action using the action definition c_{20}

Thus, we can write the complete expanded version of KAT expression as follows;

$$(c_1a_0)(c_2c_{12}a_1)((c_{23})(c_3c_{31}c_{13})a_2)(c_{22}c_{25})(((c_4c_{31})(c_{29})(c_{26}c_{14}))a_3)^*((c_{22}c_{25})(((c_5c_{31})(c_{15}))a_4)^*+1)(((c_6c_{31})(c_{26}c_{16}))a_5)(C_9(((c_7c_{31})(c_{30})(c_{26}c_{29}c_{17}))a_6+((c_8c_{31})(c_{26}c_{18}))a_7))^*((c_9c_{19})a_8)(((\neg c_{24}(c_{10}c_{32}))(c_{21}))a_9+((c_{24}(c_{10})(c_{21})))a_{10})$$

KAT Expression 2

The KAT expression 2 represents a complete and cohesive set of dependencies identified related to process in figure 1. However, the usage of expression on paper for human is limited. In other words, it requires an data structure to capture above expressions. Such a representation should capture all the semantics involved in the structure of the expression and should be analytical. Space limitations in this paper do not provide us opportunity to explore into such a construct.

5 Using Kat Expression to Locate the Impact of Evolution

Though it is not practical to analyze KAT expression 2 manually, here we conceptually demonstrate how it is done. Consider the change,*" It is no more required to capture the year of first offering in the course documents"* due other business needs.

Now it let us analyze impact of this change on the process. By analyzing the list of constraint definitions in table 3, we find that this data field is related to c_{23}. Using this as the starting point, below we present four types of impacts that could have on the rest of the process;

- **Direct impact** – Actions that are directly affected or cannot be executed because of the suggested change – highlighted in blue in the KAT expression 3.
- **Indirect impact** – Actions that that has cannot be researched because of the direct impact – highlighted in yellow in KAT expression 3 (since action a_2 is prior to other actions in a sequential ordering of actions)
- **Secondary impact** – Actions that may be performed but unable to merge with the main flow due to the direct impact – These kind of impact are not present due this particular change
- **Cautionary impact** – Actions that can be performed but requires checking to assure the accuracy – highlighted in green in the KAT expression 3.

$$(c_1a_0)\ (c_2c_{12}a_1)\ ((c_{23})(c_3c_{31}c_{13})a_2\ )\ (c_{22}c_{25})(((c_4c_{31})\ (c_{29})\ (c_{26}\,c_{14}))a_3)^*((c_{22}c_{25})((\ (c_5c_{31})\ (c_{15}))\ a_4)^*+1)\ (((c_6c_{31})\ (c_{26}c_{16}))\ a_5)\ (C_9(((c_7c_{31})\ (c_{30})\ (c_{26}c_{29}c_{17})\ )a_6+((c_8c_{31})\ (c_{26}c_{18}))a_7))\ ^*\ ((c_9\,c_{19})a_8)\ (((\neg c_{24}\ (c_{10}\,c_{32}\ ))(c_{21}))a_9+((c_{24}(c_{10})\ (c_{21})))a_{10})$$

KAT Expression 3

6 Similar Work

The similarity or difference of previous work against this research is hard to measure in one dimension. The main objective of this work is to support web-based workflow evolution, by means of capturing process element CAD among process elements into a single formal expression. The highlighted words are the key areas that this research is associated.

There are number of research works, such as [17, 18, 27-29], that aim to support process *evolution*. In these researches, the term evolution is used synonym to dynamism, flexibility, adaptability, etc. In addition, the approaches used for making workflows evolvable are different in each of these works. In particular, the works [17, 28] approaches are somewhat similar to our work, in which the constraints and dependencies among process elements are considered in supporting process evolution.

There are various researches aimed at capturing process elements associations, constraints and dependencies such as [12-19]. However in most of this works the objective for capturing process constraints is different from our main goal of supporting web based workflow evolution. Therefore, the identification of various constraints, associations and dependencies among process elements are presented in varying degrees in these researches. Nevertheless, Sadiq et al's work on specification and validation of process constraints for flexible workflows, bares certain similarities to our work, in identifying CAD among process elements [17]. However, the significance of our work is using KAT for cohesive capturing of CAD among process elements.

7 Conclusion and Future Research Directions

This paper aims to support the issue of managing the process evolution in web-based workflows. In particular, here we introduce the importance of understanding various CAD among process elements, in order to locate the impact of one change to the rest of the process.

The approach used here is to represent CAD among process elements using a set of KAT based definitions. The KAT based definitions allows capturing CAD among process elements into a single expression. Though a practical application was not presented in relation to impact resolution, the method of locating impact of one change to the rest of the process using KAT expressions was demonstrated in concept.

As the future research of this work, the foremost important one would be to implement a system that allows creating and analyzing KAT expressions to find the process element changes to the rest of the workflow. Secondly, it would be advantageous to validate the exhaustiveness of the dependencies identified in section 2, based on process examples of other domains.

References

1. J. A. Ginige, A. Ginige, and U. Sirinivasan, "Towards Effective Management of Process Evolution in Web-based Workflows: Dependencies and Constraints Model of Process Elements," Melbourne Australia, 2007 (Accepted).
2. D. Kozen, "Kleene Algebra with Tests," ACM, pp. 17, 1999.
3. WfMC, "Workflow Management Coalition - Terminology & Glossary." Winchester, Hampshire, United Kingdom: Workflow Management Coalition, Feb 1999, pp. 8-9.
4. J. A. Ginige and A. Ginige, "Meta-Model for Tracing Impact of Contextual Information Evolution in Web-based Workflows," presented at 9th International Conference on Enterprise Information Systems (ICEIS 07), Funchal, Madeira - Portugal, 2007.
5. W. M. P. van der Aalst, "Verification of Workflow Nets," Application and Theory of Petri Nets, vol. 1248, pp. 407–426, 1997.
6. W. M. P. van der Aalst, K. M. van Hee, and G. J. Houben, "Modelling and analysing workflow using a Petri-net based approach," presented at Proc. 2 ndWorkshop on Computer-Supported Cooperative Work, Petri nets and related formalisms, 1994.
7. J. W. Schmidt, "A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes," D. I. A. Wienberg, Ed. Harburg: Technische Universität Hamburg-Harburg, 2001.
8. J. Mendling, G. Neumann, and M. Nuttgens, "Towards Workflow Pattern Support of Event-Driven Process Chains (EPC)," presented at 2nd Workshop XML4BPM, 2005.
9. E. Oren and A. Haller, "Formal frameworks for workflow modelling," DERI-Digital Enterprise Research Institute, vol. 20, 2005.

10. W. M. P. van der Aalst, M. Weskez, and G. Wirtz, "Advanced Topics in Workflow Management: Issues, Requirements, and Solutions," Integrated Design and Process Science, vol. Volume 7, pp. pp 49-77, 2003.
11. R. Lu, S. Sadiq, V. Padmanabhan, and G. Governatori, "Using a temporal constraint network for business process execution," Proceedings of the 17th Australasian Database Conference-Volume 49, pp. 157-166, 2006.
12. O. Marjanovic, "Dynamic Verification of Temporal Constraints in Production Workflows," Proceedings of the 11th Australian Database Conference, vol. 7481, 2000.
13. N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Workflow data patterns," presented at 24th Int. Conf. on Conceptual Modeling (ER05), 2005.
14. N. Russell, A. H. M. ter Hofstede, W. M. P. van der Aalst, and N. Mulyar, "WORKFLOW CONTROL-FLOW PATTERNS A Revised View," BPMcenter.org 2006.
15. N. Russell, W. M. P. van der Aalst, and A. H. M. ter Hofstede, "Workflow Exception Patterns," 2006.
16. N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, "Workflow Resource Patterns: Identification, Representation and Tool Support," presented at 17th Conference on Advanced Information Systems Engineering (CAiSE'05), 2005.
17. S. W. Sadiq, M. E. Orlowska, and W. Sadiq, "Specification and validation of process constraints for flexible workflows," Information Systems, vol. 30, pp. 349-378, 2005.
18. W. Sadiq, O. Marjanovic, and M. E. Orlowska, "Managing Change and Time in Dynamic Workflow Processes," IJCIS, vol. 9, pp. 93-116, 2000.
19. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros, "Workflow Patterns," Distributed and Parallel Databases, vol. 14, pp. 5-51, 2003.
20. J. Camara, C. Canal, J. Cubo, and A. Vallecillo, "Formalizing WSBPEL Business Processes Using Process Algebra," Proceedings of FOCLASA, 2005.
21. H. Smith and P. Fingar, "Workflow is just a Pi process," Computer Sciences Corporation 2003.
22. W. M. P. van der Aalst, K. M. Hee, and K. V. Hee, Workflow Management: Models, Methods, and Systems: Mit Pr, 2002.
23. J. Hall, The High Level Petri Box Calculus: Basic Concepts: University of York, Dept. of Computer Science, 1993.
24. B. Mikolajczak, "Review of Petri Net Algebra," ACM SIGACT News, vol. 33, pp. 10-14, 2002.
25. D. Kozen, "Kleene algebra with tests, Transactions on Programming Languages andSystems " ACM, pp. 427–443, 1997.
26. K. D. Schewe and B. Thalheim, "Conceptual modelling of web information systems," Data and Knowledge Engineering, vol. 54, pp. 147–188, 2005.
27. G. Vossen, M. Weske, and G. Wittkowski, "Dynamic Workflow Management on the Web," Fachbericht Angewandte Mathematik und Informatik, vol. 24, pp. Last accessed on 10/03/2005 from http://dbms.uni-muenster.de/staff/weske/papers/fb24-96.ps.gz, 1996.
28. J. Wainer, F. Bezerra, and P. Barthelmess, "Tucupi: a flexible workflow system based on overridable constraints," presented at Proceedings of the 2004 ACM symposium on Applied computing, 2004.
29. D. Wang, Y. A. N. Xiu-Tian, W. J. Ion, W. Runxiao, et al., "An Approach to a Web-based Flexible Workflow Modeling," presented at ASR '2005 Seminar, Instruments and Control, Ostrava, April 29, 2005, 2005.

Clustering ERP Implementation Project Activities: A Foundation for Project Size Definition

Guy Janssens[1], Rob Kusters[1, 2] and Fred Heemstra[1, 3]

[1]Open Universiteit Nederland, School of Management, P.O. Box 2960, 6401 DL Heerlen The Netherlands, Phone +31-45 5762765
guy.janssens@ou.nl
[2]Eindhoven University of Technology, Department of Technology management P.O. Box 513, 5600 MB Eindhoven, The Netherlands, Phone +31-45 5762975
r.j.kusters@tm.tue.nl
[3]KWD Resultmanagement, P.O.Box 551, 3430 AN Nieuwegein, The Netherlands
f.heemstra@chello.nl

Abstract. The size of an ERP project can be a useful measurement for predicting the effort needed to complete an ERP implementation project. Because this measurement does not exist, research is needed to find a set of variables which can define the size of an ERP implementation project. This paper shows 21 logical clusters of ERP implementation project activities as a result of a formal group session. The clusters are based on 405 ERP implementation project activities retrieved from literature. These clusters can be used in further research to find variables for defining the size of an ERP implementation project.

1 Introduction

Globalization has put pressure on organizations to perform as efficient and effective as possible in order to compete in the market. ERP is a key ingredient for gaining competitive advantage, streamlining operations, and having “lean” manufacturing [1]. ERP projects are large and risky projects which affect large parts of the organization and lead to changes in the way the organization performs its tasks. The costs for implementation are usually very high and also very hard to estimate. Even cases exist where ERP implementation projects led to bankruptcy [2, 3]. Francalanci states that within the total cost of the implementation project, the software costs represent only a fraction of the overall cost of ERP projects, less than 10% over a 5-year period [4]. In addition Willis states that consultants alone can cost as much as or more than five times the cost of the software [5], this is confirmed by Von Arb who indicates that the consultancy costs can be 2 to 4 times the software license costs [6]. This indicates that the effort required for implementation of an ERP system consists mostly of effort costs. Von Arb also argues that the license and hardware costs are fairly constant and predictable and that only a focus on a reduction of effort costs is realistic. The conclusion is legitimate that the total effort is the most important and difficult factor to estimate in an ERP implementation project. Therefore the main research of the authors focuses only on the estimation of the total effort needed for implementing an ERP system.

This paper takes a first step in this research by answering which activities exist in ERP projects according to literature and how these can be clustered as a basis for defining the size of an ERP project. It will start with explaining the approach and goal of the research, followed by a literature review on ERP project activities. After that it will present the clustering approach and results followed by conclusions and discussion.

2 Research Approach

When examining more or less successful methods for predicting software development effort, it is to be expected that in the area of implementing ERP systems, measurements can also be found for predicting implementation efforts.

However, Stensrud [7] already indicated that although many effort prediction systems exist, none unfortunately have been specifically devised for ERP projects. Heemstra and Kusters [8] collected candidate cost driver variables from literature and asked experts in two major companies for their opinion about the relevance of these variables. One of their conclusions was that the size of an ERP implementation is a major cost driver in ERP implementation projects. In software development the size of the software can be expressed in a single variable such as number of program lines or function points [7]. By using this variable in a formula with several parameters, useful predictions of the development effort can be made. Can similar variables be found for predicting the implementation effort in an ERP project? According to Stensrud several variables together should be used to express this size. Francalanci [4] used three variables for her size definition: organizational size, configuration size and technical size. Von Arb [6] used two variables for size definition in his dissertation: number of users and number of ERP (sub)modules. As far as the authors can conclude from studying available publications on this topic, no further research has been done in defining the size of an ERP implementation project. All the mentioned researchers concluded that size cannot be expressed as a single variable as in software development, but should be expressed as a multidimensional variable. ERP implementation projects are complex projects where successful organizational, technical and people strategies are critical factors for success [9]. Because an ERP implementation project is confronted with many different aspects, the authors postulate the hypothesis that an ERP implementation project consists of a collection of clusters of activities with their own focus on implementation costs and project size. Clusters of activities include: the preparation of the appropriate technical infrastructure, the business process redesign or the installation of the software. Of course these clusters of activities will be related to each other, but the authors expect them to influence the total cost of the implementation project fairly independently. If size variables can be found for these clusters and these variables could be used as an estimator for the prediction of the effort needed for these clusters, these variables could be the dimensions of the multidimensional variable which defines the size of an ERP implementation project.

In order to define these clusters, the activities in an ERP implementation project must be known. In methodologies for (regular) information systems development, all relevant activities are described and defined in terms of goals, results and necessary resources. During planning, activities relevant in a specific situation will be selected from this methodology. It goes without saying that not all activities are relevant in every project. There is no reason to expect that an ERP implementation project will be different in that matter. Therefore this research is based on the assumption that a range of activities exists which represents the most relevant activities in an ERP project. The relevant ERP implementation activities were retrieved from published research. Although several authors showed the phases in an ERP project and activities within [10], a complete list of all relevant activities in an ERP implementation project was not found unfortunately. Therefore papers were collected which listed activities within an ERP implementation project. By examining papers with different views the authors of this paper expect to have found the most relevant activities.

This paper tries to lay a foundation for the definition of the size of an ERP project. Because it is expected that the costs for effort will constitute the greatest part of the total cost of an ERP implementation project, the first logical step is to define which activities that require human effort are important in an ERP project. Activities are always performed for a reason, i.e. to reach a certain goal and can be grouped into logical clusters which contribute to the same intermediary product or products. For instance, an intermediary product such as ‘trained users’ can be achieved by a cluster of activities like: ‘prepare training material’, ‘train the trainers’, ‘set up training infrastructure’, ‘train users’ etcetera.

3 Objective of this Research

The objective of this research is to define logical clusters of ERP project activities.

This paper will show the method and results in retrieving important ERP activities and the results of this first formal attempt to cluster these activities into clusters which contribute to similar intermediate products. This paper aims at answering the next research questions: Which activities in general exist in ERP projects according to literature? What is a useful method to cluster these activities? What is the result of a first clustering of these activities?

4 Literature Review on ERP Project Activities

A literature search was performed aiming at finding papers in which activities within an ERP implementation project were listed. From these papers a collection of names and expressions of activities was retrieved.

