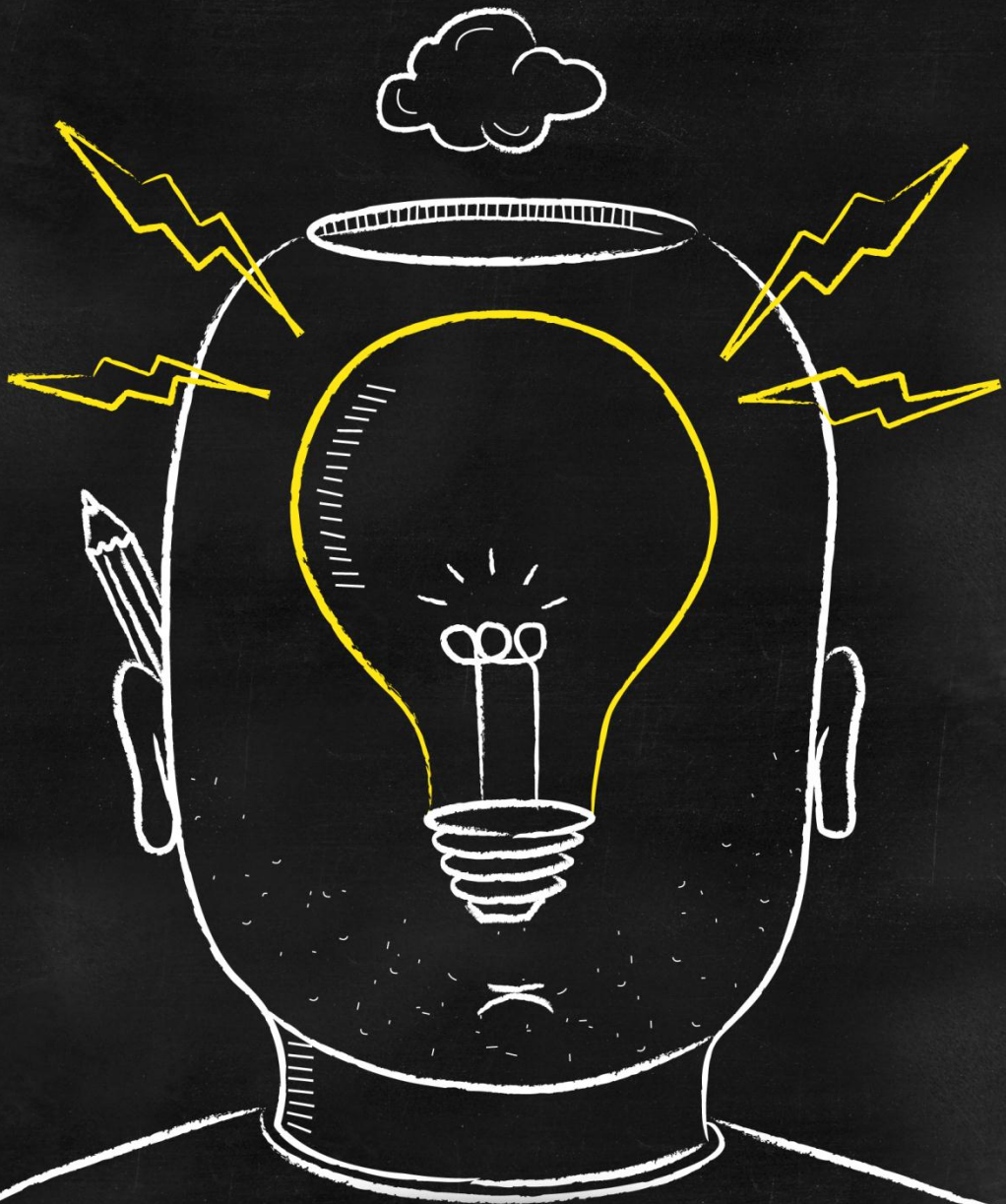


# Engineering situational methods for professional service organizations

*An action design research approach*



*Diederik Rothengatter*

Engineering  
situational methods  
for professional  
service organizations

AN ACTION DESIGN RESEARCH APPROACH

Diederik C.F. Rothengatter

## Dissertation Committee

Chairman & secretary	Prof. dr. R.A. Wessel
Promotor	Prof. dr. J. van Hillegersberg
Assistant promotor	Dr. ir. C.P. Katsma
Members	Prof. dr. S. Brinkkemper
	Prof. dr. F. Harmsen
	Prof. dr. ir. L.J.M. Nieuwenhuis
	Prof. dr. R. Wieringa



CTIT Ph.D. Thesis Series No. 11-225  
ISSN 1381-3617  
Centre for Telematics and Information Technology  
P.O. Box 217, 7500 AE  
Enschede, The Netherlands

Ph.D. thesis, University of Twente, Enschede, the Netherlands  
Printed by Ipskamp Drukkers BV, Enschede, the Netherlands

Cover design: Sjors Vervoort

Copyright: D.C.F. Rothengatter, Den Haag, 2012  
All rights reserved. No part of this publication may be reproduced without  
the prior written permission of the author.

ISBN 978-90-365-3380-5



ENGINEERING SITUATIONAL METHODS FOR  
PROFESSIONAL SERVICE ORGANIZATIONS - AN  
ACTION DESIGN RESEARCH APPROACH

DISSERTATION

to obtain  
the degree of doctor at the University of Twente,  
on the authority of the rector magnificus,  
Prof. dr. H. Brinksma,  
on account of the decision of the graduation committee,  
to be publicly defended  
on Thursday, June 7th, 2012 at 14:45

by  
Diederik Caspar Ferdinand Rothengatter

Born May 1st, 1981  
in Apeldoorn, The Netherlands

This dissertation is approved by the promotor,  
Prof. dr. J. van Hillegersberg  
and the assistant-promotor,  
Dr. ir. C.P. Katsma

Why not? Indeed!  
- Hunter S. Thompson





## i. Acknowledgments

Knowing that the acknowledgments are by far the best read section of any thesis – I haven't seen a thesis being handed over without the recipient conducting a quick scan to see if his/her name is being mentioned – I'd better start this thesis with thanking the people that made my PhD research possible and successful.

Working on a PhD on the University of Twente has been a privilege that I would recommend to everyone. I consider myself very fortunate that I was facilitated in a four year journey into the deep trenches of science; often I received joking remarks on the 'old' literature I dug up from the university library, but for me it was a very enriching period, both in professional as in private life.

For making this research possible and successful after all, I want to thank especially my promotor Jos van Hillegersberg, and daily supervisor Christiaan Katsma for their everlasting patience; it must have been a real endurance test every time I came up with radical new insights from some distant field of research, when my attitude concerning professional organizing was way too laid back, or when I needed some proper motivation. As a result of your positive approach I really enjoyed collaborating with the two of you; I especially bear to mind moments like the discussions about my research, the tense workshop sessions in Gelre, and Apple hate-it-or-love-it discussions.

I want to thank my committee members, prof. Brinkkemper, prof. Harmsen, prof. Nieuwenhuis, and prof. Wieringa for taking a critical perspective at my work; I'm honored that you have been willing to review and approve my work.

I want to thank all my colleagues and former colleagues in the IS&CM department: Fons, Ton, Elfi, Esther Jawai, Maria, Margreet, Esther Klaster, Hans, Lucas, Simon, Michel, Roland, Daniel, Jef, and

Romana. Furthermore all the other colleagues I've collaborated with during my work at the university. I also want to thank every person I worked with during my case studies at the MST and Gelre Hospitals.

Naturally I want to thank the group's secretaries; Marian, Gloria, and especially Elke for being so proactive and often a last resort to solve any kind of issue.

I especially want to thank my paranimfs with whom I shared a room for most of my time at the university; Björn: thanks for your clear view on the world and its associated value, apologies for the bad jokes about the Swedish bikini team: it's not funny, but being in the employee's quadrant is neither...; Chintan for being a non-stop stand-up comedian: I lost count of the many times you made me laugh so much my I suffered from abdominal strain the next day, thanks for the positive view you share on the world out there.

Naturally I want to thank family and friends; especially my parents for being so patient with me (although I think I'll never forget my mother being sort of disappointed when I announced in 2007 a four year career in academia: "why can't you get a *real* job?"...). You made me to the person I am now, and I'm ever grateful for that.

Diederik, May 2012, Den Haag.

## ii. Abstract

Professional service organizations are organizations predominantly employed with professionals; employees with specific and dedicated expertise in an area. IT support of the primary operations in this type of organizations is suboptimal. Methodological support of development and implementation of information systems in professional service organizations is not producing expected results, as for instance can be observed by a lacking business-IT alignment.

This research brings together ideas from situational method engineering and action design research to address this issue. From the field of situational method engineering, the concept of method chunk is introduced as a made-to-measure IS development methodology element. In this research method chunks (atomic methodology elements) are developed that address hiatus in the ISD methodology in two distinct organizational cases. Action design research combines insight from design science and action research to not only design a technology component, but also embed the designed artifact in the conditions of use.

The situational method engineering - action design research based approach produces five key contributions: (1) an analysis of information systems development in the professional service organizations domain; (2) custom tailored method chunks that address specific needs in an ISD process, (3) an method to develop method chunks in the professional service organizations domain (4) an adaptation of situation method engineering based on the action design research approach; and (5) an real-life application of the action design research approach.



### iii. Samenvatting

Professionele service organisaties zijn organisaties waar voornamelijk professionals werken; werknemers met specifieke en toegepaste kennis op een betaald vakgebied. IT ondersteuning van de primaire processen in dit type organisatie is vaak sub-optimaal. Methodologische ondersteuning van ontwikkeling en implementatie van informatie systemen in professionele service organisaties levert niet het gewenste resultaat, dit kan bijvoorbeeld worden waargenomen door een gebrekkige business-IT alignment.

In dit onderzoek worden ideeën van situationeel methode ontwerp (situational method engineering) en actie ontwerp onderzoek (action design research) met elkaar gecombineerd om de problematische aanpak voor informatie systeem ontwikkeling in dit bedrijfsdomein te adresseren. Vanuit het situationeel methode ontwerp-veld wordt het concept *method chunk* geïntroduceerd als een op maat gemaakt methode element voor informatie systeem ontwikkeling. In dit onderzoek worden method chunks (atomische methode elementen) ontwikkeld die hiaten adresseren in informatie systeem ontwikkelmethodes in twee verschillende organisatorische settingen. Actie ontwerp onderzoek combineert inzichten van ontwerp onderzoek (design science) en actie onderzoek (action research) om zo niet alleen een technologie component te ontwikkelen, maar ook de ontwikkelde component te embedden in het toepassingsdomein.

De situationeel methode ontwerp – actie ontwerp onderzoek gebaseerde aanpak produceert in dit onderzoek vijf contributies: (1) een analyse van informatie systeem ontwikkeling in het professionele service organisaties domein; (2) op maat ontwikkelde methode chunks die specifieke requirements in een informatie systeem ontwikkel proces adresseren; (3) een methode om methode chunks te ontwikkelen specifiek voor het professionele service organisatie domein; (4) een doorontwikkeling van situationeel methode ontwerp op basis van de actie ontwerp onderzoek aanpak; (5) een toepassing van actie ontwerp onderzoek in de praktijk.



## iv. Table of Contents

1	Introduction .....	1
1.1	Problems in the business domain .....	3
1.2	Solutions to the business problems.....	4
1.2.1	Past breakthroughs.....	6
1.2.2	Partial solutions .....	11
1.2.3	Promising solutions.....	16
1.3	Problem statement.....	21
1.4	Objectives.....	22
1.4.1	Contribution to body of knowledge.....	22
1.4.2	Contribution to practice .....	23
2	Research design .....	25
2.1	Design science.....	25
2.2	Action design research .....	26
2.2.1	ADR Stages.....	29
2.2.2	ADR process .....	32
2.3	Research approach .....	34
2.3.1	Project selection.....	35
2.3.2	Validity threats.....	35
2.3.3	Research process .....	36
2.3.4	Thesis structure .....	39
3	Problem formulation and initial design .....	41
3.1	Research methods .....	42
3.1.1	Method: Literature review .....	43
3.1.2	Method: Contingency theory.....	44
3.2	Factsheet .....	47
3.3	Literature review: approach.....	48
3.4	Literature: Information systems development.....	50
3.4.1	Information systems.....	52
3.4.2	Information system development .....	53

3.4.3	ISD Paradigm .....	54
3.4.4	ISD Approach .....	55
3.4.5	ISD Methodology .....	57
3.5	Literature: Professional service organizations .....	66
3.5.1	Professionals versus complex systems .....	68
3.5.2	PSO characteristics.....	70
3.6	IS Development in PS Organizations .....	79
3.7	Literature: Method engineering .....	81
3.7.1	Situational method composition.....	83
3.7.2	Method engineering meta-model .....	84
3.7.3	Method chunk and method fragment.....	86
3.7.4	Developing method fragments and chunks .....	88
3.7.5	Creating a modular methodology from fragments.....	90
3.8	IT artifact: Initial design.....	93
3.8.1	Literature review outcome.....	94
3.8.2	Domain specific approach to construct method chunks .....	95
3.8.3	Design principles .....	97
4	Building: Redesign IS in radiology department.....	101
4.1	Research methods .....	102
4.1.1	Method: Participation.....	104
4.1.2	Method: Behavioural observation: unstructured interviews	104
4.2	Factsheet .....	105
4.3	Building .....	107
4.3.1	Case description .....	108
4.3.2	Contingency variables: case analysis.....	111
4.3.3	Method chunk's scope.....	115
4.3.4	Selection, configuration and assembly.....	116
4.4	Intervention.....	135
4.5	Evaluation .....	137
5	Building: Redesign of IS in outpatient clinics.....	143
5.1.1	Method: Non-behavioural observation: record analysis .....	144



5.1.2	Method: Behavioural observation: workshop sessions .....	144
5.2	Factsheet .....	145
5.3	Building .....	148
5.3.1	Case description .....	149
5.3.2	Contingency variables: case analysis .....	151
5.3.3	Method chunk's scope .....	156
5.3.4	Selection, configuration, and assembly .....	157
5.4	Intervention .....	167
5.5	Evaluation .....	172
6	Reflection and learning .....	177
6.1.1	Method: Observational evaluation: workshop sessions .....	179
6.1.2	Method: Technology rules .....	179
6.2	Factsheet .....	180
6.3	Initial design .....	182
6.4	Evolution of initial design .....	183
6.4.1	Layer of abstraction .....	183
6.4.2	Structuredness of the method .....	184
6.4.3	End-user involvement .....	187
6.4.4	Paradigm alignment .....	189
6.4.5	Agility techniques .....	191
6.4.6	Regular feedback loops .....	192
6.4.7	Adaptive task definition .....	193
6.4.8	Multi-perspective integration .....	194
6.5	Improved design .....	195
6.5.1	Method chunk development process .....	198
6.5.2	Method chunk development products .....	201
6.5.3	Design principles .....	203
6.6	Applicability of ADR and SME .....	204
7	Formalization of learning .....	207
7.1.1	Method: Interpretivism to generalize .....	209
7.2	Factsheet .....	212
7.3	Reconnect to the underlying theories .....	213

7.3.1	ISD methodologies .....	214
7.3.2	PSO domain .....	216
7.3.3	Situational method engineering .....	220
7.4	Generalized approach .....	223
7.4.1	Method to develop method chunks .....	224
7.4.2	Method chunks for professional service organizations ISD .....	226
8	Conclusions .....	229
8.1	Review of the problem statement .....	229
8.2	Review on the research process .....	230
8.3	Key contributions .....	233
8.3.1	ISD in the PSO domain .....	233
8.3.2	Development of two method chunks .....	235
8.3.3	Method to develop method chunks .....	236
8.3.4	Situational method engineering based on ADR .....	238
8.3.5	Application of the ADR paradigm .....	239
8.4	Limitations: No silver bullet .....	241
8.5	Future research agenda .....	243
9	Appendices .....	245
9.1	Appendix A: Process description medical diagnostics, MST case .....	245
9.2	Appendix B: Process description outpatient clinics: Gelre case .....	252
10	References .....	255
11	About the Author .....	269

## v. List of Figures

Figure 1-1: Software properties that make development hard. ....	5
Figure 2-1: Stages and principles in action design research. Adapted from (Sein et al., 2010).....	29
Figure 2-2: Action design research: overview. Adapted from (Sein et al., 2010).....	31
Figure 2-3: ADR process flow. Adapted from (Sein et al., 2010). ....	34
Figure 2-4: Research process flow, adapted IT dominant ADR approach. ....	37
Figure 2-5: Meta-level process description ADR research. ....	38
Figure 3-1: Representation of Contingency Theory in IS Research. Adapted from (Weill and Olson, 1989). ....	45
Figure 3-2: Process-data diagram: Snapshot Problem formulation stage; input and output. ....	48
Figure 3-3: Structure of literature review. ....	50
Figure 3-4: Information systems development: concept map, source: (Hirschheim and Klein, 1989; livari et al., 1998; Kroenke, 2011).....	52
Figure 3-5: Paradigms in ISD, adapted from (Hirschheim and Klein, 1989). ....	54
Figure 3-6: ISD approaches, adapted from (livari et al., 1998).....	56
Figure 3-7: The role of methodology in ISD, adapted from (Lyytinen, 1987).....	58
Figure 3-8: Examples of ISD Methodologies in the four perspectives. ....	59
Figure 3-9: Evolution of ISD methodologies, adapted from (Avison and Fitzgerald 2006, Boehm 2006). ....	61
Figure 3-10: Diamond model for organizational characteristics, adapted from (Leavitt, 1978).....	68
Figure 3-11: Spatial model of effectiveness criteria, adapted from (Quinn and Rohrbaugh, 1983).....	76
Figure 3-12. Value shop model.....	77
Figure 3-13: Research positioning based on ISD paradigm, ISD approach, and ISD methodology. ....	80
Figure 3-14: Multi-level hierarchy of (situational) method engineering. Adapted from (Cossentino et al., 2006).....	83
Figure 3-15: Method engineering meta-model. Adapted from (Cossentino et al., 2006) .....	85
Figure 3-16: Method fragment meta-model. Adapted from(Cossentino et al., 2006).87	
Figure 3-17: Process model for method fragment / chunk construction. Adapted from (Ralyté, 2004). ....	89
Figure 3-18: Generic process model for SME. Adapted from (Ralyté, 2004; Ralyté et al., 2003).....	91

Figure 3-19: Process model Paradigm-based approach to SME. Adapted from (Ralyté et al., 2003).	92
Figure 3-20: Approach to construct method chunks; process model.	96
Figure 3-21: Outline of a method chunk description. Adapted from (Deneckere et al., 2008; Ralyté and Rolland, 2001a).	97
Figure 4-1: Process-data diagram: Snapshot first BI&E stage.	105
Figure 4-2: Radiology process (high level view).	111
Figure 4-3: Initial design method chunk's process flow.	116
Figure 4-4: Goal modeling method fragment - general configuration.	121
Figure 4-5: Collaborative design method fragment - general configuration.	121
Figure 4-6: Goal modeling - collaborative design process flow.	133
Figure 4-7: Method fragment description: goal modeling.	135
Figure 4-8: Goal modeling – collaborative design method chunk in use: Tropos goal model.	136
Figure 4-9: Updated approach to develop method chunks.	140
Figure 5-1: Process-data model: Snapshot second BI&E stage	145
Figure 5-2: Initial design method chunk's process flow.	157
Figure 5-3: Simulation method fragment - general configuration.	159
Figure 5-4: Collaborative design method fragment - general configuration.	160
Figure 5-5: Simulation and collaborative design process flow.	165
Figure 5-7: Method fragment description: simulation and collaborative design.	166
Figure 5-8: Simulation method fragment: Occupation rate waiting; morning only.	169
Figure 5-9: Simulation method fragment: Occupation rate waiting, afternoon only.	169
Figure 5-10: Simulation method fragment: Occupation examination room (1AA174) morning only.	170
Figure 5-11: Simulation method fragment: Occupation examination room (1AA174), afternoon only.	170
Figure 5-12: Simulation method fragment: Heat map ophthalmology outpatient clinic.	171
Figure 5-13: Simulation method fragment: 3D results from outpatient simulation (screenshots).	172
Figure 6-1: Process-data model: Snapshot Reflection and learning stage.	180
Figure 6-2: Method chunk development process model: update.	199
Figure 6-3: Method chunk description. Simulation - collaborative design example.	202
Figure 7-1: Abstraction layers in this research.	208
Figure 7-2: Process-data model: Snapshot Formalization of learning.	212
Figure 7-3: Summary of information system development.	214

Figure 7-4: SME process adapted to ADR. High-level view.....	222
Figure 7-5: Meta-model: method to construct method chunks.....	226
Figure 8-1: Relationship between development method and developed method chunks.....	233
Figure 8-2: SME and SME-ADR positioning.....	239
Figure 9-1: Overview process model HNP treatment. ....	245
Figure 9-2: Overview process model outpatient screening, HNP treatment .....	246
Figure 9-3. Medical diagnostics process, part of outpatient screening, part of HNP treatment.....	246
Figure 9-4. Radiology department process.....	247
Figure 9-5: Actors involved in the process.....	248
Figure 9-6: Conduct radiology scan process.....	248
Figure 9-7: Register patient process.....	249
Figure 9-8: Run scan process. ....	249
Figure 9-9: Finalize scan process.....	250
Figure 9-10: Process scan results process.....	250
Figure 9-11: Send results process. ....	251
Figure 9-12: Total number of patients for the outpatient clinics. ....	252
Figure 9-13: Number of patients for the outpatient clinics; 4 weeks.....	253
Figure 9-14: Number of patients present in the hospital at any given time. ....	253
Figure 9-15: Overview process flow ophthalmology.....	254
Figure 9-16: Patient perspective on ophthalmology process.....	254



## vi. List of Tables

Table 2-1: Roles and activities of the ADR team. Adapted from (Sein et al., 2010).	33
Table 2-2: Thesis structure; chapters, topics, goals and deliverables. ....	40
Table 3-1: Applied methods in the problem formulation stage. ....	42
Table 3-2: Summary PSO characteristics. ....	71
Table 3-3: Stages in the initial design to develop method chunks.....	96
Table 3-4: Rules for constructing method fragments and chunks. Adapted from (Brinkkemper et al., 1998). ....	98
Table 3-5: Design principles for a method to construct method chunks. ....	99
Table 4-1: Applied methods in the building, intervention, and evaluation stage. ...	102
Table 4-2: Requirements based on MST case analysis. ....	115
Table 4-3: Top-down and bottom-up goal modeling approaches compared.....	129
Table 4-4: Validation of developed method chunk against original requirements....	138
Table 5-1: Applied methods in the building, intervention, and evaluation stage. ...	144
Table 5-2: Requirements based on Gelre case analysis.....	156
Table 6-1: Applied methods in the reflection and learning stage. ....	178
Table 6-2: Matching abstraction level to audience capabilities.....	184
Table 6-3: Effects of evolutionary aspects of the initial design on the updated design. ....	197
Table 6-4: Revised design principles for a method to construct method chunks. ....	204
Table 7-1: Applied methods in the formalization and learning stage.....	208
Table 7-2: Diamond model: PSO characterization relevant for ISD. ....	216
Table 8-1: Summary of the ADR process to develop a method to construct method chunks.....	232
Table 9-1: Outpatient clinics involved in ISD project. ....	252





# 1 Introduction

From the entire organizational spectrum, we distinguish two characteristic organizational types; manufacturing organizations that mostly produce products or services applying sequential, well-structured processes, and professional service organizations (PSO) that mostly produce products or services applying non-sequential, ill-structured processes. Typical manufacturing organizations are logistics organizations, mass-producers, and mass-service organizations. Characteristic PSOs are health care organizations, governmental and judicial organizations, educational institutions, IT services departments of large enterprises, IT services organizations, management consulting organizations, engineering firms and advertising organizations. This research focuses on PSOs.

A PSO is characterized by its high levels of process flexibility, dynamic structures, and ad-hoc workflows. A PSO is structured around the work of highly trained, and highly specialized individuals which work on dedicated tasks. Organizations or organizational parts that are classified as a PSO are dominated by professionals; employees with a very high degree of specialization. In contrast, most employees in manufacturing organizations are less specialized, working on more general tasks.

Organizational and management research has put a strong focus on the (improvement of) the workforce's productivity and organizational performance (Davenport et al., 2002; Drucker, 1999; Mintzberg, 1998). Since Taylor first analyzed the manual workers performance early 20<sup>th</sup> century, the manual workers' compound productivity has risen fifty fold (Drucker, 1999). Productivity of professional employees however, is stagnating relative to employees in other organizations. Since 1945 the productivity of professional organizations' productivity has not even doubled (Brynjolfsson, 1993).