A paper was selected if it showed at least one list of activities performed in ERP selection, implementation or maintenance. A total of 23 papers were found with lists of ERP activities. These papers can be divided into three categories:

1. Papers which relate risk factors and Critical Success Factors (CSF’s) to activities and/or project phases.

2. Papers about cases which describe the phases and activities of the actual projects.
3. Papers which describe standard project phases and activities from consultancy firms or ERP software suppliers.

It can be expected that these three types of papers will show the important project activities.

The next section will discuss the retrieved papers grouped by the three categories.

Although the authors aimed at activities that are part of the implementation project, activities were also recorded in this literature research that belong to the pre-implementation phase and maintenance phase of an ERP system.

4.1 Papers with Research based Phases and Activities

These research studies relate risk factors, critical success factors or other influencing factors to activities and/or project phases. These authors based their framework of the standard activities and project phases on other scientific research and in some cases performed interviews with experts to enhance their framework.

A first example of this type of research is by Parr and Shanks [11]. The purpose of their research was to create a project phase model (PPM) of ERP project implementation. They based their model on other process models of ERP implementation from other researchers and tried to synthesize these models into one model which also recognizes the importance of the planning and post-implementation stages. They used the model in 2 case studies to examine the relationship between the CSF's from their earlier research and the phases to the PPM. Rajogopal [12] used a stage model to analyse six manufacturing firms that had one of the widely used ERP systems to retrieve factors of influence in the various stages of ERP implementation. He based his stage model on a six-stage model from Kwon and Xmud and other authors. Al-Mashari et al. [13] presented a novel taxonomy of the critical success factors in the ERP implementation process. They based their taxonomy on a comprehensive analysis of ERP literature combining research studies and organisational experiences. In their taxonomy they showed three major ERP phases. In these phases they also described project activities based on an analysis of ERP literature.

Ehie and Madsen [14] studied 38 critical issues in ERP implementation to measure the critical factors of ERP implementation. They developed a questionnaire based on five stages of ERP implementation. Stages are based on reviews of literature and extensive personal interviews with ERP consultants. In their investigation on critical management issues in ERP implementation Kumar et al. [15] divided the project activities into 2 phases 'dollars to assets' and 'assets to impacts'. They described the typical activities within these phases. They based their phase and activities on innovation process stage models from other authors. They used these activities in open-ended questions in a questionnaire for ERP project managers of 20 Canadian organizations. The aim of the questionnaire was to find critical management issues. Hallikainen et al. [16] developed and tested a model to support the decision which modules are implemented and in which order. They based their model on the phase model of Bancroft. In their paper in which they seek to provide a conceptual model that explains the complexity of an ERP system to project managers in a non-technical manner, Marnewick and Labuschagne [17] also present an ERP implementation methodology, which consists of 5 steps. Somers and Nelson [18] examined the ERP project from different viewpoints: Players, ERP Project Life Cycle Stages and Activities. Their main purpose was to analyze the importance of key players and activities across the ERP life cycle by designing a questionnaire which 116 companies returned. They adopted the six-stage model from Rajagopal [12]. For every phase they derived the key activities from other research studies. The same six-stage model was used by Somers and Nelson [19]. They questioned 86 organizations for retrieving the impact of Critical Success Factors (CSF's) across the stages of ERP implementations. The top CSF's listed for every ERP implementation stage, largely consist of project activities. Umble et al. [20] identified CSF's, software selection steps and implementation procedures critical to a successful implementation. Based on available resources and own experiences, including a case study they showed the most important activities for ERP system selection and implementation steps. The activities for selecting an ERP system were presented by Wei and Wang [21]. They constructed a comprehensive framework for selecting an ERP system and applied it to a case in Taiwan. Followed by a research paper in which they presented a comprehensive framework for selecting a suitable ERP system, based on the analytic hierarchy process (AHP) method from Saaty [22]. Wagner and Antonucci [23] studied whether there are different ERP implementation approaches and models for a large-scale integrated ERP system in the public sector as compared to the private sector. For their research they used a generalized structured implementation. Markus and Tanis [24] described various subjects of ERP systems for educational purposes. They based their phases on other models from other authors. For every phase they described typical activities, common errors or problems, typical performance metrics and possible outcomes. Latvanen and Ruusunen [25] used a socio-technical model of risk management of ERP projects. Mabert et al. [26] compared and evaluated the use of regression analysis, logistic (logit) models, discriminate analysis and data envelopment analysis (DEA), for empirical data from ad survey of ERP implementations in the US manufacturing sector. For this they used key planning, decision and implementation management variables for the implementation phases. They did not specify important activities within these phases. Sumner [27] identified risk factors unique to ERP projects by interviewing ERP project managers in 7 companies. For this research she used five ERP project phases.

4.2 Papers with Case-based Phases and Activities

These research studies present case studies of ERP implementation projects. The purpose of these studies is to show in detail what happened in an actual case or to use a case to test a construct.

Berchet and Habchi [28] studied an ERP implementation project at Alcatel. The project was carried out according to a five-stage model. They also described important activities for every phase. In describing the ERP implementation at Rolls-Royce Yusuf et al. [29] carried out an in-depth study of the issues behind the process of implementation. The implementation plan at Rolls-Royces consisted of 4 main phases. In their description of these phases the main activities were also described. Sarker and Lee [30] tested three critical success factors in a case. They concluded that only the CSF 'strong and committed leadership' could be empirically established as a necessary condition. The case company implemented ERP by three phases.

4.3 Papers with Project Phases from Consultancy Firms and ERP Suppliers

One paper specifically described ERP implementation methodologies used by consultancy firms or ERP suppliers.

Bruges [31] showed the phases and main activities from three methodologies: AcceleratedSAP (ASAP), The Total Solution (Ernest & Young) and The Fast Track Workplan (Deloitte & Touche).

4.4 Retrieved Activities

From these three types of papers the list of activities was retrieved. Because the intention is to cluster these activities into logical units, no attention was paid to the phases mentioned in the papers. As shown above there is a variety of the numbers and names for project phases. Therefore only the activity names were retrieved.

In total 402 activities were recorded. Of course the same activity was mentioned more than once. Double names, synonyms or homonyms were not filtered out for reasons as discussed below in the metaplan session. These activities should be categorized unbiased. A filtering of the activities before the session would result in activities selected and named by the personal preference of the researchers.

5 Clustering Approach

A grouping technique was needed in order to be able to categorize the retrieved activities into logical clusters of activities. As mentioned before the selection and testing of the technique was also a research goal.

Except for its name and in most cases the project phase, no more properties of an activity were available. Therefore the clustering can only be done by human judgement. If this is done by one human individual, bias and limited knowledge will influence the result. However judgement by several individuals and group interaction will improve the quality of the results. Unfortunately members of freely interactive groups are often dissatisfied with group interaction [32]. The number of found activities (402) also implies the need for a formal technique. According to Howard a Nominal Group Technique (NGT) improves the output and satisfaction of the group members [32]. For this research a low-cost, fast and easy clustering technique was needed. Therefore the metaplan technique was chosen, which can be viewed as a Nominal Group Technique (NGT).

The metaplan technique was developed by Wolfgang and Eberhard Schnelle and is a simple visual technique which can be used by groups to structure thinking processes within the context of group work. A moderator leads the group discussion.

Ideas are generated by group members and noted on cards. Finally, these cards are organized into categories and may show new results of which the single persons were not aware.

This metaplan session was performed as a first step in categorizing i.e. clustering ERP activities in clusters which are logical groups of activities in an ERP implementation project which contribute to the production of the same intermediary products. Of course the activities found in the papers are not comprehensive. However, it is reasonable to expect that the activities mentioned in these papers are important activities in an ERP implementation project and will influence the total project effort. The goal of this first session was to find out whether activities can easily be clustered and if a technique like the metaplan technique can be used in future to improve the clustering by more experts.

The first step in a regular metaplan session is a brainstorming part from which ideas are generated and noted on cards. In this case there was no brainstorming session for retrieving possible ERP activities. This was replaced by retrieving activities from relevant scientific papers in which phases and activities within these phases were described. The list retrieved from these activities is probably more complete and relevant than by brainstorming. Of course there are many synonyms and homonyms, but this also will be the case in an actual brainstorming session. Only the categorizing part of the metaplan technique was used. The names of the 402 activities where printed on 402 post-it notes. Of these activities the following data were printed: name, project phase if present and title of the paper.

The metaplan session was performed by the authors of this paper in a 3-hour meeting. The session was prepared by the first author. The participants of this session were instructed to categorize these post-it notes into logical clusters by sticking them to a wall. The participants should categorize by bearing strongly in mind that clusters should not relate to project phases, but that activities within a cluster should strongly contribute to the same intermediate product or products of an ERP implementation. After assigning all relevant activities to a cluster, the clusters were studied by the group in detail, which resulted in some rearranging of activities and also in some subgroups within the main clusters. After this session the clusters and activities within the subgroups were recorded in a spreadsheet and obvious double activities and synonyms removed in a two-hour separate session by the first two of the authors. In this session the cluster names and logical sequence were also enhanced.

6 Results

From the outcomes of the session it can be concluded that the metaplan technique is a suitable technique for clustering ERP activities.

Preparing the session was a labour-intensive process. The session itself took about 3 hours, mainly caused by the large number of activities (402). The categorizing itself was not a difficult task. The method could also be useful in following research where more experts should perform the same exercise. Although for practical reasons it would be advisable to perform this session by applying a method and software to do the clustering independent from time and place. If experts could perform the clustering whenever they want and wherever they want, the willingness to participate will be higher. As also shown by Howard, support of this process by a Group Decision Support System (GDSS), which can support clustering in different locations and/or at different times, leads to the same quality of the results [32].

Table 1 shows the found clusters and subclusters. Table 1 also shows that 208 unique activities were assigned to the clusters and/of subclusters. In the second session the homonyms and synonyms from the 405 activities were removed, which led to 208 unique activities. The complete list of activities is available through the authors.

Table. 1. Found clusters and sub clusters.

Clusters	**Subclusters**	**Number of unique activities**
Selection	Vendor selection	4
	Product selection	16
Project configuration		19
Project management	Management	4
	Communication to organization	4
Organizational and system design	Current state analysis	5
	Organizational requirements	7
	Requirements ERP system	8
	High level Design	6
Configuration and installation	System configuration	17
	Data conversion	4
	System integration	9
	ERP system testing	14
Customizing		7
Infrastructure		14
Reorganization		11
System implementation		21
Training	Training Implementation Staff	2
	Training users	9
	Training maintenance staff	2
Set up maintenance		25
TOTAL		**208**

7 Conclusion and Discussion

The results of the research described in this paper are clusters of activities. It forms a basis for further research on this subject. Because the clustering has been done by a group of three authors, future research should increase this group and further verify these activities and clusters. Future research should also check these activities against activities retrieved from real life projects. There should be a check whether activities from real-life projects can be categorized according to the found clusters of activities. It should of course also be checked whether the activities that can be found in real-life project documentation occur in the list of activities from the literature search.

As described before, the metaplan technique is a suitable technique for clustering activities. The use of a Group Decision Support System (GDSS) can facilitate the use of this technique. The same exercise can easily be performed by other researchers.

The results of this paper will also be used to perform a first exploration into the practical use of the clusters for defining variables which could be used to define the size of an ERP implementation project. As discussed in the research approach, the size of an ERP implementation project should be expressed in a multidimensional variable. At this point in time the authors assume that the clusters can serve as the dimensions by which an ERP implementation project can be viewed.

The first impression of the authors is that the sub clusters and not the clusters should be the starting point for the definition of variables, because the level of detail of the clusters seems to be too low to be able to easily find variables. However, this has to be verified in further research. For the subclusters the most important objects (for instance: user, trainer etcetera) should be found, followed by variables by which these objects can be measured (for instance: number of users).

References

1. Mabert, V.A., A. Soni, and M.A. Venkataramanan, *Enterprise resource planning: Managing the implementation process.* European Journal of Operational Research, 2003. 146(2): p. 302-314.
2. Holland, C.R. and B. Light, *A critical success factors model for ERP implementation.* Software, IEEE, 1999. 16(3): p. 30-36.
3. Scott, J.E., *The FoxMeyer Drugs' Bankruptcy: Was it a Failure of ERP?* Americas Conference on Information Systems, August 13-15, Milwaukee,, 1999: p. 223-225.
4. Francalanci, C., *Predicting the implementation effort of ERP projects: empirical evidence on SAP/R3.* Journal of Information Technology, 2001. Volume 16(1): p. 33 - 48.
5. Willis, T.H., A.H. Willis-Brown, and A. McMillan, *Cost containment strategies for ERP system implementations.* Production and Inventory Management Journal, 2001. 42(2): p. 36.
6. Arb, R.v., *Vorgehensweisen und Erfahrungen bei der Einführung von Enterprise-Management-Systemen dargestellt am Beispiel von SAP R/3*, in *Information Engineering.* 1997, Institute für Wirtschaftsinformatik der Universität Bern: p. 426.
7. Stensrud, E., *Alternative approaches to effort prediction of ERP projects.* Information and Software Technology, 2001. 43(7): p. 413-423.
8. Heemstra, F.J. and R.J. Kusters, *Wat bepaalt de kosten van ERP-implementatie?* Management Accounting, 2005(July/August).
9. Aladwani, A.M., *Change management strategies for successful ERP implementation.* Business Process Management Journal, 2001. 7(3): p. 266-275.
10. Robey, D., J.W. Ross, and M.-C. Boudreau, *Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change.* Journal of Management Information Systems, 2002. 19(1): p. p17, 30p.
11. Parr, A. and G. Shanks, *A model of ERP project implementation.* Journal of Information Technology, 2000. 15(4): p. 289-303.
12. Rajagopal, P., *An innovation--diffusion view of implementation of enterprise resource planning (ERP) systems and development of a research model.* Information & Management, 2002. 40(2): p. 87-114.
13. Al-Mashari, M., A. Al-Mudimigh, and M. Zairi, *Enterprise resource planning: A taxonomy of critical factors.* European Journal of Operational Research, 2003. 146(2): p. 352-364.

14. Ehie, I.C. and M. Madsen, *Identifying critical issues in enterprise resource planning (ERP) implementation.* 2005. 56(6): p. 545-557.
15. Kumar, V., B. Maheshwari, and U. Kumar, *An investigation of critical management issues in ERP implementation: empirical evidence from Canadian organizations.* Technovation, 2003. 23(10): p. 793-807.
16. Hallikainen, P., H. Kimpimäki, and H. Kivijärvi, *Supporting the Module Sequencing Decision in the ERP Implementation Process.* Proceedings of the 39th Hawaii International Conference on System Sciences - 2006, 2006: p. 1-10.
17. Marnewick, C. and L. Labuschagne, *A conceptual model for enterprise resource planning (ERP).* Information Management & Computer Security, 2005. 13(2): p. 144-155.
18. Somers, T.M. and K.G. Nelson, *A taxonomy of players and activities across the ERP project life cycle.* Information & Management, 2004. 41(3): p. 257-278.
19. Somers, T.M. and K.G. Nelson, *Proceedings of the Annual Hawaii International Conference on System Sciences The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations*. 2001.
20. Umble, E.J., R.R. Haft, and M.M. Umble, *Enterprise resource planning: Implementation procedures and critical success factors.* European Journal of Operational Research, 2003. 146(2): p. 241-257.
21. Wei, C.-C. and M.-J.J. Wang, *A comprehensive framework for selecting an ERP system.* International Journal of Project Management, 2004. 22(2): p. 161-169.
22. Wei, C.-C., C.-F. Chien, and M.-J.J.M.-J.J. Wang, *An AHP-based approach to ERP system selection.* International Journal of Production Economics, 2005. In Press, Corrected Proof.
23. Wagner, W. and Y.L. Antonucci, *An analysis of the imagine PA public sector ERP project.* System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, 2004: p. 8.
24. Markus, M.L. and C. Tanis, *Chapter 10: The Enterprise System Experience—From Adoption to Success.* Framing the Domains of IT Management: Projecting the Future Through the Past, Pinnaflex Educational Resources, Inc., 2000: p. 173-207.
25. Latvanen, H. and R. Ruusunen, *Management of Risks in an ERP Implementation Project*, Turku School of Economics, 2001. p. 20.
26. Mabert, V.A., A. Soni, and M.A. Venkataramanan, *Model based interpretation of survey data: A case study of enterprise resource planning implementations.* 2005. In Press, Corrected Proof.
27. Sumner, M., *Risk factors in enterprise-wide/ERP projects.* Journal of Information Technology, 2000. 15(4).
28. Berchet, C. and G. Habchi, *The implementation and deployment of an ERP system: An industrial case study.* 2005. 56(6): p. 588-605.
29. Yusuf, Y., A. Gunasekaran, and M.S. Abthorpe, *Enterprise information systems project implementation: A case study of ERP in Rolls-Royce.* International Journal of Production Economics, 2004. 87(3): p. 251-266.
30. Sarker, S. and A.S. Lee, *Using a case study to test the role of three key social enablers in ERP implementation.* Information & Management, 2003. 40(8): p. 813-829.
31. Bruges, P., *ERP Implementation Methodologies.* MSIS, 2002. 488.
32. Howard, M.S., *Quality of Group Decision Support Systems: a comparison between GDSS and traditional group approaches for decision tasks*. 1994, Eindhoven University of Technology: Eindhoven. p. 219.

Authority and its Implementation in Enterprise Information Systems

Alexei Sharpanskykh

Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a
1081 HV Amsterdam, The Netherlands
sharp@few.vu.nl

Abstract. The concept of power is inherent in human organizations of any type. As power relations have important consequences for organizational viability and productivity, they should be explicitly represented in enterprise information systems (EISs). Although organization theory provides a rich and very diverse theoretical basis on organizational power, still most of the definitions for power-related concepts are too abstract, often vague and ambiguous to be directly implemented in EISs. To create a bridge between informal organization theories and automated EISs, this paper proposes a formal logic-based specification language for representing power- (in particular authority) relations. The use of the language is illustrated by considering authority structures of organizations of different types. Moreover, the paper demonstrates how the formalized authority relations can be integrated into an EIS.

1 Introduction

The concept of *power* is inherent in human organizations of any type. Power relations that exist in an organization have a significant impact on its viability and productivity. Although the notion of power is often discussed in the literature in social studies [1, 2, 4, 5, 6, 7, 12, 13], it is only rarely defined precisely. In particular, power-related terms (e.g., control, authority, influence) are often used interchangeably in this literature. Furthermore, the treatment of power in different streams of sociology differs significantly. One of the first definitions for power in the modern sociology was given by Max Weber [20]: *Power is the probability that a person can carry out his or her own will despite resistance*. Weber and his followers (Dahl, Polsby) considered power as an inherently coercive force that implied involuntary submission and ignored the relational aspect of power. Other sociologists (Bierstedt, Blau) considered power as a force or the ability to apply sanctions [2]. Such view was also criticized as restrictive, as it did not pay attention to indirect sources and implications of power (e.g., informal influence in decision making) and subordinate's acceptance of power. Parsons [12] considered power as "*a specific mechanism to bring about changes in the action of organizational actors in the process of social interaction*".

Most contemporary organization theories explore both formal (normative, prescribed) and informal (subjective, human-oriented) aspects of power [4, 13, 17]. Formal power relations are documented in many modern organizations and, therefore, can be explicitly represented in models on which enterprise information systems

(EISs) are based. The representation of formal power in EISs has a number of advantages. First, it allows a clear definition of rights and responsibilities for organizational roles (actors) and a power structure. Second, based on the role specifications, corresponding permissions for information, resources and actions can be specified for each role. Third, explicitly defined rules on power enable the identification of violations of organizational policies and regulations. Fourth, data about power-related actions (e.g., empowerment, authorization) can be stored in an EIS for the subsequent analysis.

For modeling of power relations the rich theoretical basis from social science can be used. Notably many modern EISs implement no or very simplified representations of power relations and mechanisms [3, 16]. One of the reasons is that concepts and definitions provided in social theories are often not operational and, therefore, cannot be directly used in automated information systems (EISs). To make use of these theoretical findings in EISs, power-related concepts should be formally grounded.

The first step to make the concept of power operational is to provide a clear and unambiguous meaning for it (or for its specific aspects). In this paper this is done by identifying the most essential characteristics and mechanisms of power described in different approaches and by integrating them into two broad categories: formal power (or authority) and informal power (or influence), which are described in Section 2. Further this paper focuses on the formal representation of authority, for which a formal language is described in Section 3. Moreover, Section 3 illustrates how the introduced formal language can be used to model authority systems of different types of organizations. Section 4 discusses the integration of formal authority relations into an automated EIS. Finally, the paper concludes with a discussion in Section 5.

2 Power, Authority and Influence

As in many contemporary social theories [4, 13], we assume that power can be practiced in an organization either through (formal) *authority* or through (informal) *influence relations*. Authority represents formal, legitimate organizational power by means of which a regulated normative relationship between a superior and a subordinate is established. Usually authority is attached to positions in organizations. For example, authority of some managerial positions provides power to hire or to fire; to promote or to demote; to grant incentive rewards or to impose sanctions. In many approaches it is assumed that authority implies involuntary obedience from subordinates. Indeed, as authority has a normative basis that comprises formal, explicitly documented rules, it is expected that subordinates, hired by the organization, should be aware of and respect these rules, which implies the voluntary acceptance of authority.

All manifestations of power that cannot be explained from the position of authority fall into the category of influence. In contrast to authority, influence does not have a formal basis. It is often persuasive and implies voluntary submission. Some of the bases of influence are technical knowledge, skills, competences and other characteristics of particular individuals. Influence is often exercised through mechanisms of leadership; however, possession of certain knowledge or access to some resources, as well as different types of manipulation may also create influence. Influence may be realized in efforts to affect organizational decisions indirectly.

Although authority and influence often stem from different sources, they are often interrelated in organizations. For example, the probability of the successful satisfaction of organizational goals increases, when a strong leader (meaning a leader that has a great value of influence) occupies a superior position of authority. Furthermore, sometimes patterns of influence that frequently occur in an organization may become institutionalized (i.e., may become authority relations).

Modeling methods for authority and influence are essentially different. While authority relations are often prescriptive and explicitly defined, influence relations are not strictly specified and may vary to a great extent. Therefore, whereas authority relations can be generally represented in EISs, the specification of influence relations is dependant on particular (cognitive) models of agents that represent organizational actors. Relations between authority and influence can be studied by performing simulation with different types of agents situated in different organizational environments. The focus of this paper is on modeling of formal authority relations. Influence relations and relations between authority and influence will be considered elsewhere.

3 Authority: A Formal Approach

First, in Section 3.1 a formal language for specifying authority-related concepts and relations is introduced. Then, Section 3.2 discusses how the introduced language can be used for representing authority structures of organizations of different types.

3.1 A Formal Language

Simon [19] describes three contributions of authority for an organization: (1) the enforcement of responsibility, (2) the specialization of decision-making, and (3) the coordination of activity. Based on this and other theoretical findings that describe power, duties and responsibilities of organizational positions [11], a number of relations for the specification of formal authority can be identified. These relations are defined on positions (or roles), without considering particular agents (individuals). The relations are formalized using the order sorted-predicate language [10].

We represent all activities of an organization (including decision making and personnel-related activities) by tasks. Each organizational role is associated with one or more tasks. Roles may have different rights and responsibilities with respect to different aspects of the task execution. Furthermore, often several roles may potentially execute or manage certain tasks. This is represented by the relation

is_authorized_for: r:ROLE x aspect: ASPECT x a:TASK, where aspect has one of the values {execution, monitoring, consulting, tech_des (making technological decisions), manage_des (making managerial decisions), user_defined_aspect}.

All types of decisions with respect to a particular task can be divided into two broad groups: *technological* and *managerial decisions* (inspired by [1]). Technological decisions concern technical questions related to the task content and are usually made by technical professionals. Managerial decisions concern general organizational issues related to the task (e.g., the allocation of employees, task scheduling, the establishment of performance standards, provision of resources, presenting incentives and sanctions). Managers of different levels (i.e., from the lowest level line managers to strategic apex (top) managers) may be authorized for making different types of managerial decisions varying from in scope, significance and detail. A particular decision type is specified as an aspect in the is_authorized_for relation. The same holds for technological decisions. Whereas *consulting* has a form of recommendation and implies voluntary acceptance of advices, decisions imposed on a role(s) that execute(s) the task are considered as imperatives with corresponding implications.

Authorization for *execution* implies that a role is allowed to execute the task according to existing standards and guidelines. Whenever a problem, a question or a deviation from the standard procedures occurs, the role must report about it to the role(s) authorized for making technological/managerial (depending on the problem type) decisions and must execute the decision(s) that will follow.

Monitoring implies passive observation of (certain aspects of) task execution, without intervention.

Notice that other aspects of task execution described in the managerial literature (e.g., control, supervision) can be represented as a combination of already introduced aspects. In particular, control can be seen as the conjunction of monitoring and making technological and/or managerial decisions aspects; supervision can be defined as the combination of consulting and control. Furthermore, the designer is given the possibility to define his/her own aspects and to provide an interpretation to them.

Although several roles in an organization may be authorized for a certain aspect related to some task, only one (or some) of them will be eventually (or are) responsible for this aspect. For example, the responsibility of a certain role with respect to the task execution means that the role is actually the one who will be performing the task and who holds accountability of the task execution. Furthermore, responsibility for the task execution implies allowance to use resources required for the task performance. The responsibility relation is specified as:

is_responsible_for: r:ROLE x aspect:ASPECT x a:TASK: task a is under responsibility of role r with respect to aspect (defined as for authorized_for)

Some roles are authorized to make managerial decisions for authorizing/disallowing other roles for certain aspects with respect to task execution. The authorization/ disallowance actions are specified by the following relations:

authorizes_for: r1:ROLE x r2:ROLE x aspect: ASPECT x a:TASK: role r1 gives the authority for aspect of task a to role r2.

disallows: r1:ROLE x r2:ROLE x aspect: ASPECT x a:TASK: role r1 denies the authority for aspect of task a for role r2.

However, to make a role actually responsible for a certain aspect of the task, another role besides the authority to make managerial decisions should also be the superior of the role with respect to the task. Superior-subordinate relations with respect to organizational tasks are specified by:

is_subordinate_of_for: r1: ROLE x r2: ROLE x a:TASK. Then, responsibility is assigned/retracted using the following relations:

assigns_responsibility_to_for: r1: ROLE x r2:ROLE x aspect: ASPECT x a:TASK: role r1 assigns the responsibility for aspect of task a to role r2.

retracts_responsibility_from_for: r1: ROLE x r2:ROLE x aspect: ASPECT x a:TASK: role r1 retracts responsibility from role r2 for aspect of task a.

Using these relations superiors may delegate/retract (their) responsibilities for certain aspects of tasks execution to/from their subordinates, and may restrict themselves only to control and making decisions in exceptional situations.

In [7] control over resources is identified as an important source of power. Therefore, it is useful to identify explicitly which roles control resources by means of the relation has_control_over: r1: ROLE x res:RESOURCE. In the proposed modeling framework the notion of resource includes both tangible (e.g., materials, tools, products) and abstract (information, data) entities.

In many modern organizations rewards and sanctions form a part of authority relation, thus, are explicitly defined:

grants_reward_to_for: r1: ROLE x r: REWARD x r2: ROLE x reason: STRING: role r1 grants reward r to role r2 for reason

imposes_saction_on_for: r1: ROLE x s: SANCTION x r2: ROLE x reason: STRING: role r1 imposes sanction s to role r2 for reason

Specific conditions (e.g., temporal, situational) under which authority relations may be created/maintained/dissolved are defined by executable rules expressed by logical formulae. The format and specification of these rules will be discussed in Section 4.

3.2 Modeling Authority Relations in Different Types of Organizations

Authority is enforced through the organizational structure and norms (or rules) that govern the organizational behavior. In general, no single authority system can be equally effective for all types of organizations in all times. An organizational authority system is contingent upon many organizational factors, among which organizational goals; the level of cohesiveness between different parts of an organization, the levels of complexity and of specialization of jobs, the level of formalization of organizational behavior, management style (a reward system, decision making and coordination mechanisms), the size of an organization and its units. Furthermore, the environment type (its uncertainty and dynamism; the amount of competitors), as well as the frequency and the type of interactions between an organization and the environment exert a significant influence upon an organizational authority structure.

In the following it will be discussed how authority is realized in some types of (mostly industrial) organizations and how it can be modeled using relations introduced in the previous Section 3.1. Due to the space limitations only informal presentation of relations is provided.

Authority in small firms of the early industrial era was completely exercised by their owners through mechanisms of direct personal control. Firm owners were managers and technical professionals at the same time, and, therefore, had authority and responsibility for all aspects related to tasks, except for their execution, responsibility for which was assigned to hired workers. The owners controlled all resources. Currently similar types of organizations can be found in family business and small firms.

With the growth of industry, which caused joining of small firms into larger enterprises, owners were forced to hire subcontractors, who took over some of their managerial functions. This can be modeled as assigning responsibility to subcontractors by the owner for some managerial and technological decisions, as well as monitoring and consulting of workers with respect to some tasks execution. The owner reserved often the right to control for himself, which included granting rewards and imposing sanctions to/on subcontractors and workers, realized through superior-subordinate relations. Organizational resources were usually controlled by the owner.

Large industrial enterprises of XX century are characterized by further increase in number of managerial positions structured hierarchically by superior-subordinate relations. Such organizations are often defined as mechanistic [17] and have the following typical characteristics: strong functional specialization, a high level of task formalization, a hierarchical structure reinforced by a flow of information to the top of the hierarchy and by a flow of decisions/orders from the top. Responsibilities were clearly defined for every position in a hierarchy. In most organizations of this type responsibility for execution was separated from responsibilities to make decisions. Managerial positions differed in power to make decisions depending on the level in the hierarchy. Often, technological decisions were made by managers of lower levels (or even by dedicated positions to which also execution responsibilities were assigned), whereas managerial decisions were made by managers at the apex. In many of such organizations managers at the apex shared responsibility for making (some) decisions with lower-level managers. Therefore, decisions that were usually proposed by lower level managers had to be approved by the apex managers. Initially such enterprises operated in relatively stable (however, sometimes complex) environmental conditions that reinforced their structure. However, later in the second half of XX century to survive and to achieve goals in the changed environmental conditions (e.g., a decreased amount of external resources; increased competition; diversification of markets) enterprises and firms were forced to change their organizational structure and behavior. In response to the increased diversity of markets, within some enterprises specialized, market-oriented departments were formed. Such departments had much of autonomy within organizations. It was achieved by assigning to them the responsibility for most aspects related to tasks, which created products/services demanded by the market. Although department heads still were subordinates of (apex) manager(s) of the organization, in most cases the latter one(s) were restricted only to general performance control over departments. Often departments controlled organizational resources necessary for the production and had the structure of hierarchical mechanistic type.

Although a hierarchical structure proved to be useful for coordination of activities of organizations situated in stable environments, it could cause significant inefficiencies and delays in organizations situated in dynamic, unpredictable environmental conditions. Furthermore, the formalization and excessive control over some (e.g., creative and innovative) organizational activities often can have negative effects on productivity. Nowadays, large enterprises often create project teams or task forces that are given complex, usually innovative and creative tasks without detailed descriptions/prescriptions. As in the case with departments, teams are often assigned the responsibility to make technological and (some) managerial decisions and are given necessary resources to perform their tasks. Usually teams have highly cohesive plain structures with participants selected from different organizational departments based on knowledge, skills and experience required for the tasks assigned to these teams. Although many teams implement informal communication and participative decision making principles [9], also formal authority relations can be found in teams. In particular, in some project teams superior-subordinate relations exist between the team manager and team members. In this case, whereas responsibility for making technological decisions is given to team members, the responsibility for most managerial decisions is assigned to the team manager. Then, the members of such teams, being also members of some functional departments or groups, have at least two superiors. In other teams the team manager plays the integrator role and does not have formal authority over team members. In this case the responsibility for decisions made by a team lies on all members of the team. Sometimes to strengthen the position of a team manager, s/he is given control over some resources (e.g., budgets) that can be used, for example, to provide material incentives to the team members.

The principles on which teams are built come close to the characteristics of the organic organizational form [17]. Some of such organizations do not have any formal authority structure, other allow much flexibility in defining authority relations between roles. In the former case formal authority is replaced by socially created informal rules. In the latter case, authority may be temporally provided to the role that has the most relevant knowledge and experience for current organizational tasks. In many organic organizations formal control and monitoring are replaced by informal mutual control and audit. For the investigation of dynamics of organic organization, informal aspects such as influence, leaderships, mental models of employees are highly relevant, which will be discussed elsewhere. Often interactions between organic organizations (e.g., of network type) are regulated by contracts. Usually contracts specify legal relationships between parties that explicitly define their rights and responsibilities with respect to some tasks (e.g., production, supply services). Several organizations may be involved in the tasks execution (e.g., supply chains for product delivery); therefore, it is needed to identify particular aspects of responsibility in contracts for such tasks. The introduced language may be used for specifying such responsibilities and their legal consequences through reward/sanctions mechanisms.