Manual workers' productivity is relatively easily improved by applying lessons from Taylor's scientific management, but professional employees' productivity is harder to influence. Professionals don't need in house procedures or time-study analysts to tell them how to do their jobs (Mintzberg, 1998)

Davenport describes three determinants that influence productivity and performance in professional organizations: management and organization, information systems, and workplace design (Davenport et al., 2002). In this research we focus on the influence of information systems on professional employees' productivity. According to Davenport, "*many companies are toiling to create the functional equivalent of a software Swiss army knife to integrate communication, collaboration, knowledge management, virtual teaming, e-mail and instant messaging. Yet few (if any) have figured out how to get knowledge workers [or professionals] to actually use these tools*" (Davenport et al., 2002).

Information systems (IS) are used in organizations to automate and support tasks, processes and workflows. The basic aim of an information system is to improve the productivity of individual employees and the overall organizational performance. The design and implementation of information systems in every type of organization up till today have always been complex processes, which often have proven to be rather problematic. In the recent history, many initiatives have been employed to cope with this problematic setting. Frequently, initiatives that aim to overcome these problems focus on the methodological standardization of information systems development. Methodological standardization has proven to be a successful approach to attain manageable processes with effective outcomes. This research addresses methods for information systems development (ISD). In general, ISD is a well-researched field.

Information systems development can be defined as "*a change process with respect to object systems in a set of environments by a development group using tools and an organized collection of*

*techniques collectively referred to as a method to achieve or maintain some objectives*" (Lyytinen, 1987). Information systems include both the organizational as well as the computer-supported part. Information systems development therefore both includes the shaping of the organizational and the computerized parts of a system. In this research we predominantly analyze the organizational side of information systems development.

## **1.1 Problems in the business domain**

Many methods that have been developed to improve organizational performance by means of the implementation of new information systems, have proven to be successful in manufacturing organizations and other organizations that are characterized by sequential processes. It has been demonstrated that successfully designed and implemented information systems combined with process reengineering can provide a significant improvement in organizational performance (Brynjolfsson, 1993). By applying standardized methods, organizations have benefitted from successful information systems, which in turn led to a better organizational performance.

However, this approach has been failing for many years now for PSOs, which can be a key explanation for the PSO stagnation in performance (Brynjolfsson, 1993; Carr, 2003). Despite the numerous information systems methods that are designed to improve PSOs' performance, up till this date, performance in PSOs declines relatively to manufacturing organizations (Drucker, 1999). Because of the successful adoption of information systems in other type of organizations, it is remarkable that the stagnating performance of PSOs simultaneously occurred with the introduction of huge information technology (IT) investments for information systems (Brynjolfsson, 1993): in manufacturing organizations, information systems boost productivity significantly, but research has demonstrated that the introduction of information systems in PSOs

does not improve productivity or organizational performance (Carr, 2003; Hammer, 1990).

It seems that PSOs do not necessarily benefit from newly implemented information systems. Although the public opinion is that a satisfactory, highly used, successful, and effective information system leads to improved organizational performance, the positive link between the variables IT investments, IT performance, and performance in professional organizations is very often assumed, but not actually demonstrated (Weill and Olson, 1989).

One of the reasons manufacturing organizations have benefitted from the implementation of information systems and the standardization of processes is that the industry best-practices are built-in unavoidably, because main information systems in manufacturing organization are based on the Enterprise Resource Planning (ERP) architecture for primary process support. The ERP best practices are transferred to the IS adopting organization. PSOs, however, have not been able to reap the benefits from the adoption of standardized processes in newly acquired information systems. The impact of IT investments on performance in professional organizations is demonstrated to be not significant on average, but to be associated with both very high and low performers (Cron Marion and William, 1983). *“This finding has engendered the hypothesis that IT tends to reinforce existing management approaches, helping well organized firms to succeed, but only further confusing managers who have not properly structured production in the first place”* (Brynjolfsson, 1993).

## 1.2 Solutions to the business problems

Over the course of past decades many advances have been made to improve the development of information systems. Dedicated methodologies for information systems development in (professional service) organizations solve some of the difficulties in the business domain that are described in the previous paragraphs. However, no *silver bullet* has been developed yet.

In this paragraph an overview of approaches aimed at a structured information systems development is presented. These methods are grouped in three classes: past breakthroughs, partial solutions, and promising solutions, analogous to Brooks' seminal Silver Bullet article (Brooks, 1987). In Brooks' original article an overview is presented of means to improve the development of IS at the time of writing. In the sections of this paragraph, an updated view is presented.

In this specific article Brooks refers to four properties of software systems that make the development of software hard (see Figure 1-1): *complexity* (information systems are amongst the most complex man-made structures), *conformity* (information systems must conform to unique organizational characteristics), *changeability* (information systems are continuously subject to change; not only when in operations but also during the development process itself), and *invisibility* (means to visualize information systems and make them visibly and easily available).

The specific issues with information systems development in the professional service organization domain are discussed in chapter 3, where a thorough literature review is conducted on the topics information systems development and professional organizations.

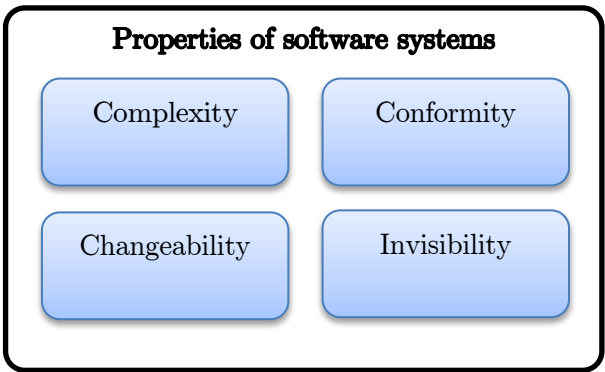


Figure 1-1: Software properties that make development hard.

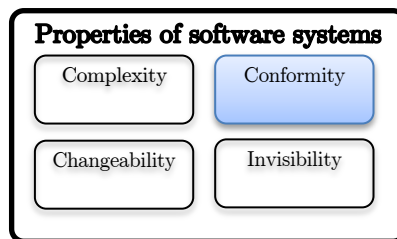
In the next sections methods for information systems development are discussed that address one or more properties of software systems, as depicted in Figure 1-1. Past breakthroughs address some but not the essential difficulties of IS development; partial solutions (*hopes for the silver*) address some of the essential difficulties, but still fail to address every aspect; and promising solutions provide expectancies about their capabilities to address the complete set of difficulties in IS development, but still did not yet realize these expectancies. The selection of methods that are listed in the next sections is based on discussions and further literature mining. The presented list of method in this chapter is merely an overview to position and motivate this PhD research, and is far from exhaustive.

### 1.2.1 *Past breakthroughs*

In this section the past breakthrough that managed to address some of the problems in professional organizations, but never delivered a solution that fully covered these problems, are presented. The approach is summarized and the limited success in addressing the key issues at ISD in PSOs is explained.

#### *End user development*

End-user development (EUD) is defined as computer applications that are developed by the people that have direct need for them. The concept of end-user development varies from customization to component configuration and



(script based) programming. Office software like spreadsheets provides the end user with the ability to customize the software to individual needs. An end-user can configure (web) components via an array of scripting languages, whereas complex solutions can be programmed with programming languages such as Java or C++. Next to the decrease of development time for IS, EUD is aimed at bringing the end-users closer to the development project; creating a

better fit between requirements and end-result. Research has demonstrated that end-users hardly ever resort to commercial of the shelf software to solve their problems, although these COTS solutions often offer better functionality than the end-user solutions (Fischer et al., 2004). Development of applications by end-users is a particularly widespread phenomenon in two computing application fields: scientific / technical computing, where information technology is directly placed in the hands of researchers or engineers, and business / commercial computing where this information technology is directly placed in the hands of clerks and managers (Brancheau and Brown, 1993). These groups basically compromise the knowledge workers and technologists that are the core employees of PSOs.

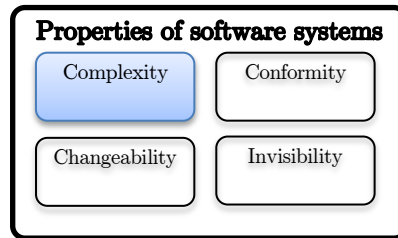
End user development can be defined as “*the adoption and use of information technology by personnel outside the information systems department to develop software applications in support of organizational tasks*” (Brancheau and Brown, 1993). An example of end user development is the use of spreadsheet programs when they are used to develop large applications. Research demonstrates that the use of spreadsheets as foundation for financial reporting systems is common practice in many small and midsize but also large organizations. Since it is known that errors occur in spreadsheets in a few percent of all cells, the information reliability of large systems based on spreadsheets is troublesome (Panko, 1998).

As might be expected, there is a delicate balance between the scope of the end-user development approach and the cost of learning. Basically, the more extended the EUD environment is, the better it is to cope with a broad variety of requirements, but the higher the costs to adopt and learn the approach. In the past, most computing technologies were based on a centralized architecture. But, since end-user development is based on a decentralized architecture, the main problems with end-user development are monitoring, quality assurance, and control (Sutcliffe and Mehandjiev, 2004). End-user development has been a management challenge for years. The main issues are the motivation of the users to adopt a certain technology, controlling the development to minimize risks, creating maintainable

software, and eliminating inaccurate and contradictory information. *The tension between quality assurance and design freedom will always color the advances* (Sutcliffe and Mehandjiev, 2004).

### ***Packaged application software***

Comprehensive packaged software solutions seek to integrate the complete range of a business's processes and functions in order to present a holistic view of the business from a single information and IT architecture (Klaus et al.,



2000; Kumar and Van Hillegersberg, 2000). Usually these packaged software systems are called enterprise resource planning (ERP) systems. Most very large organizations worldwide have adopted ERP. ERP systems are highly complex information systems. In ERP software packages business processes across organizational functions and locations are integrated and managed. However the implementation of these systems is a difficult and high cost proposition that is not without risk. Furthermore the design of ERP software is standardized to fit industry *best-practices* (Umble et al., 2003). ERP software is built component based, with modules that support for instance procurement, material management production, logistics, maintenance, sales, distribution, or strategic planning. Although the ERP package components are organized at the highest level in these different functional modules, they all follow a process-oriented view of the organization. ERP software is a standard software package, targeted at an anonymous market. Therefore, during the process of system development, the package must be tailored to the specific requirements of individual organizations (Klaus et al., 2000).

A specific modular add-on to ERP systems for professional service organizations is Professional Service Automation (PSA) software, also termed Service Process Optimization (SPO) software (Wang and Swanson, 2007).



*Professional Services Automation is the term used to describe a new family of applications designed for professional services organizations that enable service professionals to become more productive and profitable by increasing their efficiency on the job through increased employee utilization and integrated knowledge management (Hofferberth, 2002).*

PSA software is commonly produced by either ERP vendors, positioning it as a modular add-on to their enterprise system offering, or by pure-play vendors. Most PSA products have a modular architecture and modules typically include planning and scheduling, project management, performance management, and billing.

PSA software until now never became mainstream software for organizations in the PSO domain. There are several reasons for this. PSA software is often adopted on a modular scale; not the complete suite is implemented, but just some functionality. Other functionality is often handled by (in house developed) existing software (Hofferberth, 2002). This is also the reason why many IT service providing organization never adopted PSA software; functionality was already developed in-house – although the in-house developed software often lacks integration. Furthermore, critics claim that integrated PSA solutions cannot be customized to meet the requirements of the diverse processes in a professional service organization and still be scalable and robust (Wang and Swanson, 2007).

A common problem when adopting package software has been the issue of *misfits*; the gaps between the functionality offered by the package and that required by the adopting organization. Because of the misfits, organizations have to choose between adopting new functionality, inventing workarounds or customizing the package (Soh et al., 2000): *the problem is exacerbated because ERP implementation is more complex due to cross-module integration, data standardization, adoption of the underlying business model [...], compressed implementation schedule, and the involvement of a large number of stakeholders.*

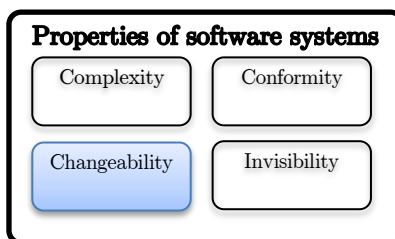
Another problem with packaged application software is the complexity of these systems. The idea to integrate the support of all business functions leads to massive programs with millions of lines of code, thousands of installation options and numerous interrelated pieces. The result is massive complexity (Rettig, 2007). It is known that, as a problem gets more complex, the software solving that problem gets more complex too. An estimation is that for every 25% increase in complexity of the task to be automated, the software complexity rises by 100% (Glass, 2003). As a result the concept of a single monolithic system failed for many companies. The implementation of packaged application software may lead to a spaghetti like infrastructure comprising many *legacy systems*, and autonomous systems acquired via independent purchases or organizational mergers and acquisitions (Rettig, 2007).