4 Integration of Authority Relations into an EIS

In our previous work a general framework for formal organizational modeling and analysis is introduced [15]. It comprises several perspectives (or views) on organizations. In particular, *the performance-oriented view* [15] describes organizational goal structures, performance indicators structures, and relations between them. *The process-oriented view* [14] describes task and resource structures, and dynamic flows of control. In *the agent-oriented view* different types of agents with their capabilities are identified and principles for allocating agents to roles are formulated. Concepts and relations within every view are formally described using dedicated formal predicate-based languages. The views are related to each other by means of sets of common concepts. The developed framework constitutes a formal basis for an automated EIS.

To incorporate the authority relations introduced in this paper into this framework, both syntactic and semantic integration should be performed. The syntactic integration is straightforward as the authority relations are expressed using the same formal basis (sorted predicate logic) as the framework. Furthermore, the authority relations are specified on the concepts defined in the framework (e.g., tasks, resources, performance indicators). For the semantic integration rules (or axioms) that attach meaning, define integrity and other types of organization constraints on the authority relations should be specified. A language for these rules is required to be (1) based on the sorted predicate logic; (2) expressive enough to represent all aspects of the authority relations; (3) executable, to make constraints (axioms) operational. Furthermore, as authority relations are closely related to dynamic flows of control that describe a temporal ordering of processes (tasks), a temporal allocation of resources etc., a language should be temporally expressive. A language that satisfies all these requirements is the Temporal Trace Language (TTL) [8]. In [18] it is shown that any TTL formula can be automatically translated into executable format that can be implemented in most commonly used programming languages.

TTL allows specifying a temporal development of an organization by a trace. A trace is defined as a temporally ordered sequence of states. Each state corresponds to a particular time point and is characterized by a set of state properties that hold in this state. State properties are formalized in a standard predicate logic way [10] using state ontologies. A state ontology defines a set of sorts or types (e.g., ROLE, RESOURCE), sorted constants, functions and predicates.

States are related to state properties via the formally defined satisfaction relation |=: state(γ, t) |= p, which denotes that state property p holds in trace γ at time t. Dynamic properties are specified in TTL by relations between state properties. For example, the first axiom on the authority relations expresses that roles that are responsible for a certain aspect related to some task should be necessarily authorized for this:

∀r ROLE ∀a:TASK ∀aspect:ASPECT ∀γ:TRACE ∀t:TIME state(γ, t) |= [responsible_for(r, aspect, a) ⇒ authorized_for(r, aspect, a)]

Another axiom expresses the transitivity of the is_subordinate_of_for relation: r1: ROLE x r2: ROLE x a:TASK. Due to the space limitation other axioms are not considered.

In general, rules that describe processes of authorization, assigning/retracting of responsibilities may have many specific conditions. However, to assign responsibility for some aspect of a task a role should necessarily have at least the responsibility to make managerial decisions and be the superior (with respect to this task) of a role, to which the responsibility is assigned. All other conditions may be optionally specified by the designer. Responsibility may be assigned on a temporal basis. To specify that a responsibility relation holds in all states that correspond to time points in the time interval limit, a responsibility persistency rule should be defined:

∀asp: ASPECT ∀r1,r2:ROLE ∀a:TASK ∀γ, ∀t1, t2:TIME state(γ, t1) |= is_responsible_for(r1, asp, a) & state(γ, t2) |= assigns_responsibility_to_for(r1, r2, asp, a) & (t1-t2) < limit ⇒ state(γ, t1+1) |= is_responsible_for(r1, asp, a)

Using concepts and relations from other organizational views, more complex constraints related to formal authority can be described. For example, "the total amount of working hours for role r1 should be less than a certain limit". This property can be automatically verified every time when roles are assigned additional responsibilities for some tasks. This is particularly useful in matrix organizations [17], in which roles often combine functions related to different organizational formations (departments, teams), and, as a result, their actual workload may not be directly visible.

Another example is related to rewards/sanctions imposed on a role depending on the task execution results. As shown in [15], performance indicators (PIs) may be associated with organizational tasks that represent performance measures of some aspects of the tasks execution. Depending on the PIs values, a company may have regulations to provide/impose some rewards/sanctions for roles (agents) responsible for the corresponding tasks. Although such rules are rarely completely automated, still an EIS may signal to managers about situations, in which some rewards/sanctions can be applied. For example, the system may detect and propose a reward granting action to the manager, when a role has been keeping the values of some PI(s) related to its task above a certain threshold for some time period [period_start, period_end]. In TTL:

∀γ, t1 t1 ≥ perod_start & t1 ≤ perod_end & state(γ, t1) |= [is_responsible_for(r2, execution, a1) ∧ measures(PI1, a1) ∧ is_subordinate_of_for(r2, r1, a1) ∧ PI1.value > limit] ⇒ state(γ, period_end+1) |= grants_reward_to_for(r1, bonus_5_procent, r2, excellent_performance_of_a1)

Based on these rules each EIS user (a role) receives only relevant to him/her information and is allowed to perform actions that are in line with his/her (current) responsibilities defined in the system. Furthermore, (possible) outcomes of each action of the user are evaluated on a set of (interdependent) authority-related and other organizational constraints, and based on this evaluation the action is either allowed or prohibited. Moreover, some authority-related constraints may be defined and checked by managers to investigate certain aspects of organizational performance (e.g., efficiency and redundancy of authority structure). An automated method that enables such analysis is described in [14].

5 Discussion

This paper makes the first step towards defining the formal operational semantics for power-related concepts (such as authority, influence, control), which are usually vaguely described in organization theory. In particular, this paper addresses formal authority, different aspects of which are made operational by defining a dedicated predicate logic-based language. It is illustrated how the introduced relations can be used for representing authority structures of organizations of different types.

Modern enterprises can be described along different dimensions/views: e.g., human-oriented, process-oriented and technology-oriented. However, most of the existing EISs focus particularly on the process-oriented view. An extension of the models on which EISs are built with concepts and relations defined within the human-oriented view allows conceptualizing more static and dynamic aspects of organizational reality, thus, resulting in more feasible enterprise models. Among the relations between human actors authority deserves a special attention, as it is formally regulated and may exert a (significant) influence on the execution of enterprise processes. This paper illustrates how the concepts and relations of authority can be formally related to other organizational views, thus resulting into an expressive and versatile enterprise model.

In the future it will be investigated how the proposed authority modeling framework can be applied for the development of automated support for a separation task (i.e., maintaining a safe distance between aircrafts in flight) in the area of air traffic control. Originally this task was managed by land controllers, who provided separation instructions for pilots. With the increase of air traffic, the workload of controllers rose also. To facilitate the controllers's work, it was proposed to (partially) delegate the separation task to pilots. This proposal found supporters and opponents both among controllers and pilots. The resistance to a large extent was (is) caused by ambiguity and vagueness of issues related to power mechanisms. Such questions as “whom to blame when an incident/accident occurs?”, “which part of the task may be delegated?”, “under which environmental conditions the task can be delegated?” still remain open. By applying the framework proposed in this paper one can precisely define responsibilities of both controllers and pilots and conditions under which the responsibility can be assigned/retracted. Notice that these conditions may include relations from different views on organizations (e.g., “current workload is less than x”, “has ability a”), which allows a great expressive power in defining constraints.

References

1. Bacharach, S.B. and Aiken, M.: Communication in administrative bureaucracies. Academy of Management Journal, 18 (1977) 365-377
2. Blau, P.M. and Scott, W.R.: Formal Organizations.Chandler Publishing (1962)
3. CIMOSA – Open System Architecture for CIM; ESPRIT Consortium AMICE, Springer-Verlag, Berlin (1993)
4. Clegg, S.R.: Frameworks of Power. London: Sage (1989)
5. Friedrich, C.J. (ed.) Authority. Cambridge: Harvard University Press (1958)
6. Gulick, L.H. and Urwick, L.F. (eds.) Papers on the Science of Administration. Institute of Public Administration, New York (1937)
7. Hickson, D. J., Hinings,C.R., Lee,C.A., Schneck, R., Pennings, J. M.: A strategic contingency theory of intraorganizational power, Administrative Science Quarterly, 16 (1971) 216-229.
8. Jonker, C.M., Treur, J.: A temporal-interactivist perspective on the dynamics of mental states. Cognitive Systems Research Journal 4 (2003) 137-155
9. Lansley, P., Sadler, P.J., Webb, T.D.: Organization Structure, Management Style and Company Performance, Omega, London (1975)
10. Manzano, M.: Extensions of First Order Logic. Cambridge University Press (1996)
11. Mintzberg, H.: The Structuring of Organizations. Prentice Hall, Englewood Cliffs (1979)
12. Parsons T.: The institutionalization of Authority. In: Max Weber, The Theory of Social and Economic organization. Oxford University Press, New York (1947)
13. Peabody, R.L.: Organizational authority: superior-subordinate relationships in three public service organizations. Atherton Press, New York (1964)
14. Popova, V., Sharpanskykh, A.: Process-Oriented Organization Modeling and Analysis Based on Constraints. Technical Report 062911AI, VUA, http://hdl.handle.net/1871/10545
15. Popova, V., Sharpanskykh, A.: Modelling Organizational Performance Indicators. In: Barros, F. et al. (eds) Proc. of IMSM’07 conference (2007) 165–170
16. Scheer, A.-W.; Nuettgens. M.: ARIS Architecture and Reference Models for Business Process Management. In: van der Aalst, W.M.P. et al. (eds.), LNCS 1806, Berlin (2000) 366-389
17. Scott, W.R.: Institutions and organizations. SAGE Publications, Thousand Oaks (2001)
18. Sharpanskykh, A. and Treur, J.: Verifying Interlevel Relations within Multi-Agent Systems. In Proc. of the 17th European Conf. on AI, ECAI'06. IOS Press, 2006, pp. 290-294.
19. Simon, H.A.: Administrative Behavior. 2nd edn. Macmillan Co., New York (1957)
20. Weber, M.: From Max Weber: Essays in Sociology. In: Gerth, H.H. and Wright Mills, C. (ed.) Oxford University Press, New York (1958)

Elements of Perception Regarding the Implementation of ERP Systems in Swiss SMEs

Catherine Equey and Emmanuel Fragnière

Haute Ecole de Gestion, 7 route de Drize, 1227 Carouge, Switzerland
{Catherine.Equey,Emmanuel.Fragniere}@hesge.ch

Abstract. ERP systems are more and more adopted in large companies. It seems that this trend is followed by small and medium companies too. We have conducted a questionnaire based survey to identify how Swiss SMEs perceive this phenomenon. The sample size is 687 of which 125 have actually implemented an ERP. Our main findings are twofold. First SMEs that have not implemented ERP invoke concerns (e.g. costs), which are typically not perceived as major problems by SMEs that went through an ERP implementation. Indeed the latter companies generally acknowledge that ultimately benefits (e.g. improved business information) significantly exceed costs and difficulties of implementation. Second, this survey brings new empirical knowledge on the implementation, utilization and benefits provided by ERP systems in Swiss SMEs. We primarily show that satisfaction provided by the use of the ERP system is not dependent on the size and sector of the SMEs.

1 Introduction

Since the late nineties, the vendors of integrated management tools also called ERP (Enterprise Resource Planning) are facing a saturation of their main market, which is essentially intended to large companies. To find new customers they are trying to sell their products to the "mid-market" (companies from 100 to 500 persons) represented partially by Small and Medium Enterprises (SMEs). However, it seems that few SMEs have actually implemented an ERP (which is confirmed by this present survey).

We have conducted a survey research (questionnaire based) to study the level of implementation and of use of ERP systems in Swiss SMEs. To our knowledge this is the first study of this type ever conducted in Switzerland. The originality of this work also lies on the qualitative aspects addressed in the questionnaire like the value added provided by ERP systems in terms of satisfaction, as well as managerial difficulties encountered when implementing and using ERP systems.

We learn for instance that the main difficulties encountered during the implementation phase correspond to the "complexity" of these systems. In terms of difficulty of use, companies cite on top of all the "resistance to change" as well as the "lack of training". Satisfaction regarding expectations of benefits does not differ significantly between small and medium companies as well as types of industry. On the other hand if size does not affect satisfaction perceptions, companies belonging to a group have been in general forced by the headquarter to adopt the ERP.

In this paper, we provide findings under the form of summarized descriptive statistics and hypothesis testing. For the hypothesis testing section, we solely focus on the satisfaction perceived by ERP users. All information gathered in the questionnaire related to cost is voluntary skipped because of the limited length of the paper. It is organized as follows. In Section 2, we present a literature review related to managerial implications of ERP systems in companies. In Section 3, we briefly present the questionnaire and the sampling strategy. In Section 4, we present the main descriptive statistics obtained from the survey. In Section 5, we test a few hypotheses related to the theme retained for this paper: the satisfaction of the companies having implemented an ERP. In conclusion, we indicate limitations of this study and directions for future research.

2 Literature Review

Authors of [1] present a consistent review of the research literature between 1990 and 2003. First of all, they present an overview of ERP systems and of their evolution. Then they explain the nature of the ERP market. There are, in 2001, more than a hundred providers worldwide. However, only five ERP software vendors control about 70 per cent of the market share (SAP, Oracle, JD Edwards, Peoplesoft and Bann). These authors also do a comparison of papers in the field of ERP selection criteria.

In [2], difficulties to come up with one definition of ERP are explained. Authors point out the diversity of perspectives of academic experts and outline that "ERP is not a term referring to a distinct object but rather a category (…) a range of similar products". They show that ERP does not only focus on resources but, also on business processes and they reveal terminology deficiencies. These authors finally conduct a historical analysis of MRPII and ERP. They conclude that ERP-related concepts are complex and that we still need to provide a comprehensive definition.

[3] identified ten critical factors to the successful outcome of acquiring an ERP system. The factors that stand out the most are as follows: "clear and unambiguous authority, a structured, rigorous and user-driven process, its planning, the establishment of criteria and the sense of partnership that the team works to establish not only with various user commitments, but also with potential vendor." They believe that the acquisition success depend on the combination of several critical factors.

[4] presents a new and dynamic model of ERP success factors which should to improve implementation strategies. They point out the relationships between critical successes factors such as: organizational context, supporters, project organization and outcomes.

[5] analyses software development failures that costs organizations billions of dollars. The author reveals that one-third of all software developments fail. He points out that generally this is the biggest and the most complex projects which fail. Clear and realistic goal and team's expertise are also crucial to the success of these projects.

[6] compares the perceptions of managers and end-users on selected implementation factors. He proposes, by understanding these differences of perception, interventions such as training and communication that can help implementation success.

[7] shows that expert groups seek to influence the ERP's implementation and development. Especially accountants use their position and their professional expertise to influence the introduction of ERP system.

[8] studies informal control mechanisms on information system (IS) adoption. His study shows that informal controls should be applied to the ERP systems implementation in order to enhance tacit and social aspects of IS management. He points out that "uncertainty avoidance culture and intrinsic motivation of end users in ERP implementation influence individual user's ease of use and usefulness of such systems".

[9] studies differences between ERP's user expectations and managerial policy by a case study of SAP implementation.

As per large companies, the literature about ERP and SMEs is rich and varied. Again critical success factors and ERP selection processes have the favor of the authors. [10] adopts the case study research methodology to study the implementation activities in order to point out criteria which allow a successful installation. They indicate that "effective executive management commitment can help a project to achieve success" and that the choice of the "executive sponsor" is important.

[11] studies critical success factors of ERP in order to propose a structured approach to help SMEs. Based on the literature, they consider five critical success factors (CSFs): management and organization, process, technology, data and people. They emphasize that some CSFs are more important than others. For instance, "people" is the main CSF.

[12] studies factors affecting ERP system adoption and compares SMEs to large companies. Their empirical research shows a strong correlation between company size and ERP adoption. In the opposite, the business complexity seems to be a "weak predictor of ERP adoption".