Foremost, ERP supports recurring business processes like procurement or logistics and is not focused on less structured, irregular processes like marketing, product development, or project management (Klaus et al., 2000). This is the main problem for professional organizations. Since most of their core processes are not repetitive, sequential, standardized processes, the focus of the functionality of ERP software is not on the core processes' support, but on the supporting processes in the organization.

### ***Software mass customization***

Mass customization relates to the ability to provide individually designed products and services to every customer through high process flexibility and integration (Da Silveira et al., 2001). Analogous to the mass customization method

used in manufacturing, software mass customization takes advantage of a collection of parts capable of being automatically composed can configured in different ways to create different products. Each product in the product line is manufactured by an automated



production facility capable of composing and configuring the parts based on an abstract and formal characterization of the product's desired feature profile (Krueger, 2006). Software mass customization relies to a high degree on automation, which should make the method easier to adopt and sustain. Products are developed with less effort due to reuse of the core assets.

Using the mass customization methodology, software configurators enable development via automated product instantiations (Krueger, 2006). To automatically create product instances, these configurators take two types of inputs: core assets and product models. The core assets are the common and varying software assets for the product line, such as requirements, architecture, design, source code, test cases, and product documentation. The product models are abstractions that characterize the different product instances in the product line (Czarnecki and Eisenecker, 2000). With the software configurators, all developed software exists within the consolidated collection of core assets, making everything available for reuse (Krueger, 2006).

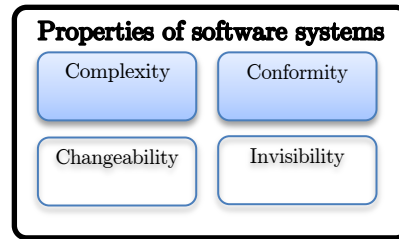
Although the potential benefits are order-of-magnitude improvements in software engineering performance, the up-front costs, level of effort, assumed risk, and latency required to create a barrier to adopt software mass customization by many organizations that could benefit (Krueger, 2005).

### ***1.2.2 Partial solutions***

Every past breakthrough discussed in the above section solved some difficulty in dealing with the complex environment of information systems in professional organizations. In the following section methodological developments are considered that are most often advanced as potential partial solutions. Each development is discussed over what problems they address and the remaining difficulties.

### *Project management methodologies*

A project in an organization is a temporary endeavor undertaken to create a unique product, service or result. In essence a project is a temporary organization within an organization (Shenhar, 2001).



Implementing an information system can therefore be considered a project. To bring a project to its successful end, the integration of many management functions such as controlling, communication, and costs management is required. For many of those functions additional project management techniques are developed (Shenhar and Dvir, 1996). Project management is the discipline of planning, organizing, and managing resources to bring about the successful completion of specific project goals and objectives. In every project there are certain constraints like time, resources and budget. There are many different methods for managing projects in organizations; from the traditional PERT (program evaluation and review technique), to the modern PRINCE2 (projects in controlled environments 2), or PMBoK (project management body of knowledge) methodology.

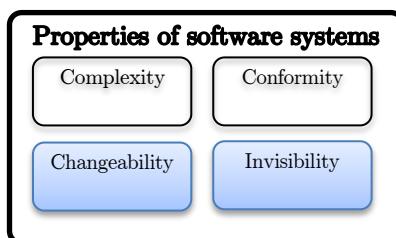
Mainstream project management methods like PRINCE2 can be applied to a very broad array of industries including the implementation of information systems, and are very flexible and scalable. Characteristic features of these approaches are that the managed projects are divided into phases starting with initiation and ending with closing the project. Since these project management methods are in theory very flexible and scalable, the exact configuration is determined by the project manager.

Despite these rich methodologies, failure of projects is very common. Literature often ignored the fact that not all projects are the same and there is no universal set of managerial characteristics to a project (Shenhar, 2001); *“Indeed, several authors have recently*

*expressed disappointment in the universal ‘one-size-fits-all’ idea*”. Other criticisms include the high level of bureaucracy, the very formal approach to manage relationships, and the complex planning effort (Crawford et al., 2006). The very structured approach in project management provides a ground for conflicts when applied in the ill-structured professional organizations’ environment.

### ***Unified modeling language***

Unified Modeling Language (UML) is one of the standards for object-oriented analysis and design of information systems (Agarwal and Sinha, 2003). UML can be applied as modeling technique in many phases of the development



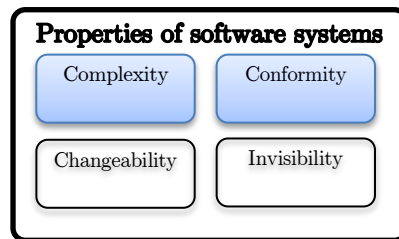
process. In essence is UML itself not a modeling method, only a means to model and graphically represent various layers of information systems. The main feature of UML is the conceptual notation of information systems (Dobing and Parsons, 2006). UML enables a software engineer to model both static as well as dynamic aspects. Static aspects can be modeled with *class diagrams*, *object diagrams*, *component diagrams*, and *deployment diagrams*. For dynamic processes UML provides *use case diagrams*, *collaboration diagrams*, *sequence diagrams*, *activity diagrams*, and *state diagrams*. UML can be used for visualizing, specifying, constructing, and documenting the artifacts of software systems (Booch et al., 2005). UML is not only useful for specifying systems requirements and capturing design decisions, it is also useful for promoting communication among key individuals involved in an information systems design (Agarwal and Sinha, 2003).

The Object management group (OMG) adopted UML as its standard modeling language in 1997, by which UML has been formally recognized as an international standard. The major benefits of the recognition of UML as an international standard (the wide recognition and acceptance), conflict with the major disadvantages

that come with a standardization process. From a business perspective, the timescales of standards usually conflict with the competitive needs to use the latest technology as early as possible. The technical downside of standardization is the *design by committee* process, which implies major tradeoffs (Kobryn, 1999). UML diagrams do not have a positive perception of usability as a means of ease-of-use. Systems developers need to be trained to be able to use UML properly and reap the advantages. Any major systems development effort involves several stakeholders in addition to systems professionals. Though UML diagrams promote communication among those stakeholders by providing a standard, method-independent notation, non-expert users first need to be trained in reading the notation (Agarwal and Sinha, 2003). UML is criticized for delivering insufficient value to justify the costs. Since UML is a very rich and complex, but also rigid notation language, training users takes a lot of effort (Moody and Van Hillegersberg, 2009). Simply put UML may be too complex, and more extensive educational programs are needed, both to increase the number of analysts familiar with UML and provide ongoing support to help them make fuller use of its capabilities (Dobing and Parsons, 2006).

### ***Software Process Improvement***

Software process improvement (SPI) aims at integrating a broad array of models, standards and methodologies that can help an organization improve their processes. Since the integration of separated approaches is very hard, SPI



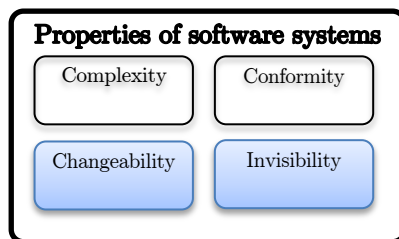
eliminates the barriers that are in the way of optimizing the total process flow that meet the business goals of an organization. A established method in the SPI field is Capability maturity model integration (CMMI). According to the authors CMMI is based on best-practices (Team, 2006). Initially CMMI originated in software engineering, but the method is adapted to be used in various business environments.

CMMI describe practices and goals for information system development areas, and has a framework for measuring the compliance of organizations with the goals and practices in these process areas (Staples et al., 2007).

There are a substantial amount of claims regarding the high benefits of CMMI, however an extensive literature review reveals that there has been almost no published evidence about the experiences of organizations who adopt CMMI (Staples and Niazi, 2008). As research demonstrates does CMMI not fit every organizational environment. Especially small and mid-sized organizations cannot benefit from the CMMI method, because of high costs and specialized skills required (Staples et al., 2007).

***Participative systems design***

A main problem with many information systems development methods is the comprehensibility of the requirement analyses, models and system specifications for end users. Users are not software developers, and are



often untrained to analyze systems design. However, contemporary methods result most often in a mismatch between the requirements specification and the concrete users' desires and demands for the new system. An alternative approach to systems development is participatory or participative design (Nielsen, 2002). This method refers to the handling of responsibilities for design and means of introduction of a new system to a group of workers that must use the system. Participative design is based on the belief that individuals should have a say in their destinies, therefore the interest of the user is protected by this approach. This method allows individuals to redesign their jobs and working environment. Because the execution of activities is ultimately controlled by the ones who perform them, participative design can lead to higher compliance with the system. Furthermore, participative design can improve the

motivation of employees in actually using the system. Therefore the participative design method can potentially create an optimal information system – organization fit (Hirschheim, 1983). Participation requires active user involvement: *user participation does not mean interviewing a sample of potential users or getting them to rubber stamp a set of system specifications. It is, rather, the active involvement of users in the creative process we call design* (Greenbaum and Kyng, 1991). Participative design assumes that failure in contemporary software development projects is partially analytics, but foremost because of the failure to understand the nature of users' activities and needs (Shapiro, 2005).

Participative design can offer many advantages in information systems development projects: for example it can create a better understanding of goals, formulation of needs, design of coherent visions for change, combining business-oriented and socially sensitive approaches, initiating participation and partnerships with different stakeholders, establishing mutual learning processes with users from the work domains, conducting interactive experiments aiming at organizational change, managing stepwise implementation based on comprehensive evaluations, and providing a large toolbox of different practical techniques (Simonsen and Hertzum, 2008).

### ***1.2.3 Promising solutions***

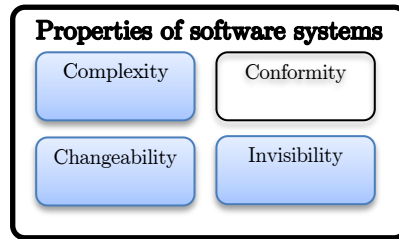
Even though no methodological breakthrough delivered some kind of systematic solution to the identified problems in the professional organizational business domain, we have identified some promising solutions that might fill the gap. The solutions we discuss below conceptually address the design and implementation problems of information systems in professional organizations.

#### ***Business process mining***

Business process mining techniques are most often applied when no formal description for a business process can be obtained via other means. Process mining aims at extracting information from event logs to capture the business process as it is being executed. Instead



of focusing on the realization of IT support (as for instance packaged application software does), the process mining method aims at monitoring the operational business process (van der Aalst and Weijters, 2004). Basically the business



events are recorded by contemporary information systems in so-called event logs. Business process mining takes these logs to discover process, control, data, organizational, and social structures (Van Der Aalst et al., 2007).

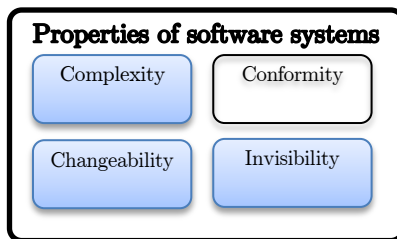
Most enterprise information systems store relevant events in some structured form in log files, for instance the start and completion of activities, process transactions, or customer interactions. Process mining aims at the automatic construction of models explaining the behavior observed in these logs. By applying the process mining approach to a log file, a process model can be expressed as a Petri net. It is assumed that the resulting process model resembles the reality to a large extent. These models can be used to analyze, diagnose and improve the current processes in an organization.

In an overview paper, van der Aalst addresses the main challenges in process mining (van der Aalst and Weijters, 2004). The main problems with process modeling are the inability to assure that the resulting process map reflects every aspect of the process. Some tasks are *hidden* in the system, tasks can be duplicated, and loops can be included. In addition, it is very hard to mine different perspectives of the same process. Most often only the control perspective is included in the final result. Another concern is the inability of many process mining techniques to deal with process noise and incompleteness. The resulting process map in both cases is rather hard to comprehend. Yet another concern is the problems that occur when process data has to be mined from heterogeneous sources. As discussed before, most enterprise systems are not based on one uniform software supplier, and all kinds of external systems and

legacy systems are attached to the enterprise system. Mining a process that travels through multiple heterogeneous systems can be very time consuming. The main concern however is the challenging situation when a process is finally mined: how should the results be interpreted, visualized, and analyzed in such way that they are easy to understand? There are relatively few or no methods developed to facilitate these issues.