[13] studies differences in ERP system selection processes between SMEs and large sized organizations. The main differences are "a different approach to staffing the group performing the selection process", for instance large organizations engage more persons in decision making processes than SMEs. SMEs also select ERP with less complex models and less expensive methods.

In conclusion, we see that solely the paper [12] studies the criteria which affect the adoption of ERP but their research focus on the differences between SMEs and large companies. Research papers do not explore the situation of SMEs in relation to the adoption of ERP (rate of use), neither in Switzerland nor in any other country. Typically, the profile of SMEs which use ERP and their "perception" regarding the ERP implementation are unknown. For instance, data about cost, project length and number of employees involved in ERP implementation are rarely raised. Perceived rate of success and satisfaction are also not very much explored in the literature.

3 Questionnaire and Sampling Plan

The methodology to address the research question is based on a questionnaire survey. In the first phase of the research project, we have conducted in-depth interviews with Swiss-French companies. This multiple cases study (see [14]) led to the development of a few research questions along with associated research hypotheses. We had then the material to design the questionnaire. The first version of the questionnaire was built with the help of Abacus (which is the leader of ERP vendors' for SMEs in the German part of Switzerland), Microsoft, Oracle and SAP senior consultants. The final version of the questionnaire included 7 major parts: contact, activities and financial information about the enterprise, specificities of ERP implemented, implementation project description, project organization, benefits and outcomes related to the use of the ERP system, difficulties and troubles encountered.

From November 2005 to April 2006 more than 4'000 Swiss SMEs (evenly spread in the Swiss territory, so this is actually a national survey) were contacted to take part to this study. The questionnaire was administered essentially by mail. An online version of the questionnaire was also available. The questionnaire was declined in four versions: French, German, Italian and English. The French version is integrated in the appendix of the French technical report (see [15], for the other versions, please contact the authors of this paper).

Addresses of Swiss' SMEs were received from the Swiss office of statistics (OFS) and the selection was made according to the main two following criteria: the size (in terms of numbers of employees only) as well as the linguistic area.

The stratification of the sample was realized in such manner that 75% of the sample are companies of the German part of Switzerland, 20% are companies of the French part of Switzerland, and 5% are companies of the Italian part of Switzerland. Moreover, we took into account that 84% of companies employ 1 to 49 employees, and 16% of companies employ 50 to 249 employees.

In order to increase the number of answers, a follow up was done by phone's interviews. We ultimately obtained a response rate of about 17, 2%. Finally, a total of 687 Swiss SMEs have answered the questionnaire. Out of the 687 answers received only 18.2% of SMEs are indicating using an ERP (ERP users: 18.2% or 125, non ERP users: 81.5% or 560, no response: 0.3% or 2). This indicates a low level of penetration in Swiss SMEs (less than 20%). Data have been analyzed with the STATA and SPSS statistical packages.

4 Descriptive Statistics

The main part of the questionnaire was dedicated to companies which use an ERP. However, companies that had not implemented an ERP were asked about their motivation for not implementing an ERP. "High cost" (21%), "non necessity" (11%) and "lack of knowledge" (5%) are the main reasons invoked by Swiss SMEs. More than 40% of the respondents gave no answer to this question, indicating that a large part of the respondents do not seem to be concerned by ERP systems.

As indicated in the literature review, [12] provided findings regarding the ERP system adoption based on a survey analysis by comparing a sample of large companies with a sample of SMEs. They rejected the hypothesis that, the reasons of not implementing an ERP in a SMEs is due to the business complexity. Indeed they showed that the main reason invoked is a perception of high cost related to the implementation of an ERP system.

It is relevant that, as we would see further, to emphasize that the reasons for not implementing an ERP in SMEs are not similar to the difficulties meet up by SMEs' users of ERP. SMEs that have implemented an ERP are frequently dissatisfied by the complexity of these tools. On the other hand, our study shows that cost is rarely an issue of dissatisfaction for ERP users. An explanation that could be added based on our study (i.e. 40% of respondents give no answer) is that this perception of high cost comes from the lack of knowledge about ERP systems. However this point should be investigated in a further research to validate this hypothesis.

In this paper, we skip most of the detailed results related to the ERP software specificities (for more information, the reader can have a look at [15]). We just report the main figures. The following descriptive statistics are drawn from the sample of 125 respondents corresponding to ERP users who had to fill in the detailed version of the questionnaire.

The choice to implement or not an ERP is not related to the cultural and/or linguistic characteristics of the companies. The language and canton (i.e. Swiss states) of residence, variables capturing this kind of cultural differences, confirm this statement.

The size of SMEs, in terms of number of employees, is an important factor explaining the adoption of ERP systems. Indeed, the comparison with the Swiss national average and our sample reveals that close to 86% of the Swiss companies have less than 50 employees. In our sample of ERP users, companies with less than 50 employees accounts for only 53% of them. In the same way, the Swiss economy counts only 1% of SMEs of more than 100 employees, whereas our sample of ERP users is compose with nearly 26% of companies of more than 100 employees. These descriptive statistics indicate that larger SMEs are more inclined to adopt an ERP system.

Most of the respondents indicate that they are in a phase of growth. Only 7% of the ERP users sample acknowledges a reduction in their sales turnover. Among these companies in phase of recession, 75% installed their ERP more than 5 years ago, period during which their financial situation might have been different.

Industry (or the secondary sector as opposed to the tertiary and primary sectors) is over-represented in the sample of ERP users regarding the actual importance of the tertiary sector in Switzerland. Is it due to the fact that ERP systems are built upon the Material Requirement Planning (MRP) structure and are thus naturally more employed in manufacturing plants (i.e. secondary sector)? The question remains open.

Only 36% of the companies belonging to the ERP users sample have declared to be a subsidiary of a Swiss company and 19% a subsidiary of a foreign company. However, globaly, 44% of the companies declare to belong to a group. We can think that the group imposes the use of the ERP on the subsidiary company and that without this obligation, the use ratio of ERP by SMEs could be even weaker.

A significant part of the companies does not turn to the most known vendors like SAP or ORACLE. Indeed, 50.4% of the respondents ticked the item titled “other” in the question related to the ERP system installed in the company. It is also notable that no particular program dominates in the category titled "other". This result is quite surprising, since we might have thought that to ensure business sustainability companies would relate to software well positioned in the market.

Another astonishing point is that certain companies mention programs which do not have the characteristics of an ERP (e.g. AS400, Clipper). This confirms the lack of consensus regarding a clear definition of ERP systems.

Regarding the choice of the ERP system, Swiss French and Swiss German companies differs significantly. "Oracle" is the first choice of Swiss French companies. "Abacus" and "Microsoft" are the first choices of Swiss German companies. So it shows that the Swiss market for ERP systems is segmented in 2 distinct markets (the Italian part is insignificant).

The installation of an ERP lasts in 80% of the cases less than 1 year (including, for 53% of the cases, less than 6 months). Nevertheless, in 4.6% of the cases, the installation seems problematic because it requires more than 1 year and half.

The number of consultants (relative with the interns) also does not appear related to the duration of the installation of an ERP. The number of consultants required by the implementation of an ERP remains however important (one consultant for 1 employee involved in the implementation project). The companies evaluate in 71% of the cases a need for an external assistance.

The more or less important implication of the direction in project ERP has also only a moderated impact on the duration of the installation even if a strong implication of the direction contributes to drastically reduce the probability of seeing the installation lasting more than a year and half.

In terms of difficulties encountered during the implementation phase, we see that 45% of the respondents indicated the “complexity” of these systems. It is followed by the “work overload” (38%) and the “difficulties adapting the ERP system to your process (customization)” (32%). In terms of difficulties encountered when using the ERP system, we see that 32% of the respondents indicated the “resistance to change”. It is followed by the “lack of training” (29%) and again the “complexity” of these systems (25%). We thus conclude that managerial issues are the prominent difficulties associated with ERP systems. Indeed, all technical difficulties were always ranked at the bottom.

We have skipped all the results related to the costs of implementation and use, since it appears to be a minor element regarding the overall satisfaction provided by the use of ERP systems (for more details see [15]).

5 Hypotheses Testing

The main research question developed in this paper is: “Is the satisfaction regarding the benefits provided by ERP systems evenly spread among Swiss SMEs”? We have chosen to focus in this paper on one of the qualitative aspects developed in our survey. Indeed, we believe that the originality of this study lies on the measurement of qualitative variables such as the satisfaction of the use of ERP systems, and the difficulties (e.g. resistance to change).

To address this research question we propose first to test the following hypothesis scheme:

H0: Satisfaction provided by the use of the ERP system is not dependent on the size and sector of the SME

Ha: Satisfaction provided by the use of the ERP system is dependent on the size and sector of the SME

The satisfaction variable corresponds to the average for all ERP modules (finance, SCM, HR, inventory, production …) employed by each given respondent. This variable is expressed as a "likert" scale, with 1 being the weakest value and 5 being the strongest value. The size variable is expressed over 4 levels: between 10 and 49 employees, between 50 and 99 employees, between 100 and 199, and 200 and 249 employees. The sector variable is defined as either the secondary or the tertiary sector.

Table 1. Satisfaction means and standard deviation regarding the size and sector.

Size	N	Mean	Standard Deviation
10 to 49	64	3.76	0.83
50 to 99	25	3.66	0.48
100 to 199	22	4.02	0.57
200 to 249	9	3.86	0.70
Total	120	3.80	0.72

Sector	N	Mean	Standard Deviation
Secondary	63	3.69	0.78
Tertiary	57	3.91	0.63
Total	120	3.80	0.72

The statistical test we have employed to analyse this hypothesis is to compare means (of statisfaction) for every sample of size or sector involved (see Table 1). In a general manner, we notice that satisfaction is on average quite high. Practically, we have conducted an ANOVA (Analysis of Variance) for the size case, which is the method for comparing means of more than 2 independent samples. For the sector case it is just a t-test for 2 independent samples. We have retained a significance level of 5% that is the first-type error (or the risk to reject the null hypothesis when it is actually correct).

The p-value of 0.355 for the size case indicates that we cannot reject the null hypothesis at the significance level of 5%. So we conclude that satisfaction is not affected by size. The (2 tailed) p-value of 0.09 for the sector case indicates that we cannot reject the null hypothesis. However, this is not clear as it is for the size case. Nevertheless, we can conclude that the overall satisfaction related to the use of ERP systems is generally good in SMEs whatever their size or sector.

Other statistical tests can be conducted, that are typically suited for dealing with variables defined upon nominal scales (e.g. "yes" or "no", which is often the case with qualitative variables). Still related to the notion of satisfaction, Table 2 presents descriptive statistics about value different "attributes" of value added provided by ERP systems.

Table 2. Value-added by ERP Systems.

	Yes	No	No response
Improved Information	96%	3%	1%
Cost savings	48%	38%	14%
Time Saved	74%	20%	6%
Improved quality of work	95%	5%	0%

For instance, for the "Time saved" variable which was defined over the nominal scale "yes" or "no", we could explore the relationship between this variable and again the size and sector variables (see [16]). We thus test the hypothesis whether there is really a relationship between the time saved and the SME size or sector of activity. This is done through a Chi-square test, which can treat nominal variables. Here a detailed analysis would show that there is unlikely a relationship between the time saved and the size and sector variables. This analysis shows that satisfaction is on average quite high and homogeneous among the population of Swiss SMEs (other aspects than size and sector such as language have also been tested). This is the same for the benefits provided by the use of the ERP system (except maybe for the cost savings with only 48% of yes). We can thus infer that vendors and consultants should specifically address the needs and expectations of Swiss SMEs. On the other hand, there is likely no necessity to segment the SMEs market due to its homogeneity.

6 Conclusion

In this paper, we have presented the first results of the national survey we conducted on the implementation and use of ERP systems in Swiss SMEs. We have tackled the population of Swiss SMEs, because they constitute in Switzerland the essential part of the economy. We unfortunately notice that the academic literature dedicated on ERP systems and SMEs essentially focuses on Critical Success Factor's and not on satisfaction. Indeed the main contribution of this survey is to have included the perception of SMEs regarding qualitative aspects of the implementation and use of ERP systems. An ERP system leads to important organizational changes in the company. We believe that qualitative variables studied through a questionnaire-based approach can bring value to the current knowledge on ERP systems. In particular, we have shown that satisfaction of Swiss SMEs ERP users is good and quite homogenous in terms of industry type and size. However, an empirical research that attempts to measure business perceptions, is also associated with limitations. Perceptions biases are inevitable. So findings should be taken with precautions. The study also brings managerial or practical implications. In particular, developers and consultants should put more emphasis on making these systems more accessible for SMEs. We also noticed that the knowledge regarding ERP systems is quite weak among SMEs. This point will be investigated in a further research to validate this hypothesis. Moreover, the few SMEs that use an ERP systems seems to be satisfied and to acknowledge important benefits such as improved information and quality of work. Consequently, ERP systems for SMEs should become a growing and sustainable market if properly handled.

References

1. Shehab, E.M., Sharp, M.W., Supramaniam L., Spedding, T.A.: Enterprise resource planning / An integrative review. Business Process Management, Vol. 10. N°4 (2004) 359–386
2. Klaus, H., Rosemann, M., Gable, G.G.: What is ERP?. Information Systems Frontiers, Kluwer Academic Publishers (2000)
3. Verville, J., Bernadas, C., Halingten, A.: So you're thinking of buying an ERP ? Ten critical factors for successful acquisitions. Journal of Enterprise Information Management, Vol. 18. N°6 (2005) 665–677
4. King, S., Burgess, T.: Beyond critical success factors : A dynamic model of enterprise system innovation. International Journal of Information Management, Vol. 26. N°1 (February 2006) 59–69
5. Ewusi-Mensah, K.: Software Development Failures. MIT Press (2003)
6. Amoako-Gyampah, K.: ERP implementation factors. A comparison of managerial and end-user perspectives. Business Process Management, Vol. 10. N°2 (2004) 171–183
7. Newman, M., Westrup, C.: Making ERPs work: accountants and the introduction of ERP systems. European Journal of Information Systems, Vol.14. N°3 (2005) 258–272
8. Hwang, Y.: Investigating enterprise systems adoption: uncertainty avoidance, intrinsic motivation, and the technology acceptance model. European Journal of Information Systems, Vol. 14. N°2 (June 2005) 150–161
9. Lim, E.T.K., Pan, S.L., Tan, C.W.: Managing user acceptance towards enterprise resource planning (ERP) systems – understanding the dissonance between user expectations and managerial policies. European Journal of Information Systems, Vol. 14. N°2 (2005) 135–149
10. Muscatello, J.R., Small, M.H., Chen, I.J.: Implementing enterprise resource planning (ERP) systems in small and midsize manufacturing firms. International Journal of Operations & Production Management, Vol. 23. N°8 (2003) 850–871
11. Sun, A., Yazdani, A., Overend, J.: Achievement assessment for enterprise resource planning (ERP) system implementations based on critical success factors. Int. J. Production Economics 98 (2005) 189–203
12. Buonanno, G., Faverio, P., Pigni, F., Ravarini, A., Sciuto, D., Tagliavini, M.: Factors affecting ERP system adoption. A comparative analysis between SMES and large companies. Journal of Enterprise Information Management, Vol. 18. N°4 (2005) 384–426
13. Bernroider, E., Koch, S.: ERP selection process in midsize and large organization. Business Process Management, Vol. 7. N°3 (2001) 251–257
14. Equey, C., Rey, A.: La mise en place d'une solution de gestion moderne (ERP/PGI), quels enjeux pour une PME/PMI ?. 1ère partie : étude de cas détaillés, Working paper N°HES-SO/HEG-GE/C--06/1/4--CH (2004)
15. Equey, C.: Etude du comportement des PME/PMI suisses en matière d'adoption de système de gestion intégré, Working paper N°HES-SO/HEG-GE/C--06/12/1--CH (2006)
16. Bryman, A., Ducan C.: Quantitative Data Analysis with SPSS Release 10 for Windows. Routlege (2001).

ERP Non-implementation: A Case Study of a UK Furniture Manufacturer

Julie Dawson[1] and Jonathan Owens[2]

[1]Research and Commercial Partnerships, University of Lincoln, Brayford Pool, Lincoln LN6 7TS, United Kingdom
jdawson@lincoln.ac.uk
[2] Lincoln Business School, University of Lincoln, Brayford Pool, Lincoln LN67TS, United Kingdom
jowens@lincoln.ac.uk

Abstract. Enterprise Resource Planning (ERP) systems are pervasive information systems that have been fundamental in organisations for the past two decades. ERP systems may well count as the most important development in technology in the 1990s'. There are many ERP success stories; equally there are as many failure stories. This paper reviews current literature of the Critical Success Factors (CSF) of ERP implementations. This review will be used in conjunction with the case of a UK furniture manufacturer's (Company X) implementation of an ERP system. This paper considers the factors that resulted in the failure of the ERP at Company X in the initial phase of the implementation.