### *Agile software development*

Agile development methods are a reaction to the traditional ways of developing information systems like the waterfall approach. Since the traditional methods are documentation driven, heavyweight software



development processes, the focus in these methods is on elicitation of a complete set of requirements, followed by architectural and high level design, development, and inspection (Cohen et al., 2004). According to the initiators of agile development, traditional approaches assume that developers could anticipate the complete set of requirements early and reduce cost by eliminating change (Highsmith and Cockburn, 2002). Agile software development methods are the lightweight alternatives to the traditional heavyweight methods. Agile methods are designed to deal with rapidly changing market forces, systems requirements, implementation technology, and project staff (Cockburn and Highsmith, 2002). This approach implements the idea that if change in projects is made unwanted or impossible, the project will be unresponsive to business conditions. Agile methods are aimed at the ability to implement ever changing business conditions rapidly in the software development process. Most agile methods promote development of teamwork, collaboration and process adaptability throughout the project life cycle. There are two concepts underlying agile development: working code is a measure for project quality, and people working together like a well-oiled machine are most effective.

Although the capability of agile methods to adapt to changing project requirements is most often mentioned as the main asset of these methods, critics claim that over responding to change can be a source of many software disasters. On top of that, some critics claim that agile software development mostly resembles the *hackers method* of irresponsibly throwing code together with no regard for engineering discipline (Boehm, 2002). According to Rakitin this approach of responding to change over following a plan turns into chaos generators (Rakitin, 2001).

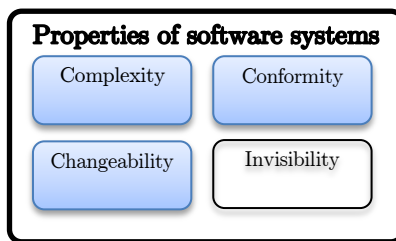
Agile development methods include Scrum, Extreme programming, Feature-driven development, Adaptive software development, and Dynamic systems development method.

### ***Model driven engineering***

The model Driven Development (MDE) is a software development methodology aimed at providing a set of guidelines for the structuring of specifications. These specifications are expressed as

models in an abstract model language. A specific (trademarked) implementation of MDE is Model Driven Architecture (MDA). Since MDA is developed by the Object Management Group (OMG), which also developed the UML standard, the preferred model language used to express the specifications for MDA is UML (Mellor et al., 2002). In essence, the models derived from the business requirements are fed into platform definition model to create platform specific models. Eventually these models are run as computer code using a higher level programming language.

The aim of MDE is to separate design from architecture. This approach allows information systems' developers to decouple the technologies used to realize the design from the design itself. If either



one advances, the information systems' developers can easily choose from the best fitting methods for design and architecture.

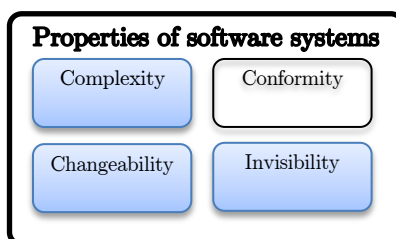
There are many software tools available for MDE development. As such there are dedicated tools for the creating of models, the analysis of the models, the improvements on the models, and the testing and simulation of the models.

Although MDE has been described as very promising; the method never really took off as proven approach. The main concern with MDE is the standards it is built on, for instance UML, which is hard to comprehend for non-experts. The aim of MDE is to make the development and implementation of information systems pragmatic and easy to comprehend. However, highly specialized experts are required to model, analyze, improve, and test the models for correctness.

### ***Simulation / Gaming***

Simulation is extensively discussed in literature. Simulation models are often used as a method in problem solving and decision making (Sargent, 2005). Games and simulations can give greater insight into the business

problems that are the root cause for the development of an information system. A developer is enabled to improve the requirements and design by dealing with multiple realities and can look for solutions to complex problems without destroying their variety, and test alternative courses of action (Szmankiewicz et al., 1988). Simulations can be used to explore and gain new insights into new processes, and estimate the performance of systems too complex for analytical solutions (Strogatz, 2007). There is a wide variety of simulation and gaming applications, ranging from management games to complex computer simulations. The basic aim of every type



of simulation is to analyze future process configurations. The results of each simulation run can be used as input for further development.

There is a fundamental difference between modeling and simulation; whereas the predictive value of models is limited by the scope of the design, simulations can demonstrate behavior that extends the initial design. In simulations only the input and process flows are defined, the behavior is generated during the simulation itself. In models this behavior is hard coded in the model.

In modeling a real world problem there is always a tradeoff between many types of validity, amongst others the internal and external validity, the educational validity, and representational validity (Tsjernikova, 2009). Educational validity means the level of complexity and to what extent the model still can be understood. Representational validity implies the actual model similarity with the real world. Simulations can be: *philosophical exploration of the logical consequences of a set of assumptions without any necessary regard for the real-world accuracy or usefulness of the assumptions* (Meadows, 2001).

### 1.3 Problem statement

The methods and approaches discussed so far address some characteristic problems in information systems development. However there is no *silver bullet* (Brooks, 1987) amongst them. Therefore the lack of methodological support for information system design and implementation is identified as the main research problem. In this research we further specify on the professional service organizations domain, since the methods available that are (partially) successful in other organizational types, add relatively little value in the professional service organization domain. We propose that the causes of the problematic operational performance lay in the PSO's characteristics and the knowledge workers' unique approach to their work. Information systems in these organizations are not aligned and correctly implemented to deal with the professional organizations characteristics. Methods for information

systems design and implementation that are successfully applied in manufacturing organizations do not provide the desired results.

Problem statement:

---

*What constitutes a situational approach to develop information systems for professional service organizations?*

The aim of this research is to provide an approach to develop information systems development methods that address the specific professional organizations domain problems. The term situational is introduced here to exemplify the need for an information system development method that addresses the specific characteristics of professional service organizations.

## 1.4 Objectives

Based on the above formulated problem statement several research objectives are formulated:

### *Research objectives*

---

- Characterize the professional service organizations domain
- Identify the information systems development problems for the professional service organization domain
- Design a configurable approach to develop IS for the PSO domain.

#### *1.4.1 Contribution to body of knowledge*

The contribution in this research is the proof that professional organizations can benefit from information systems and improve their performance. Drawing from the available literature in the body of knowledge, a structured approach for engineering suitable methods that satisfy the business needs and specific context of PSOs is engineered in iterative cycles.

### *1.4.2 Contribution to practice*

This research focuses on the design and implementation of information systems in professional organizations, health care in particular. The goal of this design research is to develop an approach with which methods can be engineered that provide professional organizations with the means to successfully design and implement information systems. With this approach professional organizations can reap the benefits of information systems and stimulate organizational performance.





## 2 Research design

Research in the information systems field is currently dominated by two streams: behavioral research, and design science research. This PhD research is an application of *action design research*; a specific branch of design science research. In the following sections the essence of design science theory and action design research are discussed. Next to that, the selected projects in which this PhD research is applied are presented.

### 2.1 Design science

In the IS field, there are two research paradigms; the behavioral, truth finding paradigm, and the design science, utility aimed paradigm (Hevner et al., 2004). This PhD research is firmly grounded as design science research, since the aim is to engineer methods for ISD (Wieringa, 2009). The goal of any study in the design science paradigm is to improve real life situations, and therefore gain some utility (March and Smith, 1995). In this research the aim is to develop an approach for ISD in professional service organizations, specifically in the early phases of development. Design science always results in a designed artifact; the solution to the problem in a real environment. Whenever in this research is mentioned to the (IT-) artifact, an approach to develop IS for PSOs is referred to.

Hevner et al. present a foundation and approach for conducting design science research in the IS field (Hevner et al., 2004). According to these authors, design science research should include both the relevance from the research environment, and the rigor from the body of knowledge. In the design science paradigm, the product of the research is an artifact that can be evaluated to test its suitability. Information systems' artifacts are innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information

systems can be effectively accomplished (Denning, 1997). The design artifacts in design science studies are not necessarily the ultimate solutions – a designed optimal configuration is the best solution only until a better one is designed. Therefore, only the relative utility can be shown. Hence the design science approach in this research is very pragmatic. The goal of this research therefore is to satisfy the research objective and show the solution's utility over the current configuration.

This research applies a variation of design science; the action design research approach is applied for designing a solution to the problem statement (Cole et al., 2005; Sein et al., 2010).

## 2.2 Action design research

Action research design (ADR) is a synthesis of two established research approaches; the design research approach (as discussed in the previous section); and action research, which aims at learning from organizational intervention (Baskerville and Wood-Harper, 1998). ADR is a research methodology for generating design knowledge through building and evaluating IT artifacts (Sein et al., 2010). ADR is a research approach developed to help researchers to both make scientific contributions and to assist in solving current and anticipated problems of practitioners (Cole et al., 2005; Iivari, 2007; Sein et al., 2010).

IT artifacts developed in the design science paradigm, can be first designed and developed and second be evaluated for usefulness in an organizational environment. The addition of ADR to design science is that IT artifacts in the action design research paradigm require not only the design of a technology component but also action that embeds the artifacts in the conditions of use. This approach results “ *in a form of inquiry that focuses on the building, intervention, and evaluation of an artifact that reflects not only the theoretical precursors and intent of the researchers but also the influence of users and ongoing use in context*” (Sein et al., 2010). The main extension of ADR to design research is that the IT artifact design,

build, and evaluation cycles are not executed in sequence, but occur reciprocally, see Figure 2-1. Since this research is based on ADR, the research process and the structure of this thesis adapts the ADR scheme depicted in this figure.

The aim of action design research is twofold; “(1) *addressing a problem situation encountered in a specific organizational setting by intervening and evaluation; and (2) construction and evaluating an IT artifact that addresses the class of problems typified by the encountered situation*” (Sein et al., 2010). To paraphrase the first chapter; the research statement of this research is to develop an appropriate approach to deal with information systems development in professional service organizations. The so-called *IT artifact* in Seins definition is operationalized in this research by a *configurable method to develop ISD methodologies*, applied in the *PSO* problem domain. Whenever the term *IT artifact* is mentioned in this research, the research objective *approach to improve ISD methodologies* is referred.

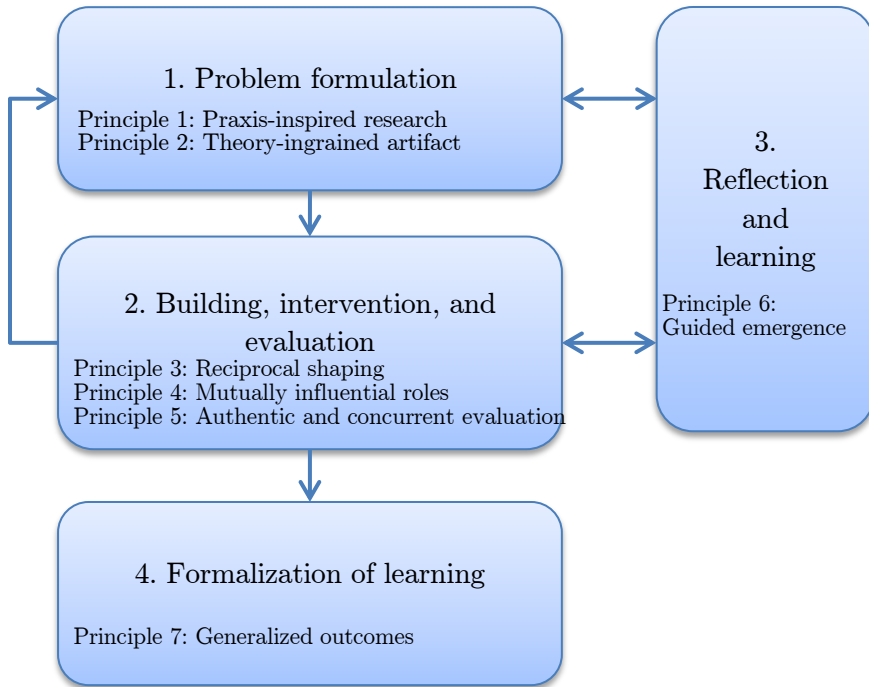
In many ways ADR is quite similar to technical action research (TAR), as is described in (Wieringa and Morali, 2012). TAR is a particular kind of action case study in which a researcher tests a technique in a case by using it in practice. Main difference with the application of ADR in this research and TAR as described by Wieringa, is that in this research the IT-artifact is not developed yet before it will be applied in the case studies; the IT-artifact is developed and further improved during the intervention in the organizations. Therefore the outcome of this research is not an evaluation of the IT-artifact, but the IT-artifact itself.