1 Introduction

November 2005, the authors were brought into a UK furniture manufacturer (Company X) to implement an integrated financial, manufacturing and distribution package; an ERP system. April 2006, Company X decided not to continue with the ERP adoption. The ERP system failed to be implemented.

Company X's case is not unusual; the Gartner Group (1998) reported that seventy percent of all ERP projects fail to be fully implemented, even after three years [16]. Soh *et al* [14] aid for support, they state that many companies that have installed an ERP system have had to abandon their efforts.

ERP failures have received a great deal of attention in literature [7]. Buckhout *et al* [1] found that seventy percent of ERP implementation projects fail to achieve their corporate goals. Ross [13] later found that most ERP systems fail to deliver their anticipated benefits. This problem is still profound in recent years, Ho *et al* [4] reported that currently there are relatively few ERP success stories, Kansel [6] also stated that a large number of ERP implementations still fail to meet expectations.

This paper addresses the issue of ERP failure. It considers nine critical success factors (CSFs) in the initial phase of ERP implementation. The case of Company X is analysed with respect to the identified CSFs. The findings allow conclusions to be made as to why the implementation of the ERP system at Company X failed in its initial phase.

2 What is an ERP System?

An Enterprise Resource Planning (ERP) system is a commercial software package [2, 10, 7] that promotes seamless integration of all the information flowing through a company [2]. Laudon *et al* [8] explain that an ERP system collects data from various key business processes, he states that the key business processes are: manufacturing and production, finance and accounting, sales and marketing, and human resources. The system then stores the data in a single comprehensive data repository where they can be used by other parts of the business. Managers have precise and timely information for coordinating the daily operations of the business and a firm wide view of business processes and information flows.

Davenport [2] explains how an ERP system can work, '*A Paris-based sales representative for a U.S. computer manufacturer prepares a quote for a customer using an ERP system. The salesperson enters some basic information about the customer's requirements into his laptop computer, and the ERP system automatically produces a formal contract, in French, specifying the products configuration, price and delivery date. When the customer accepts the quote the sales rep hits a key; the system after verifying the customer's credit limit, records the order. The system schedules shipment; identifies the best routing; and then working backward from the delivery date, reserves the inventory; orders needed parts from suppliers; and schedules assembly in the company's factory in Taiwan*'.

3 Why an ERP System?

During the 1990's, ERP systems became the de facto standard for the replacement of legacy systems[1] [5]. Somers *et al* [15] claim there are numerous reasons for the increasing demand of ERP systems, for example, competitive pressures to become a low cost producer, expectations of revenue growth, ability to compete globally and the desire to re-engineer the business. Markus *et al* [10] explains that ERP systems are rich in terms of functionality and potential benefits. She continues to explain that companies are implementing ERP systems for many different reasons, some companies have largely technical reasons for investing in ERP systems, other companies have mainly business reasons (Table 1).

[1] Legacy systems are existing computer systems, often referred to in this way to refer to existing systems as 'antiquated'.

Table 1. Reasons for Adopting Enterprise Systems, Markus *et al* [10].

Technical reasons	Business reasons
• Solve Y2K and similar problems • Integrate applications cross- functionally • Replace hard-to-maintain interfaces • Consolidate multiple different systems of the same type (e.g., general ledger packages) • Reduce software maintenance burden through outsourcing • Eliminate redundant data entry and concomitant errors and difficulty analyzing data • Improve IT architecture • Ease technology capacity constraints • Decrease computer operating costs	• Accommodate business growth • Acquire multi language and multicurrency IT support • Provide integrated IT support • Standardize different numbering, naming, and coding schemes • Improve informal and/or inefficient business processes • Clean up data and records through standardization • Standardize procedures across different locations • Reduce business operating and administrative expenses • Present a single face to the customer • Reduce inventory carrying costs and stockouts • Acquire worldwide "available to promise" capability • Streamline financial consolidations

4 Understanding Critical Success Factors and Failure Factors

Gargeya e*t al* [3] reported on the success and failure factors of adopting SAP, a popular ERP system. Six factors that contributed to the success and failure of ERP implementation were identified in total.

1. Working with SAP functionality/maintained scope
2. Project team/management support/consultants
3. Internal readiness/training
4. Deal with organisational diversity
5. Planning/development/budgeting
6. Adequate testing

They noted that the primary factors for success (Factor 1 and Factor 2), were different to the primary factors for failure (Factor 3 and Factor 5). Gargeya *et al* [3] noted the factors that contribute to the success of SAP implementation are not the same as the factors that contribute to the failure. This point states that this paper should be focusing on one set of factors to understand failure and another set of factors to understand success.

Umble *et al* [17] claim there are nine CSFs, from these they proposed ten reasons why ERP implementations failed within a manufacturing environment (Table 2). This work contradicts the conclusions of Gargeya *et al* [3] because nine of the reasons for failure are the same as what they define to be CSF. They are in fact stating that the factors for success and failure are the same.

Table 2. CSFs and reasons for failure, Umble *et al* [17].

Critical Success Factors		Reasons for Failure	
1.	Clear understanding of strategic goals	1.	Strategic goals are not clearly defined
2.	Commitment by top management	2.	Top management not committed
3.	Excellent project management	3.	Implementation project management is poor
4.	Organisational change management	4.	The organisation is not committed to change
5.	A great implementation team	5.	A great implementation team is not selected
6.	Data accuracy	6.	Data accuracy is not ensured
7.	Extensive education and training	7.	Inadequate education and training
8.	Focused performance measures	8.	Performance measures are not adopted to ensure that the organisation changes
9.	Multi site issues	9.	Multi site issues are not properly resolved
		10.	Technical difficulties

Researchers have focused extensively on the CSF's of ERP implementations. Loh *et al* [9] state that there is an increasing amount of research in this area. Somers *et al* [15] suggest that the failure of ERP implementations calls for a better understanding of CSFs. So indeed it has been a common trend in research, to understand and compile CSFs in order to help practitioners avoid failure.

Although Gargeya *et al's* [3] work states that different factors contribute to success and failure of ERP implementation. Umble *et al*'s [17] work contradicts this. In this vein, this paper believes it is logical to consider the CSFs of ERP implementations in terms of Company X in order to understand why the project failed.

5 The Initial Phase of an ERP Implementation

Somers *et al* [15] suggest that CSFs are much richer when viewed within the context of their importance in each stage of the implementation process. This paper will concentrate on ERP CSFs at what it defines to be the 'initial phase' of implementation, as this is the relevant research in relation to the case study of Company X. This paper defines the initial phase as a mixture of what Parr *et al* [12] consider to be called the 'planning phase' and Markus *et al* [10] label as the 'Chartering phase'.

Table 3. The Planning Phase [12] and the Chartering Phase [10].

Parr *et al* (2000) Planning Phase	Markus *et al* (2000) Chartering Phase
• Assembly of a steering committee	• Building a business case
• Determination of high level project scope and broad implementation approach	• Selecting a software package
• Selection of a project team manager	• Identifying a project manager
• Resource determination	• Approving a budget and a schedule

Table 3 outlines what Parr *et al* [12] and Markus *et al* [10] believe is involved in this phase. To clarify, the initial stage of ERP implementation in this paper is defined as involving; building a business case, the assembly of a steering committee and a project manager, approving a budget and a schedule and selecting a software package.

6 Critical Success Factors in the Initial Phase of an ERP Implementation

Loh *et al* [9] considered twenty one CSFs in SMEs. The CSFs were deduced to ten based on the grouping of similar factors together and the need for referral by five references. These ten critical success factors were then linked to their particular phase of ERP implementation adapted from the phases of Markus *et al* [10]. Loh *et al*'s [9] CSFs for the chartering phase are:

1. Project Champion
2. Business Plan and Vision
3. Project Management
4. Top Management Support

Based on the work of earlier papers, Nah et al [11] identified eleven factors that were critical to ERP implementation success. They too classified their CSFs into Markus et al's [10] phases of the ERP project lifecycle. In the chartering phase the factors noted by them are:

1. ERP Teamwork and Composition
2. Top Management Support
3. Business Plan and Vision
4. Effective Communication
5. Project Management
6. Project Champion
7. Appropriate Business and Legacy systems

Parr *et al* [12] considered two organisations implementing an ERP system and what CSFs they used in each stage. The CSFs identified at the planning phase were:

1. Management Support
2. Commitment to the Change
3. A Champion for the Project
4. A Vanilla ERP Approach

This paper intends to combine the work of Loh *et al* [9], Nah *et al* [11] and Parr *et al* [12] to obtain a unified framework of CSFs for the initial phase of an ERP implementation (Table 4).

Table 4. Unified framework of CSFs at the initial phase of an ERP implementation.

CSFs at the Initial Phase	Loh *et al* (2004)	Nah *et al* (2001)	Parr *et al* (2000)
Project Champion	✓	✓	✓
Project Management	✓	✓	
Business Plan and Vision	✓	✓	
Top Management Support	✓	✓	✓
ERP teamwork and Composition	✓	✓	
Effective Communication	✓	✓	
Appropriate Business & Legacy systems		✓	
Commitment to the Change			✓
A Vanilla ERP Approach			✓

6.1 Project Champion

Parr *et a*l [12], state that a project champion is an advocate for the system who is unswerving in promoting the benefits of the new system. The project champion should be a high-level executive sponsor who has the power to set goals and legitimise change [11]. It is a CSF that there is a project champion with these attributes is involved in an ERP implementation.

6.2 Project Management

According to Loh *et al* [9], good project management is vital and that the scope of the ERP implementation project should be established and controlled. This includes the system implemented, the involvement of business units and the amount of project reengineering needed. They continue to explain that the project should be defined in terms of milestones and critical paths. Deadlines should also be met to help stay within the schedule and budget and to maintain credibility [9].

6.3 Business Plan and Vision

A clear business plan and vision to steer the direction of the project is needed throughout the ERP lifecycle [1]. There should be a clear business model, a justification of investment, a project mission and identified goals and benefits [11].

6.4 Top Management Support

Parr *et al* [12] describe top management support as top management advocacy, provision of adequate resources and commitment to the project. Top management need to publicly and explicitly identify the project as a top priority [11]. Senior management must be fully committed with its own involvement and have a willingness to allocate valuable resources to the implementation effort [5].

6.5 ERP Teamwork and Composition

The ERP team should consist of the best people in the organisation [1]. Building a cross functional team is also critical [11]. The team should have a mix of consultants and internal staff so the internal staff can develop the necessary technical skills for design and implementation. Both business and technical knowledge are essential for success [11]. Managers should be assigned full time to the implementation and partnerships should be managed with meetings scheduled regularly [9].

6.6 Effective Communication

Effective communication is critical to the success of ERP implementations [9]. Communication includes the formal promotion of project teams and the advertisement of project progress to the rest of the organisation [5]. Expectations at every level need to be communicated [11]. Nah *et al* [11] state that communication should penetrate all levels in the company, from upper managers to bottom operators, everyone should know what to expect in the business process change. They continue to explain that communication increases the willingness of people to change and take part.

6.7 Appropriate Business and Legacy Systems

Nah *et al* [11] believe that appropriate business and legacy systems are important in the initial phase of the project as a stable and successful business setting is essential. They continue to explain that business and IT systems involving existing business processes, organisational structure, culture, and information technology effect success. The existing business and legacy systems determine the IT and organisational change required for success [5].

6.8 Commitment to the Change

Parr *et al* [12] define the commitment to the change as perseverance. They state that a company should have determination in the face of inevitable problems with implementation.

6.9 A Vanilla ERP Approach

According to Parr *et al* [12], a company should have a vanilla ERP approach in order to be successful. Parr explains that essentially a vanilla approach involves a minimum customisation and an uncomplicated implementation strategy. Organisations should be willing to change the business to fit the software with minimal customisation [5]. Holland *et al* [5] state that an organisation should try to purchase the package that fits best into its business processes.

7 Case Study: ERP Non-Implementation at Company X

7.1 Project Champion

The project management (the author) was mainly responsible for the role as the project champion. As a new employee solely employed to project manage the project, most questions and queries were directed towards her. The product Company X had chosen to implement was a well known ERP system, implemented in many companies worldwide. The project manager was able to promote the product knowing the functionality and the quality of the ERP system. However, as the project progressed and the project managers confidence in the system dropped due to the mismatch of the system and the companies requirements, the project manager no longer felt the same way about the system and this fact was picked up on by other members of staff.

7.2 Project Management

Project management at Company X could have been better. At the beginning of the project there was much uncertainty of the tasks that needed to be involved to complete the project. Communication errors had led the ERP vendors to believe they were implementing a smaller system than was required, for example, they had not been informed that any manufacturing modules were required. They were also not enlightened as to the timescales that the company wished to work to, so consequently their initial implementation dates and plans were effectively useless. The company created their own project plan including milestones. Again this suffered from being produced with communication errors as the ERP vendors input was not used. The plan was created not knowing any detail of the ERP system and how long the implementation of the software would actually take. Once the differing project plans were recognised, the problem was addressed. New plans were not drawn up however because the problem of the ERP system not fitting the company was highlighted, and this problem needed to be addressed before any further plans could be made.

7.3 Business Plan and Vision

Upon investigation it was evident Company X did not have a clear business plan and vision for the ERP system. Although some goals and benefits were identified, nothing was documented properly and defined in a united format. In November 2005, at the beginning of the project manager's recruitment, there was no clear idea what the ERP systems intention was. The modules of the ERP system that were purchased contradicted the majority of senior managers' ideas of what the system would do and what was actually required by the system. It was established that a project mission was non existent. The justification of investment was also a subject not approached in great detail.

7.4 Top Management Support

The senior management at Company X were committed to the project. Three of the four Directors were members of the steering committee, which meant they gave their input on the project on a regular basis. All of the steering committee members were encouraged to be committed to the project by the Directors. The budget for the ERP system had been approved and committed in the form of a contract between Company X and the ERP vendor. Overall time and money was allocated. The project however, may have benefited from top management publicly and explicitly identifying the project as their top priority. As the project progressed, the existing Managing Director left the company. A new Managing Director joined the company. The resources that had previously been allocated for the project were now in question. Especially as the project scope looked as if it was going to increase which of course meant the cost of the project would increase. The concerns from the new Managing Director made the commitment of the other Directors involved in the projects waver. This was portrayed in lack of attendance in meetings.

7.5 ERP Teamwork and Composition

The steering committee was cross functional. It consisted of the senior manager of each department and all of the directors that were available. An ERP consultant was also involved at the initial phase of the project. The ERP consultant's knowledge regarding the chosen ERP system was limited, although he still contributed well to the team. It would have been preferable to assign more than one person to the project full time, however because Company X is a SME this was simply not feasible. All the team were committed and meetings were scheduled regularly. Overall Company X achieved well in terms of teamwork and composition.

7.6 Effective Communication

Company X communicated well within the steering committee group. However, communication to outside of the steering committee was limited. Users could find out about the ERP project by asking questions of the steering committee, however no other formal way of communication was identified. In hind sight the project should have had newsletters or made use of notice boards and intranets.

7.7 Appropriate Business and Legacy Systems

Company X had its business faults prior to the ERP system implementation. Some business processes were duplicated or ineffective, especially processes that stretched over departments. Employees were allowed to carry out tasks in their own ways, which led to array of formats and systems. Business processes did not seem to be the businesses priority, rightly or wrongly the opinion seemed to be that as long as the job got done, it was ok. The organisational culture was not completely open to a new computer system either. Previous failed implementations of an ERP system had left the organisation guarded. Company X possibly was not the right company to adopt an ERP system, especially at that time.

7.8 Commitment to the Change

Company X was committed to the project. All the steering committee gave the project their full attention in terms of attending all the required meetings, doing all of the work required and being positive about the project. When problems occurred with the fit of the ERP system, the steering committee focused on all of options that were available at the time and came up with the most appropriate solution. The company can be seen as being committed to the change from this perspective.

7.9 A Vanilla ERP Approach

Company X had a vanilla ERP approach. They realised the time and cost implications of customising an ERP system extensively. They were extremely anti –customisation, this was made clear in all of the initial meetings.

Reviewing the business processes began shortly after the employment of the project manager. It immediately became apparent that the ERP system selected was a bad fit for Company X. Company X manufacture make to order furniture, they need the flexibility to make almost anything requested. This means that Company X needed an ERP system with a good product configurator. The ERP system selected did not have a product configurator. The vendors did not suggest using an external configurator or integrating the system with the existing bill of material system. It was later discovered that this may have been because the ERP vendor had never implemented the system in a similar manufacturer and the system was mainly marketed towards service organisations not manufacturing. Although Company X had a vanilla ERP approach. This approach was distorted because of the current situation. Company X could not implement the chosen ERP system in a vanilla format because it was a bad fit for the company.

Table 4: Summary of the extent Company X achieved each CSF.

CSFs at the Initial Phase	Company X achieved	Company X partly achieved	Company X did not achieve
Project Champion		✓	
Project Management		✓	
Business Plan and Vision			✓
Top Management Support		✓	
ERP Teamwork and Composition	✓		
Effective Communication		✓	
Appropriate Business & Legacy systems			✓
Commitment to the Change	✓		
A Vanilla ERP Approach		✓	

8 Discussion

Company X's ERP system failed to be implemented, it failed in the initial phase. What constitutes ERP failure can be questioned. However it is almost irrefutable that the ERP system at Company X failed.

So why did it fail? Company X successfully achieved the CSFs, teamwork and composition and commitment to the change. They had a cross functional steering committee who were committed to the project, attended scheduled meetings regularly and faced the problems with the ERP system with determination. However, Company X's ERP implementation failed to be implemented; so it appears that achieving two CSFs, teamwork and commitment to change were not enough to make a successful ERP implementation in the initial phase.

It can be said that Company X only partly achieved the CSFs, project champion, project management, top management support, effective communication and a vanilla ERP approach. There was a project champion, however as the project progressed, the project champions promotion for the project diminished. Project Management of the ERP implementation was not admirable; the project plans differed between the ERP vendor and Company X so the timescales were not defined. The project was clearly supported by three Directors, however top management support was hindered by the existing Managing Director leaving the company. His replacement was a new Managing Director who did not support the project. Communication within the steering committee was good, however, communication outside of the steering committee outside of the project was limited. There was no formal way of communicating the project to users. Although the company had a vanilla ERP approach, the selected ERP system was not a good fit to the companies processes so the vanilla approach was distorted. Company X only partly achieved five CSFs. A partly achieved CSF could have been a reason for the ERP failure. For example, the Project Champion's diminishing support for the project could have led to the rejection of the ERP system to the users which then progressed to the rejection of the system by the steering committee and a discontinuation of the whole project. Effectively, to only partly achieve can be said to fail in some way, and that partial failure could have led or contributed to the whole system failure.

Company X did not achieve the CSFs, business plan and vision and appropriate business and legacy systems. Although some goals and benefits were identified, nothing was documented properly and defined in a unified format. The company did not have the appropriate business and legacy systems. The culture at Company X was guarded against a new IT system. Failing to achieve these CSFs could have been the reasons for the failure of the ERP in the initial phase.

This research has distinguished that there is not one reason for the failure of Company X's ERP system. This research has defined seven reasons (five partly achieved CSFs and two not achieved CSFs) out of nine reasons that may have caused failure. Gargeya [3] found that factors leading to success and failure are complex and do not occur alone. This research clearly supports this assumption.

9 Conclusion

Considering CSFs has allowed the authors to explore a wide variety of explanations for the ERP failure in the initial phase of implementation at Company X. Prior to this research, it was deemed by the authors that Company X's ERP demise was simply due to lack of support of the project by the Managing Director. However, looking at the case in terms of CSFs, it was discovered that although lack of top management was a critical factor, it was not the only factor that led to the ERP failure. Company X only partly achieved and did not achieve in total seven CSFs. So in total, this paper has identified seven reasons for the failure of the ERP implementation at Company X. This research is extremely useful for Company X to understand when they undertake IT projects in the future. Recognising what the CSFs are and that they need to be achieved will encourage successful IT implementations. This research is also useful for researchers and practitioners who wish to implement ERP systems. They will know that nine factors need considering in the initial phase of ERP implementation, and failure to meet or to only partially meet seven of these factors results in failure.

This paper is limited because it fails to identify the importance of each CSF in the initial phase of ERP implementation, for example whether the non achievement of one CSF is more critical to failure than others. This paper proves that the failure and the partial failure of seven out of nine CSFs cause ERP failure in Company X, however it does not recognise how many non achieved CSF were actually necessary to cause failure. This paper also fails to recognise if a particular combination of factors led to the ERP failure, for example, if the non achievement of two CSFs together signifies failure.

This research would be validated further if the same conclusions were made by considering the subject from a different perspective; possibly looking at the ERP implementation from a user perspective, the Managing Director's perspective or the ERP vendor's perspective.

References

1. Buckhout S, Frey E and Nemec J, 'Making ERP Succeed: Turning Fear into Promise', IEEE Engineering Management Review, Second Quarter, 1999, pp116-123.
2. Davenport T H, 'Putting the Enterprise Back into the Enterprise System', Harvard Business Review, July-August 1998, pp121-131.
3. Gargeya V B and Brady C, 'Success and Failure factors of adopting SAP in ERP System Implementation', Business Process Management Journal, Volume 11, No 5, 2005, pp 501-516.
4. Ho C F, Wu W H and Tai Y M, 'Strategies for the Adaptation of ERP systems', Industrial Management and Data Systems, Vol 104, No 3, 2004, pp 234-251.
5. Holland C P and Light B, 'A Critical Success Factors Model for ERP Implementation', IEEE Software, May-June 1999, pp 30-36.
6. Kansel V, 'Enterprise Resource Planning Implementation: A Case Study', The Journal of American Academy of Business, Cambridge, Vol 9, No 1, March 2006, pp165-170.
7. Kim Y, Lee Z and Gosain S, 'Impediments to successful ERP implementation process', Business Process Management Journal, Vol 11, No 2, 2005, pp 158-170.
8. Laudon KC and Laudon J P, 'Management Information Systems', Pearson Education, New Jersey, 2006.
9. Loh T C and Koh S C L, 'Critical Elements for a Successful Enterprise Resource Planning Implementation in Small and Medium Sized Enterprises, International Journal of Production Research, Volume 4, 17 September 2004, pp 3433-3455.
10. Markus M L and Tanis C, 'The Enterprise System Enterprise – From Adoption to Success', in R W Zmund, 'Framing the Domains of IT Research', Pinnaflex Educational Resources, 2000, pp173-207.
11. Nah F F and Lau J L S, 'Critical Factors for Successful Implementation of Enterprise Systems', Business Process Management Journal, Vol 7, no 3, 2001 pp 285-296.
12. Parr A and Shanks G, 'A Model ERP project Implementation', Journal of Information Technology, Vol 15, 200, pp289-303.
13. Ross J W, 'Surprising Facts about implementing ERP', IT Professional, Vol 1, No 4, 1999, pp 65-68.
14. Soh C, Kien S S and Tay-Yap, 'Cultural Fits and Misfits: Is ERP a Universal Solution', Communications of the ACM, Vol 41, No 4, April 2000, pp 47-51.
15. Somers T M and Nelson K, ' The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations', Proceedings of the 34th Hawaii International Conference on System Sciences, January 3-6 2001.
16. Taube L R and Gargeya V B, 'An Analysis of ERP System Implementation', The Business Review Cambridge, Vol 4, No 1, Summer 2005, pp 1-6.
17. Umble E J, Haft R R, Umble M M, 'Enterprise Resource Planning: Implementation Procedures and Critical Success Factors', European Journal of Operational Research, Volume 146, 2003, pp 241-257.

Memory as an Elephant: How prior Events Determine User Attitudes in ERP Implementation

Lene Pries-Heje

Software Software Development Group, IT University of Copenhagen
Rued Langaardsvej 7, 2300 Copenhagen
lpries@itu.dk

Abstract. Assimilation of a standard ERP system to an organization is difficult. User involvement seems to be the crux of the matter. However, even the best intentions for user involvement may come to nothing. A case study of a five year ERP implementation process reveals that a main reason for this miss may be that the perception of usefulness in any given phase of the implementation is heavily dependant on preceding events – the process. A process model analysis identifies eight episodes and nine encounters in the case showing that users attitude towards an ERP system change between acceptance, equivocation, resistance and rejection depending on three things: (1) Dynamic between user and consultants, (2) Dynamic between different user groups, and (3) Understanding of technical, organizational and socio-technical options.

1 Introduction

Organizations may have very diverse technical and strategic reasons for adopting integrated Enterprise Information Systems [1], [2], and success is not a monolithic concept; rather it is multidimensional and relative [1]. However use is one of the most frequently reported measures of system implementation success [3]. We therefore assert that a successful implementation must be coupled with high quality of use in order to fully realize the benefit of the ERP system. Quality of use defined as "how well an end user understand a piece of software and how effectively the user can exploit the capabilities of the software" [4]. Although research shows that most organizations implementing ERP systems may expect difficulties using the system for a shorter period after go-live [2], some organizations continue to struggle with the use of the system for years after go-live[5]. The poor quality of use can be caused by:

- The capabilities of the ERP Package being unable to meet the organizational requirements[6], [7].
- Poor design decisions regarding configuration of the ERP Package, customizations and integration with other systems [1], [8]
- Human factors: Unforeseen human enactment of the software [4] and resistance to change[9].

Having users participate and being involved in ERP implementations are considered essential for success ([10], [9], [8]and will result in a better fit of user requirements, achieving better system quality, use and acceptance [11]. “User participation” refers to the behaviours and activities that users perform in the system implementation process. User involvement” refers to a psychological state of the individual, and is defined as the importance and personal relevance of a system to a user [12]. However we should not assume that having users participate will automatically result in personal involvement and commitment to the result. To better understand how user involvement during an ERP experience is changing over time, and how it is affecting the attitude toward the new system, this paper uses a longitudinal case-study and a process-oriented view inspired by Newman & Robey’s [13] model. The analysis show how the user involvement and perceived usefulness of the system change over time as the dynamic between the participating users and consultants changes and knowledge to re-design the system and the organizational work processes is generated.

2 Research Question and Method

A longitudinal case study within the interpretive tradition of information technology studies [14] is conducted. The aim of the research is to better understand how and why the user involvement and the perceived usefulness of the system change over time. Especially how the dynamic and the communication between the users and the developers are influencing the outcome. In line with the assumptions of interpretive research I focused on the participants’ subjective descriptions of the implementation process and their expressed feelings and thoughts about their involvement and the perceived usefulness of the ERP package. The first initial analysis of the interviews revealed that actions and events in the case were strongly influenced by prior events, and that user involvement and the perceived usefulness of the system had changed over time. Thus a process model inspired by Newman and Robey’s model [13] was used to guide the analysis to focus on the social dynamics between the users and the developers (consultants) during the implementation.

The study was carried out in the Danish headquarter of an international engineering company called Alfa (pseudonym). In January 2001 Alfa started up the process of selecting and implementing a standard ERP system, and in October 2003 they went live. The case study covers a 5½ year period from January 2001 to summer 2006.

Data collection was carried out through interviews with the ERP project program manager, users serving as team leaders during the implementation, managers and end-users from all functional areas within project scope, a consultant participating in the project on the vendor side, and the vendor’s solution architect. All 16 interviews were semi-structured and lasted 1½ to 2 hours. The interviews were taped and transcribed. An initial interview with the project program manager were conducted in February 2005 and the rest of the interviews from November 2005 to August 2006. It has not been possible to follow the project from the start although it had been preferable. Thus to cover the implementation from the beginning and up to date, the interviews have been conducted with a retrospective focus.

One of the difficulties using the interview material is, that the interviewees’ interpretations of the history as well as the immediate situation often is influenced by difficulties or conflicts taking place at the time of the interview. Written project documentation has therefore been used to verify the interviews where possible. Alfa has provided elaborate documentation including detailed requirements specification, documented workshop evaluations of the candidate systems, business cases, gap analysis, issue-log and change requests.

The data analysis were an iterative process going back and forth between coding and collecting data to allow gaps to be identified and addressed, and different interviewees interpretations to be commented and reflected on by others. A hermeneutic interpretive approach [14] has been used going back and forth between the field material and different interpretations. Going through the transcribed interviews the initial analysis made it clear, that end users and IT-experts had very different interpretations of the usefulness of the new systems, and that the end users perceived the usefulness of the system to have changed over time. Therefore another round of analysis was conducted using a process model inspired by [13].

The process model focuses on sequences of events over time in order to explain how and why particular outcomes are reached. The constructs in the process model are *antecedent conditions*, *episodes* (a series of events that stand apart form each others), *encounters* (mark the beginning and end of an episode), and *outcome* over the course of time (see figure 1). The historic context of the ERP implementation is expressed through the antecedent conditions. During each episode the antecedent conditions of the episode may be challenged and the users may choose to respond by changing their involvement and attitude toward the new system. In this paper the process model is used to analyse how and why user involvement and the user's attitude toward the ERP implementation is changing over time. I have adopted Newman and Robey's four categories of episodes which are: (1) episodes led by the IT-expert – focus on the technology, (2) episodes led by the users - focus on the business, (3) episodes of joint development, and (4) "*wait and see*" episode when both party are uncommitted. As possible responses to each episode four categories of user attitude are used: (1) acceptance of the system, (2) equivocation, (3) resistance, and (4) rejection. Category 1, 2 and 4 are included in Newman and Robey's original process model, however I found, that a fourth category were necessary; rejecting a system or parts of a system may not be an option for the users, but resisting or enacting the system in order to minimise the use or the consequences is a milder but still powerful way to express non-acceptance.

3 Implementation of an ERP System at Alfa Engineering

The case organization Alfa is an engineering company with more than 80 years of experience - a leading supplier of systems, consultancy and engineering services to the pharmaceutical and biotechnological industry. The organization has 1200 employees in Europe, China and USA. Employees typically have a degree from a technical university. Most of the work in Alfa is conducted in large projects lasting several years and costing billions of US$. Thus Project Managers are quite powerful and influential.

The ERP project started in January 2001 at Alfa's headquarter in Denmark. A Project Manager with extensive ERP Project Management experience was hired to manage the project. From the very beginning it was clear, that this was a common project for management and employees in Alfa. It was never questioned that users should participate throughout the project and in all aspects of the project. A project organisation was set up and user representatives from all functional areas included in the project scope were appointed.

Alfas core business is project administration and project management on behalf of there customers, but they don't do any production. Alfa was aware that ERP systems in general are not targeted at their line of business. Therefore a thorough evaluation and selection process was to be conducted to ensure that the standard system meet their needs. Alfa spend almost a year specifying requirements, evaluating candidate systems and finally selecting a system and 9-10 month configuration and customizing the system before going live in October 2003.

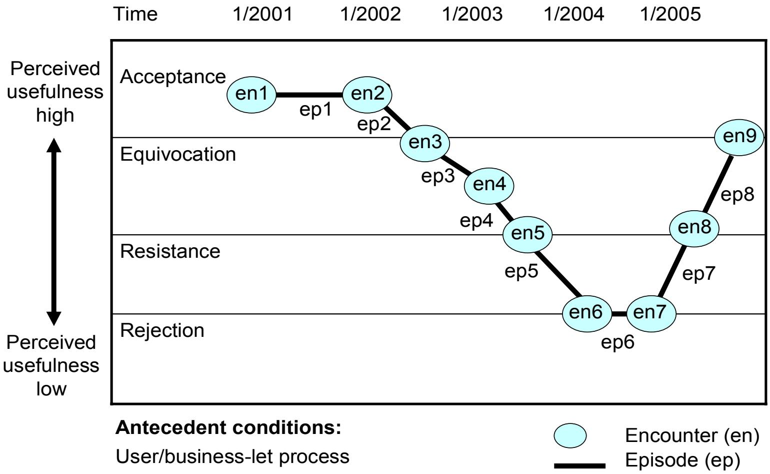


Fig. 1. Alfa's process model adapted from Newman & Robey (1992).

3.1 A Process Model for Alfa's ERP Experience

In this section Alfa's ERP process model is described and a graphical representation is depicted in figure 1.