ADR is also comparable to Canonical Action Research (CAR). Where the original Action Research (AR) is criticized for its lack of methodological rigor, its lack of distinction from consulting, and its tendency to produce either ‘research with little action or action with little research’ (Davison et al., 2004): *The term ‘canonical’ is used to formalize the association with the iterative, rigorous and collaborative process-oriented model developed by (Susman and*

*Evered, 1978) that has been widely adopted in the social sciences and hence which has gained 'canonical' status.*

CAR is a specific form of AR. Baskerville and Wood-Harper and others identified around identified a dozen deviations from the original AR (Baskerville and Wood-Harper, 1998; Davison et al., 2004): Canonical Action Research, Information Systems Prototyping, Soft Systems, Action Science, Participant Observation, Action Learning, Multiview, ETHICS, Clinical Field Work, Process Consultation, Reflective Systems Development, and Collaborative Practice. CAR aims to address organizational problems while at the same time contributing to scholarly knowledge, and for that purpose a set of five principles is defined, see (Davison et al., 2004). CAR is in many ways similar to ADR since it is iterative, rigorous and collaborative. It is different from ADR since the focus is not on the development of an IT-artifact, but on both organizational development and the generation of knowledge.

ADR is specifically applicable in this research since this research is a typical application of the traditional *engineering* approach. The problem statement in this research addresses a broad plethora of problems with IS development in professional organizations. However the root problem is at the stage of stating the problem statement not yet determined. As with traditional engineering, the problem (statement) is not easily scoped and hence the solution direction for the problem is to be developed during the interaction with the problem and its environment. ADR facilitates this approach by letting the problem formulation stage, the building and intervention stage, and the reflection stage to be reciprocally and mutually influential. Over the course of this research the problem statement is better defined which affects the building stage, which on its turn affects the reflection stage, all with reciprocal feedback loops to the other stages.



**Figure 2-1: Stages and principles in action design research.** Adapted from (Sein et al., 2010).

### ***2.2.1 ADR Stages***

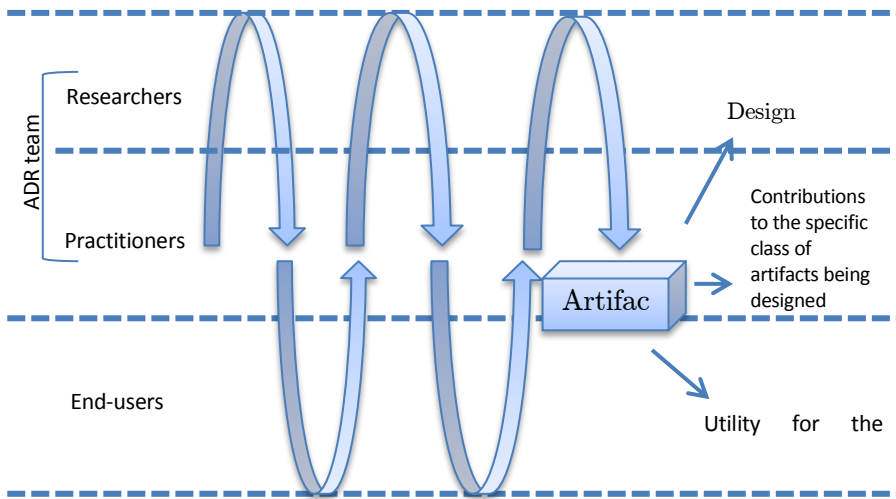
In essence, ADR is carried out in four stages that are reciprocally interconnected; (1) Problem formulation, (2) Building, intervention, and evaluation, (3) Reflection and learning, and (4) Formalization of learning. Each stage is accompanied with certain research principles, depicted in Figure 2-1. In the following sections the stages of ADR are briefly explained. Since this research is carried out according to the ADR approach, this thesis is structured likewise; in paragraph 2.3.3 the outline is presented. *A detailed description of the principles and methodologies applied in each stage is presented at the beginning of each chapter.*

### **Problem formulation**

The aim of the first ADR stage is to define the research problem, both in a concrete organizational setting as well as perceived as an instance of a class of problems. The problem domain of this research is professional service organizations, which are often suffering from low efficiency and low organizational output. These organizations desire to implement new information systems to overcome the low efficiency and low organizational output status. In this stage of ADR the class of problems is analyzed from a theoretical perspective. The analysis of literature on the problem domain and known solution directions for the IT artifact provides input for an initial design of the IT artifact. This initial design is the input for the next stage (building, intervention, and evaluation) in which the IT artifact is further developed and tested in case studies.

### **Building, intervention, and evaluation**

In the second stage of the action design research the results from the first stage are used to create an initial design of the IT artifact (Hevner et al., 2004; Sein et al., 2010). Since this research is carried out according to the ADR approach, the IT artifact under development is not designed on an abstract class level, but is practical applications in the organizational problem domain. PSOs are therefore analyzed on an individual organizational level. In paragraph 2.2.2 two separate organizations are selected in which the method to be developed is initiated and further developed.



**Figure 2-2: Action design research: overview.** Adapted from (Sein et al., 2010).

After the initial design, the artifact under construction will be iteratively improved by means of the intervention: the artifact will be implemented in the organization; and evaluation; feedback from the intervention will be used to improve the initial design. The ultimate goal of the *building, intervention and evaluation* stage is to support an iterative process at the intersection of the IT artifact and the organizational environment, see Figure 2-2. The aim of this stage is the realization of an embedded artifact. Characteristic of this stage is the continual design and evaluation as a result of emerging theory and design approaches for the class of problems under review.

### **Reflection and learning**

In the building, intervention and evaluation phase focuses on the realization of an IT artifact that meets the organizational problem's requirements. The aim of ADR is more than to provide an IT artifact that solves problem in one instance of a type of problems, but to induct from the solution to a specific problem a solution for a whole class of problems. This stage in the action design research approach aims to tackle this problem.

The reflection and learning stage is carried out in parallel with the problem formulation and the building, intervention and evaluation stage. During this stage the research process is adjusted to resemble improved understanding of the domain specific problem and the IT artifact that provides the solution. Furthermore, the contributions to the body of knowledge are identified. This stage draws on the principle of *guided emergence*.

### **Formalization and learning**

In the fourth stage of action design research, the IT artifact that has been developed as a solution to the specific organizational problem is to be developed into a general solution concept for a class of problems (Sein et al., 2010). The general solution concept can be characterized as design principles for the set of problems identified in the problem formulation stage, and can further refine theories that contributed to the initial design from the building, intervention, and reflection stage.

The complexity of ADR is that the final artifact's definition that will satisfy the initial problem statement is not yet seen before. Therefore, the new artifact will demonstrate a new domain for which the current knowledge base might not provide sufficient background. During the design and development of the artifact that will eventually (and hopefully) satisfies the problem statement, new additions will have to be made to the knowledge base as a sort of byproduct. New methods and measures that determine the designed artifact's relative success can be part of these new additions to the knowledge base (Hevner et al., 2004). There are various forms in which the formalized results from ADR can be added to the knowledge base, one very important is scientific publications about the research, similar to the communication principle in Design Research (Hevner et al., 2004).

#### ***2.2.2 ADR process***

Because of the many feedback loops and iterative processes the ADR approach itself is an ill-structured method to develop and research a



problem and its solution in practice, see Figure 2-1. This research though is structured to certain extent, predominantly for the sake of writing this thesis. In this paragraph two elements of the research process are discussed; the research team and the research process flow.

### Research Team

The research team consists of the researchers, practitioners involved and the end-users of the IT-artifact. The first and second group together form the ADR team (Sein et al., 2010): *the intent of the ADR team is not to solve the problem per se as a software engineer or a consultant might. Neither is it to only engage in an intervention within the organizational context in which the problem is observed. Instead, the action design researcher is interested in generating knowledge that has the potential of application to the class of problems that the specific problem exemplifies.* In this research's case the class of problems is the incapability to develop information systems development methods tailored to the PSO domain.

Roles	Activities
Researchers	Problem formulation Building, intervention and evaluation Reflection Formalization and learning
Practitioners	Problem formulation Building, intervention and evaluation Reflection Formalization and learning
End-users	Building, intervention and evaluation Reflection Formalization and learning

Table 2-1: Roles and activities of the ADR team. Adapted from (Sein et al., 2010).

Since the aim of any ADR research is not only to construct an IT-artifact that addresses the class of problems typified by the encountered situation, but also to address the specific problem situation encountered in the organizational setting, end-users are

involved in the ADR process too. The activities of the end-users deal with experiencing and evaluating of the implementation of the IT-artifact. In Figure 2-3, the flow of the process is depicted along with the activities of the groups that make up the ADR team. In Table 2-1 the roles and their corresponding activities are listed.

### Process Flow

The stages that are listed in Figure 2-1 and the roles that are mentioned in the previous section are depicted in chronological order in Figure 2-3. In essence the ADR research stages overlap / interact during the research process.

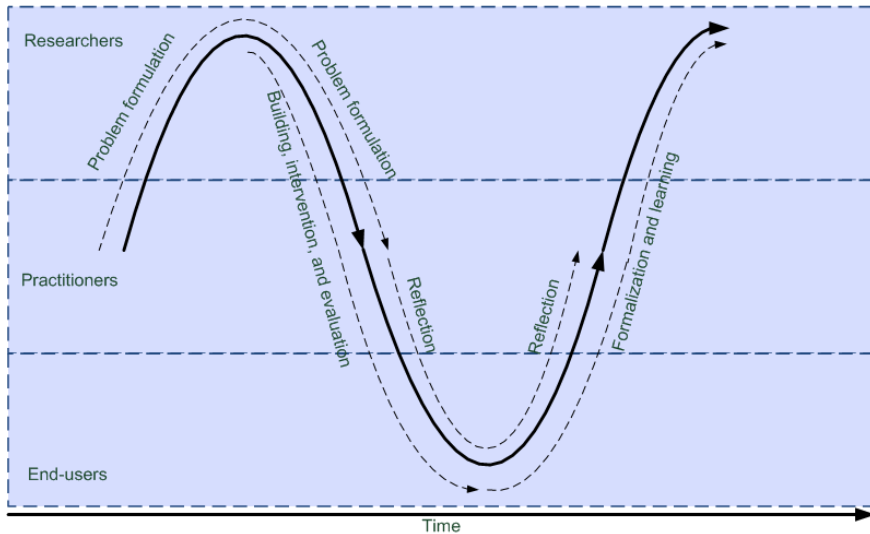


Figure 2-3: ADR process flow. Adapted from (Sein et al., 2010).

The ADR process results in design principles about the IT artifact, Contributions to the specific class of artifacts being designed, and utility for the end users.

### 2.3 Research approach

This paragraph describes the selection of projects, the research process, and the structure of this thesis.

### ***2.3.1 Project selection***

In this research we aim to engineer an approach to develop ISD methods for the specific PSO domain. Because we apply the ADR approach on this research, it is important to select appropriate projects to formulate, build, intervene, and evaluate our IT artifact under development. For using the ADR framework, it is necessary to be able to execute these activities in close cooperation with practitioners and end-users. Therefore two projects are selected which we assume to be highly suitable for our research:

1. The selection of an information system and process redesign in a radiology department of a mid-sized hospital (MST case).
2. The design and implementation of an information system and process redesign in the outpatient clinics in a mid-sized hospital (Gelre case).

Both projects are in situated in a PSO environment. In both projects, we, as researchers, are directly cooperating with practitioners, and have direct access to end-users.