Antecedent Conditions: Up until the decision to buy and implement the ERP system Alfa had no experience with integrated systems, and very limited experience with standard systems. Historically software had been developed specifically for functional areas and allowed the users significant influence on the design of the software.

Encounter 1 – project initiated: In order to improve the quality of services offered to the customers, improve resource management, and provide better financial control. Alfa as many companies before them had however come to the conclusion, that an integrated ERP Package providing real-time sharing of data were necessary Managers as well as users were aware that it would require the organization to adapt to the ERP system, but at the same time they wanted to continue the tradition of user participation they had good experience with and therefore an approach involving the users from the very beginning and throughout the project were decided on.

Users' attitude toward the system: Users, the ERP Project Manager and top management at ALFA acknowledged the need for a new system and the intended approach, thus the project stated out with wide acceptance.

Episode 1 - ALFA business processes and requirements specification: First all business processes that should be part of the new system were described including Finance, Purchase, Project Administration and Resource Management. A large number of users throughout the organisation were involved in the process, and a number of simple business process models on different levels were produced using Power Point as a tool. For each of the four areas knock-out criteria were defined. After the Business processes were defined, they served as a common reference for discussing the requirements focusing on input (data) triggering a process, steps within a process and output from a process. More detailed requirements for each area were defined in a dialogue between the Project Manager and the participating users. It was a long and difficult process especially because it involved a large number of users who had little or no experience defining requirements. The requirements should at the same time reflect existing processes and be open towards processes within a standard system. The users did not know what to expect from a standard system and to inspire them a couple of standard systems were demonstrated by different vendors.

Alfa defined more than 800 detailed requirements which were simple and prioritised on a scale from 1-4. Finally all the requirements were included in a spreadsheet and mailed to the candidate vendors.

Encounter 2: The requirement specification is finished; users from the four functional areas were in charge of the requirement process and the users influence on the process has not been challenged. There personal involvement is high and the expectations to the new system high.

Users' attitude toward the system: Acceptance (no changes from the antecedent conditions).

Episode 2 - Evaluating candidate ERP systems: The vendors performed a written reply and for each requirement they defined too what degree the system could meet the requirement, they used 4 categories: 'Fully as standard', 'Customisation included in future upgrades', 'Customisation not included in future upgrades', 'Not at all'.

Parallel with the requirements definition a set of criteria for evaluating the vendor were defined, and knowledge about the industry and the vendors desire to understand Alfas situation were among the more important criteria.

The three pre-qualified vendors were invited to demonstrate there system in an all-day workshop using scenarios defined by Alfa. 10-15 users participated in the workshops evaluating the system and vendor performance using an evaluation framework. Finally 1-2 reference customers for each vendor were visited. An evaluation report comparing the three candidate systems and the tenders from the vendors were composed. The results from the evaluation process were summarized and presented as quantitative and qualitative scores in a number of different areas.

Encounter 3 - Evaluation: Alfas board of directors decided to follow the recommendation given by the project group and Oracle were chosen as Alfas new ERP system.

Users attitude toward the system: The attitudes toward the systems are somewhat mixed. Some users developed an equivocate attitude but acceptance is still the domination attitude. The users were in charge of the evaluation and the analysts in charge of the demo. Some users, especially form project management, realise that the systems may not fulfil there needs. They have started to realise, that the approach for this implementation; minimise customization and require the organization to adapt to the ERP Package, will challenge there historical influence on the systems design, and therefore there anticipated usefulness of the system.

Episode 3- Re-scoping: Due to financial difficulties the ERP project were asked to cut the project cost by 5 million DKK before even starting. To re-scope the project Alfas ERP project manager and user representatives from the four functional areas together with consultants from Oracle implemented a 'Conference Room Pilot' a quick examination of the original requirements and scope. For each requirement, the implementation consultants would show the solution in Oracle, and the possibility of cutting something was discussed. This process very quickly made it visible that it would be necessary to add as well as cut requirements and scope. In the original requirement specification process the users had relied on assumptions about what a standard system would provide. Therefore the requirements now appeared to be incomplete. At the end of the two weeks the 5 millions were found and a contract were signed defining scope, price etc.

Encounter 4: A fixed price contract is signed with the vendor based on the original requirement specification with adjustments decided on during episode 3.

Users' attitude toward the system: Equivocation and some resistance, most users have started to feel that the historic situation has changed. The users got to make the re-scoping decisions, but the final result has to be approved by the steering committee. The project manager is driving the process very strictly to cut project cost and the users have to rely on the consultants' knowledge and judgment about the ERP system. Most users have now realized, that the requirement specifications not necessarily will help them achieving significant influence on the system's capability, and a feeling that the system will not provide what they asked for is developing. The process is now challenging the users' historic position of having significant influence.

Episode 4 – Configuration: In the following nine month three Conference Room Pilots were conducted. In each pilot the system "to-be" was (re-)scoped at a more detailed level and the configuration decisions were documented. The work was conducted in small workshops where user representatives and consultants worked together; the users provided knowledge about the existing work practice (requirements) and the consultants their knowledge about the standard system. The processes in the ERP Package were guiding the work and Oracle's process tool was used. The requirement specification was used as a checklist.

Encounter 5: The first version of the new ERP system is finished.

Users' attitude toward the system: Resistance and some rejection. Most of the user representatives are disappointed with the results and know it will be difficult to "sell" it to the users in there department. The functionality within resource management is considered so poor, that the users have started rejecting the functionality. The consultants were in charge of the implementation process, the capability of the ERP Package is constraining the design and the requirement specification is used to evaluate the progress of the work. The users lack experience in the configuration process and knowledge about the capabilities of ERP Package, they are totally depended on the consultants. They have all realized, that the new system will not meet the expectations of there peers and some of them feel much stressed reporting back to the peers. Conflicting interests among the users is also influencing the process, most of the user representatives feel, that the financial department is too dominant.

Episode 5 -Training and testing: The project is under an extreme time pressure. In the Alfa concern it is not allowed to implement a new financial system the last quarter of a financial year, therefore the system have to go-live at the beginning of October. Thus the training of the users takes place alongside the final testing and data conversion.

Encounter 6: 8th of October 2003 Alfa's Oracle solution went live.
Users' attitude toward the system: A lot of resistance toward the system is building up in the organization during episode 6. The users succeed at this late stage having the resource management module taken out of the implementation because the functionality in there opinion is to poor. The consultants were in charge of the configuration of the system and the modifications to the system. However the user representatives have taken over responsibility regarding training and testing, and the overall responsibility for the socio-technical design.

Episode 6 – Go-live and stabilizing the system: Because of the time pressure many reports were still outstanding and a lot of promising and nice functionality were left to be implemented in a later phase. During the next months the users struggled with the system. Some parts of the system they learnt to manage but others they refused to use or used incorrectly thereby causing data quality problems as well as system malfunction in other areas. After a very turbulent period the system were stabilized and the most important reports were developed.

Encounter 7: An internal IT-competence centre was formed consisting of the project manager and some of the user representatives, and a former Oracle consultant, and a new project is decided on to improve the usefulness of the system.
Users' attitude toward the system: Resistance/rejection. As the users became more experienced using the system, they also became convinced that parts of it had to be redesigned thus using there political influence to have the design of the system support there daily work. No significant pattern in some areas the users is in charge of the process in others the members of the competence centre is in charge.

Episode 7 - The follow up project: Some of the consultants participating in the configuration had moved on to new project and some were still helping out correcting errors. Members of the new internal Oracle competence centre were assigned the roles of technology experts. The fit of the system were in some areas more problematic then in others. Meetings were set up where people from the competence centre met all user groups within Alfa. The analysts met the users with an open mind and all issues reported were noted without considering the relevance or the reason; resulting in a list of more than 500 issues. Afterwards the reasons for the issues were discussed and the appropriate action decided. Some issues were met with end user education and some with reconfiguration or customizations of the system; some were researched thoroughly, but could not be solved due to the design of standard system.

Encounter 8: At the end of 2004 the follow-up project was completed.
Users' attitude toward the system: More functionality is accepted. In general the perceived usefulness of the system increased and some of the rejected functionality were re-designed or just re-introduced and now accepted. The users in general gained more self-confidence and they started to fight to get more influence on the socio-technical design of the new system. Project managers are still fighting what they perceive as very poor system design refusing to use some functionalities and have project assistances and secretaries use the system on there behalf.

Episode 8 - Continuous improvement: Users throughout Alfa and members of the competence centre are working to increase the quality of use. To do so they are customizing the software to change the original capabilities of the ERP Package, reconfiguring the system and enacting the software. The relationship between the users and the competence centre is in some areas problematic, but in other areas the relation is very good and fruitful based on a more joint development approach.

Outcome: The project is being considered a success from a project management point of view. Cost and time estimates (episode 3-5) were met, the functionality promised were delivered and after a chaotic go-live the system is now used throughout the organization and more functionality are implemented as an ongoing process. The selection and implementation process was relatively participatory, users were involved in the selection process, the scoping of the system, the configuration and implementation, and last but not least the user's issues that remained after the go-live phase were collected and seriously addressed. Participation was encouraged and organised for. However the quality of use can be questioned, many users are still complaining, that they lack knowledge to use the system correct, more powerful users resist using functionality with what they consider poor user interface. Users participating in the ongoing implementation of new functionality are complaining about not being able to understand the capabilities of the software and the consequences of different design possibilities. Episode 3 and 4 left the organization with a lot of internal conflicts and frustration, influencing the users' behaviour in the following episodes, and causing a lot of re-design and customizations years after the project officially finished in encounter 7.

3.2 Discussion

The purpose of this paper is to understand what made the users attitude toward the system change over time. As we can see from the process model, the users were enthusiastic and actively involved in the first episodes, during episode 3 and 4 a dramatic change happened, that culminated during episode 5 and 6 having users reject functionality and resist using the system. During episode 7 and 8 a more positive attitude to the system was developed although the quality of use is still low in many areas.

In the process model the outcome of an episode implies the preceding events [13] therefore to understand what led up to the rejection and resistance in episode 5 and 6 the preceding episodes are examined. In episode 3 and 4 the user representatives found them selves in a situation, where the consultants had taken over the design process of the new system and the users had realised that the result would constrain the organizational processes in ways the end users would properly not accept. However there was not much they could do to change the situation. Most of them had themselves been involved in defining the requirement specification and choosing the system, and a fixed prise contract had been signed with the vendor using the requirement specification as a basis for setting up the new system. At the same time, the users had very limited knowledge about the ERP Package and its capabilities and design options, and therefore there ability to influence the design was rather limited. Furthermore interest conflicts between users from different functional areas were causing sub optimization, having the overall perceived usefulness of the new system deceased. Some of the user participants expressed frustration having to report back to their peers about the progress of the project because they thought they had nothing but bad news.

During episode 3 and 4 the user representatives were providing the consultants with enough knowledge about Alfa's organization to set up business processes within the scope of the project. Instinctive many of the user representatives know the usefulness of the business processes were doubtful, however the design process did not included activities evaluating the usefulness of the business processes giving them arguments to reject the design.

In episode 5 and 6 the new system was presented to the end users. Most of them had either been involved or had only very limited involvement in the previous episodes; therefore they had no loyalty conflict rejecting the new system.

Historically users in Alfa had had significant influence on the design and use of software, and probably for a good reason. Most of the employees have a university degree, there work are not easily automated, and they are knowledge workers being expected to take responsibility for the result of there work. They are not easily told just to do something. In episode 5 and 6 they were introduced to a new system that did not successfully meet there needs, they had no part in the design and were given very limited help to assimilate the new system. In response they were sending a very strong signal; large part of the systems functionality were rejected completely or enacted in ways that would cause no or very little change to the existing work practise. The user groups that during episode 3 and 4 had felt dominated by another user group chiefly resisted and rejected the system.

In episode 7 and 8 the attitude toward the system is changing direction becoming more positive. As a reaction to the resistance of the new system and the rejection of functionality in episode 5 and 6 the competence centre in episode 7 goes into a dialogue with the end users, where the user organization's perspective is in focus. The users are now in a situation where they are being heard and have an opportunity to influence the re-design of the new system although there understanding of the ERP software's capabilities and design options are still causing difficulties. In episode 8 the users influence on the re-design of the new system is continued. As users gain more experience with the system, they help each others within departments and across functional areas to understand the capabilities of the ERP software and find ways to re-design or enact the system to support their needs. Much of the work takes place without involving the competence centre or in a dialogue with them but on the end-users initiative and terms.

4 Conclusion

Assimilation of a standard ERP system to an organization is difficult. User participation and involvement seems to be critical for success. However, as this case study shows even the best intentions for user involvement may come to nothing. A case study of a five year ERP implementation process reveals that a main reason may be that the perception of usefulness in any given phase of the implementation is heavily dependant on preceding events – the process. User-led development was the antecedent condition in the case organization, and the general perceived usefulness of information systems was high. The case organization intended to continue the tradition having users participate and influence the design of the system when implementing an ERP system. However, a process model analysis shows that in reality the consultants/IT-experts take charge during the configuration and customization process challenging the status qua. Alfa is not aware, that lack of knowledge about the technological options, a fixed price contract and a strict timetable left them with no choose. As a response to the change in user-developer dynamic the users' attitude towards the ERP system change from accepts/equivocation to resistance/rejection and the users are fighting to regain control. When succeeding to regain control almost two years after go live their attitude toward the ERP system changes again. In summary the case analysis show, that the users attitude toward the system change between acceptance, equivocation, resistance and rejection over time depending on three things: (1) Dynamic between user and consultants, (2) Dynamic between different user groups, and (3) Understanding of technical, organizational and socio-technical options.

References

1. Markus, M.L. and C. Tanis, The Enterprise System Experience – From Adoption to Success, in Framing the domains of IT management: Projecting the future through the past, R.W. Zmud and M.F. Price, Editors. (2000), Pinnaflex Educational Resources: Cincinatti, USA.
2. Shang, S. and P.B. Seddon, Assessing and Managing the Benefits of Enterprise Systems: The Business Manager's Perspective. Information Systems Journal, (2002). 12: p. 271-299.
3. DeLone, W.H. and E.R. McLean, Information systems success: The quest for the dependent variable. Information Systems Research, (1992). 3(1): 60-95.
4. Boudreau, M. and D. Robey, Enacting Integrated Information Technology: Humn Agency Perspective. Organization Science, (2005). 16(1): p. 3-18.
5. Pries-Heje, L. ERP misfits: What is it and how do they come about? 17th Australasian Conference on Information Systems. (2006). Adelaide, Australia.
6. Davenport, T.H., Putting the Enterprise into the Enterprise System. Harvard Business Review, (1998).

7. Kien, S.S. and C. Soh, An Exploratory Analysis of the Sources and Nature of Misfits in ERP Implementations. 1 ed. Second-wave Enterprise Resource Planning Systems, Implementing for Effectiveness, ed. G. Shanks, P.B. Seddon, and L.P. Willcocks. (2003), Cambridge University Press. 373-387.
8. Robey, D., J. Ross, and M. Boudreau, Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change. Journal of Management Information Systems, (2002). 19(1): p. 17-46.
9. Nah, F.F.-H., K.M. Zuckweiler, and J.L.-S. Lau, ERP Implementation: Chief Information Officers' Perceptions of Critical Success Factors. International Journal of Human-Computer Interaction, (2003). 16(1): p. 5-22.
10. Kawalek, P. and T. Wood-Harper, Finding of Thorns: User Participation in Enterprise System Implementation. Advances in Information Systems, (2002). 33(1): p. 13-22.
11. Esteves-Sousa, J. and J. Pastor-Collado. Towards the unification off critical success factors for ERP implementations. in 10th Annual Business Information Technology conference. (2000). Manchester, UK.
12. Barki, H. and J. Hartwick, Rethinking the concept of user involvement, and user attitude. MIS Quarterly, (1994). 18(1): p. 59-79.
13. Newman, M. and D. Robey, A social process model of user-analyst relationships. MIS Quarterly, (1992). 16(2): p. 249-266.
14. Klein, H.K. and M.D. Myers, A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. MIS Quarterly, Special Issue on Intensive Research, (1999). 23(1): p. 67-93.

Author Index

Proceedings of the
1st International Joint Workshop on
Technologies for Collaborative Business Processes and
Management of Enterprise Information Systems
TCoB & MEIS 2007
ISBN: 978-972-8865-99-3
http://www.iceis.org