### ***2.3.2 Validity threats***

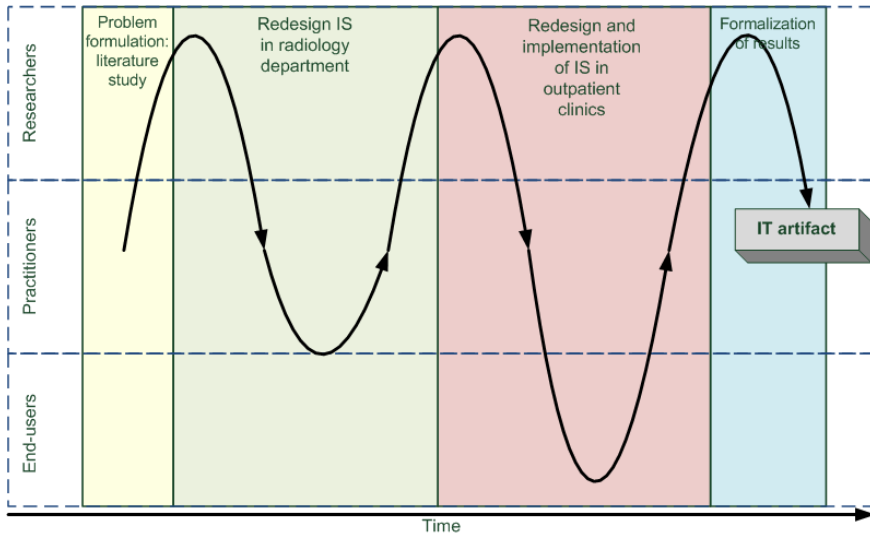
This action design research validity is an important aspect. Yin advises to consider construct validity, internal validity, external validity, and reliability for research in general (Yin, 2009). Since this is Action Design Research, internal validity is not of major importance; we are not deducting causal relations in this research (see: the discussion about contingency theory in paragraph 3.1.2). Considering the other types of validity; in line with Yin's suggestion, multiple sources of evidence are used to ensure *construct validity*. In chapter 4 and 5 the IT-artifact under-development is implemented in an organizational setting. In each chapter a description is presented how the various sources are gathered and used to support the findings in both chapters, see paragraph 4.2 and 5.2. Since this research is carried out in two distinct organizations, we ensure that the *external validity* is asserted by applying an organized process (as

far as that is possible within the ADR approach) that is conducted twice in the two organizations. Based on this standardized approach generalizations can be made about the findings. *Reliability* of the research is guaranteed by a strict recording and documentation of all the findings in this research.

### ***2.3.3 Research process***

Sein et al. identify two ends of the *building, intervention and evaluation* stage spectrum: the IT dominant approach (aimed at novel IT artifacts that require creating an innovative technological design at the outset) and the organizational-dominant approach (aimed at generating design knowledge with the nature of the artifact as the primary source of innovation). In practice, the main difference between these approaches is the level of involvement of the practitioners and end-users in the design process.

The *organizational-dominant* alternative seems to be the best fit with this research, because the aim is to create design knowledge about methods for ISD in the specific context of professional organizations. However, because of the research being carried out in studies in practice we decided to apply the *IT dominant* approach is selected for this research. This implies that the first research iteration in practice will only be a close cooperation between researchers and practitioners and omit direct end-user participation, see Figure 2-4.



**Figure 2-4: Research process flow, adapted IT dominant ADR approach.**

Next to the stages in ADR that are presented in Figure 2-4, also the roles in the process are depicted. In ADR there are three main roles; the researchers, the practitioners, and the end-users. The researchers in this research are the author of this thesis, accompanied by other (university) researchers. The practitioners in this research are the project management responsible actors that are employed by the case organizations. The practitioners are internally responsible for developing the IS in the target organizations. The end-users in this research are the stakeholders that are the actors that are to be working with the developed IS; in the selected cases these are next to the medical staff of the hospitals also the administrative staff, and management.

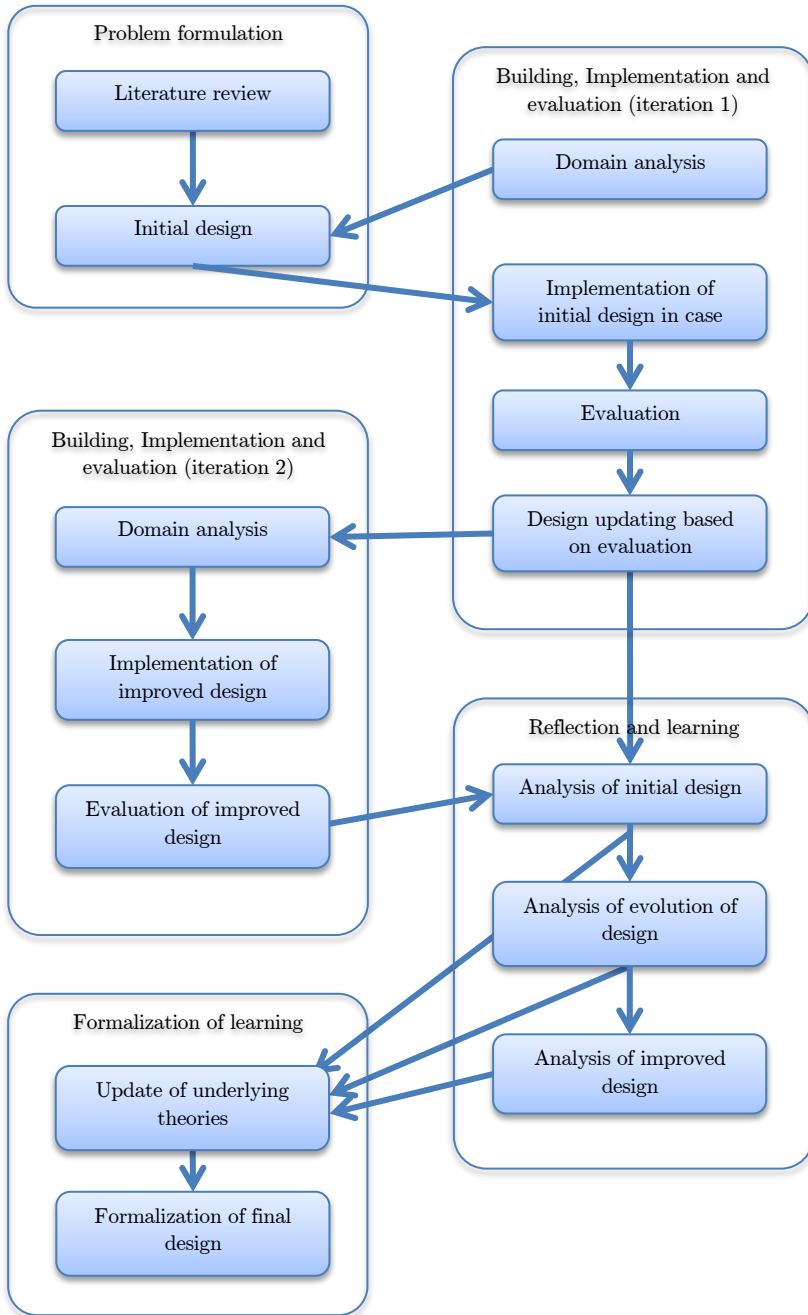


Figure 2-5: Meta-level process description ADR research.

A meta-level process model of this action design research is presented in Figure 2-5. In accordance to the reciprocal process model for ADR, the process of this research has feedback loops and interaction between activities in different stages. In the following chapters the meta-level process model is extended to a process-data diagram (van de Weerd et al., 2005), and is discussed in a section *factsheet* see paragraphs 3.2, 4.2, 5.2, 6.2, and 7.2. Each specific block is discussed in detail in a separate chapter, see Table 2-2 in paragraph 2.3.4. Throughout this thesis the meta-level process flow is used to point out the stage of the research and its accompanying inputs and deliverables, parallel to the approach presented in Incremental Method Evolution (van de Weerd et al., 2010).

### 2.3.4 Thesis structure

This thesis is structured according to the ADR approach. In Table 2-2 the outline of this thesis is listed. There are two iterative studies in this research, therefore the building, intervention, and evaluation stage is two times the topic of a chapter. Since this research is carried out as an IT-dominant version of ADR, the first study is not reflected with the end-users (so there is no chapter dedicated to the reflection of the MST case). The Gelre case goes through the full spectrum of the ADR cycle and therefore there are two chapters dedicated to this case: and building intervention and evaluation chapter, and a reflection chapter.

Chapter	Topic	Goal	Deliverable
1	Introduction	Define problem statement	Research questions
2	Research methodology	Describe research approach	Research methodology
3	Problem formulation	Develop initial artifact based on literature review	Initial design of IT artifact
4	Building, intervention and evaluation in MST case	Continuation of the development of the initial design of IT artifact in case setting	Updated design of IT artifact

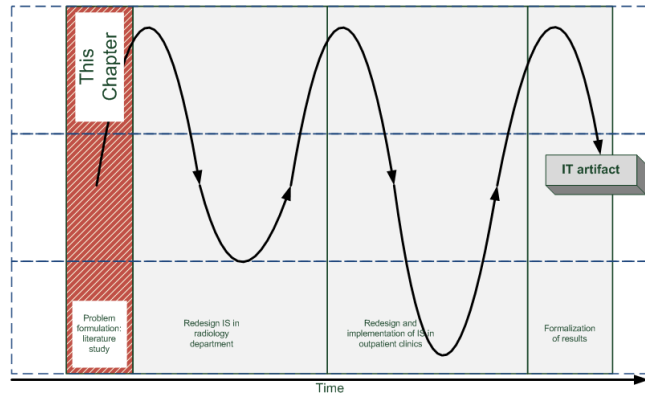
5	Building intervention and evaluation in Gelre case	Based on updated design of IT artifact develop it further in case setting	New insights about IT artifact
6	Reflection MST and Gelre case	Analyze new insights about IT artifact based on MST case	Utility for the users and contributions to the IT artifact class being designed
7	Formalization and learning	Analyze results from both MST and Gelre case on general principles	Design principles about IT artifact
8	Conclusions	Summarize the key contributions of this research with their connection to the original problem statement	Key contributions in light of the original problem statement

**Table 2-2: Thesis structure; chapters, topics, goals and deliverables.**



### 3 Problem formulation and initial design

The aim of this chapter is to develop an initial design for the main goal of this research: to develop an appropriate approach to deal with



information systems development in professional service organizations (the IT-artifact). The process of this research and the part of this chapter in the research project is depicted in the figure above. The initial design is based on the relevant literature. There are three fields from the body of knowledge that relate to this research; professional service organizations, information systems development, and situational method engineering; the latter being an approach to develop information systems development methodologies tailored to a specific situation. These three research fields form the foundation of the initial design of the IT artifact in this research, which is the outcome of this chapter. Paragraph 3.4 discusses information systems development and the relation to this research, in paragraph 3.5 the professional service domain is analyzed, in paragraph 3.6 situational method engineering is introduced as a means to this research end, and in paragraph 3.8 the literature reviews are brought together to develop an initial design of the research direction.

The research methodology referring to each stage of the ADR approach is discussed in detail at the beginning of each chapter. Table 3-1 presents an overview of the research methods that are applied in the problem formulation stage of this research that is addressed in this chapter.

Method	Goal
Literature review	Gain theoretical insights of both the organizational problem domain and the technological solution space
Contingency theory	Develop initial design of the IT artifact in this research

**Table 3-1: Applied methods in the problem formulation stage.**

### 3.1 Research methods

Since this research follows the ADR methodology, this chapter adopts two principles: the principle of praxis inspired research, and the principle of a theory ingrained artifact.

#### Principle 1: Praxis-inspired research

Action design research views problems as knowledge creation opportunities. For ADR, the most promising opportunities are to be found at the intersection of technological and organizational domains. The aim of this research is to develop an ISD solution (technological domain) to the problems encountered in PSOs (organizational domain). “ *The intent of the ADR is not to solve the problem itself per se as a software engineer or a consultant might. Neither is it to only engage in an intervention with the organizational context in which the problem is observed. Instead, the action design researcher is interested in generating knowledge that has the potential of application to the class of problems that the specific problem exemplifies*” (Sein et al., 2010).

ADR aims to develop a solution that is derived from a *technical solution space* for a problem that is part of an *organizational domain*. The solution under-development derived from the technical solution space in ADR is in this research referred to as *IT artifact*. In chapter 2 two organizational cases are selected in which a PSO suffers from poorly designed and implemented IS. These two cases provide practical inspiration to carry out this research and to further specify the problems that are addressed in this research.

## **Principle 2: Theory-ingrained artifact**

The second principle in the problem formulation stage in ADR, is the principle that emphasizes that the artifacts created and evaluated via ADR are theory-ingrained; artifacts both use as well as test and generate theories. According to Gregor's classification of theories in the information systems field, Type IV (Explanation and Prediction theories) and Type V (Design theories) are likely candidates to be used in ADR (Gregor, 2006; Sein et al., 2010). In this research both contingency theory (Type IV) and situational method engineering (Type V) are applied. Both these fields are analyzed in this chapter by means of a literature review.

### ***3.1.1 Method: Literature review***

The goal of any literature review is to create a firm foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists, and uncovers areas where research is needed (McConnell, 1993). Literature reviews can be written when an author has completed or made substantial progress on a stream of research (and is thus well positioned to tell his/her colleagues what is learned), or when a literature review is completed prior to embarking on a project and thus has developed some theoretical models (Webster and Watson, 2002). The literature review method derived from Webster and Watson is applied to analyze (a) the characteristics of professional organizations that are, according to contingency theory, the organizational variables influencing the ISD process, and (b) the available method components for the specific stages in ISD that are most characterizing for professional organizations.

The method for conducting a literature review in this research is basically founded on Webster's and Watson's approach, extended with some techniques developed by (Schwartz and Russo, 2004). In essence this approach consists of (1) the assembly of literature in which relevant articles in leading IS journals are located. The search for articles is extended with the use of forward and backward

citation analysis; (2) the filtering of relevant articles; (3) the identification and evaluation of the located content; (4) the grouping and clustering of purposes (themes) and objectives in papers; and (5) triangulation, with which the final data is reviewed.

### ***3.1.2 Method: Contingency theory***

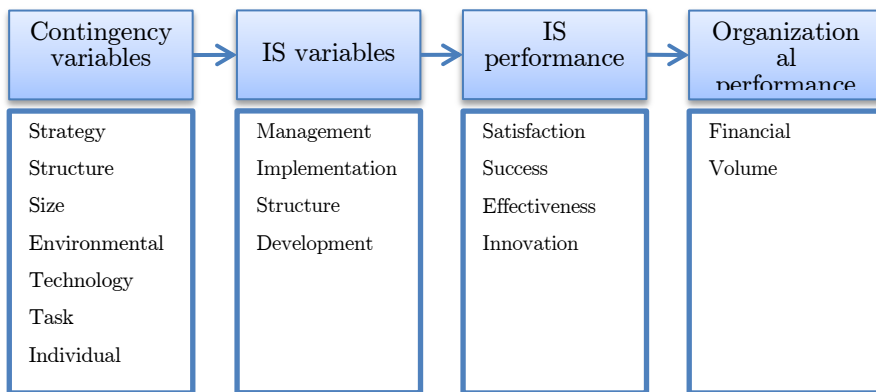
Contingency theory implies that the best way to organize an ISD project depends on the nature of the environment to which the organization must relate (Scott, 2002). The main ideas that underlie contingency theory are (Donaldson, 2001; Morgan, 1998):

- Organizations are open systems that need careful management to satisfy and balance internal needs and to adapt to environmental circumstances
- There is no best way of organizing. The appropriate form depends on the task or environment
- Management must be concerned with achieving alignment and good fits
- Different types or species of organizations are needed in different types of environment

A significant part of IS research is based (implicitly or explicitly) on the contingency theory approach (Weill and Olson, 1989). This behavioral theoretic approach very much aligns with the direction taken in this research. To develop ISD methodology specific for professional organizations the link (if there is any) between methodology, information system variables, information systems performance, and organizational performance has to be clearly identified. Chapter 1 explains that the PSO domain significantly differs from other organizational domains, and because of that generic approaches to ISD have not been successful. Therefore the contingency theory approach is brought into this research to identify the (mis-) fit between ISD and PSOs.

Whereas applications of contingency theory in organizational science “*attempt to understand the interrelationships within and among*

*organizational subsystems as well as between the organizational system as an entity and its environments. It emphasizes the multivariate nature of organizations and attempts to interpret and understand how they operate under varying conditions”* (Szilagyi et al., 1983), applications of contingency theory in information systems research study the fit between information system and organization and try to determine the IS performance, which affects the organization’s performance (Weill and Olson, 1989). The fit can be analyzed and measured by isolating contingency variables (e.g. organizational size, strategy, or structure), see Figure 3-1. As Weill formulated it: “ *the contingency approach suggests that a number of variables influence the performance of information systems; the better the fit between these variables and the design and use of the IS, the better the IS performance”* (Weill and Olson, 1989). In contingency theory the causal relation between IS performance and organizational performance is thereafter often assumed.



**Figure 3-1: Representation of Contingency Theory in IS Research. Adapted from (Weill and Olson, 1989).**

Basically, in contingency theory it is indicated that there is a causal relationship between the contingency variables, IS variables, IS performance, and organizational performance. Figure 3-1 demonstrates the assumed causal relationships between the contingency theory constructs.

Based on literature on contingency theory, it has become clear that a methodology for ISD can only be standardized to a certain extent, in order to be effective. The organizational contingency variables, the IS variables, and the IS performance variables vary in every instance of a system's development.

Contingency theory thus informs us that there is no uniform approach to ISD. The methodology should be adapted for each specific instance of systems development. However, many ISDMs cover a broad spectrum of generic development activities in order to deal with each specific development project's complexity (Avison and Fitzgerald, 2003). Especially traditional methodologies are designed to be able to cope with every deviation from the ideal situation. This approach, in which methodologies are not developed for specific purposes but in a one-method-fits-all approach, has led to a backlash against complex and extensive methodologies (Wastell, 1996).

There are four assumptions underlying the contingency model in IS research; (1) *fit*: there is a match between at least two contingency variables, for instance the fit between IS characteristics and IS performance, or IS performance and organizational performance; (2) *performance*: the dependent variable in IS research is often IS performance, sometimes the organization's performance; (3) *rational actors*: IS development, use, and implementation are highly influenced by the organizational actors. These actors are rational actors, acting in accordance with optimization of organizational performance; (4) *deterministic model*: most studies in IS research assume a deterministic model in which only causality is demonstrated, similar to Figure 3-1 (Weill and Olson, 1989).

In literature, some critics point out that there are some issues in the application of contingency theory (Schoonhoven, 1981; Weill and Olson, 1989). The first criticism is that a study that explicitly or implicitly applies contingency theory most often blindly follows the native meta-theory. Furthermore there are conflicting empirical results from studies measuring similar constructs that implicates low correlations. A third issue is that the concepts of fit and performance

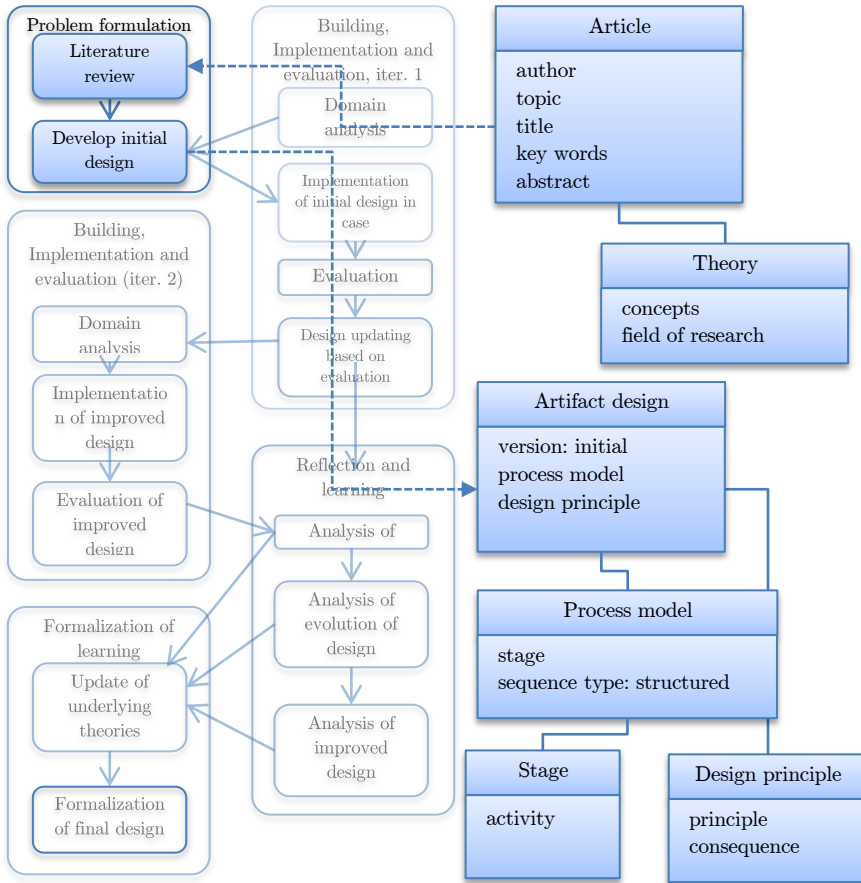
are ill-defined. Organizational performance is often reduced and operationalized to one single measure, for instance ROI, profit or net worth. Even the positive link between IS performance and organizational performance is questionable, and more often assumed rather than measured. The last main criticism is that researchers that (explicitly or implicitly) apply contingency theory often have a narrow perspective. It has been demonstrated that contingency variables chosen in any study account only for a small percentage of the variance in organizational performance.

Taking these criticisms in account, it is evident that it is natural to apply contingency theory as a foundation in this research, however it should be noted that, when drawing conclusions, the correlation between contingency variables does not necessarily implicate causal relationships.

In this research contingency theory is used to structurally indicated and analyze the fit between the IS and the PSO in the development process. Because of the criticisms on the strict application of contingency theory (Weill and Olson, 1989), in this research the theory will only be applied as an high-level concept, not to formally measure and analyze the *fit*.

## **3.2 Factsheet**

In Figure 3-2 this research's meta-level process model is extended to a process-data model in line with the approach taken in Incremental Method Evolution (van de Weerd et al., 2010). In this figure both input that is used in this chapter as well as output that is produced is described. In successive chapters this model is extended in line with the updates of the design of the IT artefact under-development.



**Figure 3-2: Process-data diagram: Snapshot Problem formulation stage; input and output.**

### 3.3 Literature review: approach

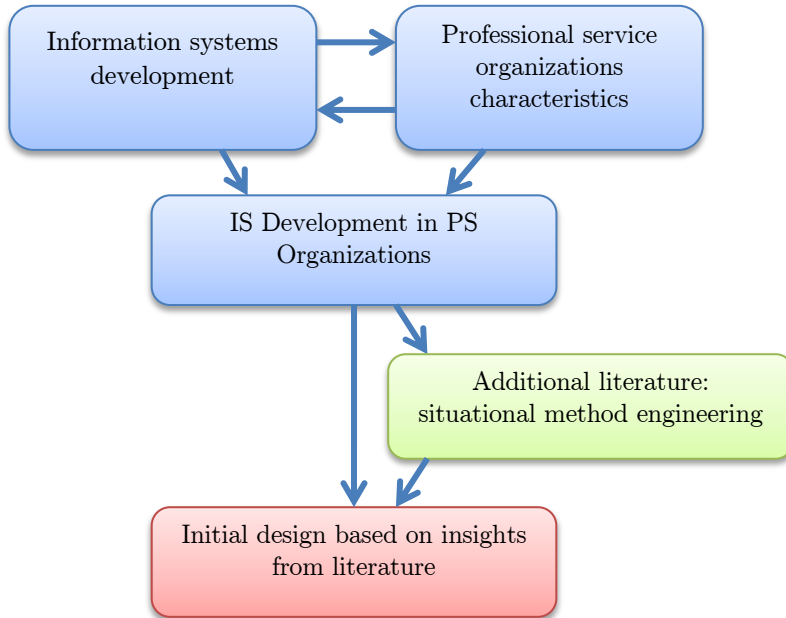
In the first chapter a rough scope is set for the aim of this research. Since this research is based on the ADR paradigm, this is in line with ADR's incremental ideas about the development of solutions for problems in the business domain. The incremental, reciprocal nature of the research approach has consequences for the approach to conduct a literature review, of which the result is presented in this



chapter. This paragraph clarifies the structure of the literature review in this chapter.

There are three main ingredients for the literature review that aids in the problem formulation stage in this research: the information systems development field; the professional service organizations field, and the combination of these two fields: information systems development for professional organizations. Figure 3-3 depicts the structure of this literature review, containing the three main ingredients of the review. As can be seen in this figure two additional elements are added to the literature review in the problem formulation stage: a review on literature on situation method engineering (reasons to add this are explained in paragraph 3.6) and the eventual initial design of the artifact that is based on the outcomes of the literature review.

As with the action design research paradigm, the structure of this literature review is iterative and reciprocal; results from the review on ISD are used to specify the literature review on PSO and vice versa. Both these review are input for the analysis of ISD in PSOs. As will be explained in paragraph 3.7 situational method engineering is a vital part of the initial design of the artifact in this research. Both the review on ISD in PSO and the review on situational method engineering combine as input for the initial design that is presented in paragraph 3.8.

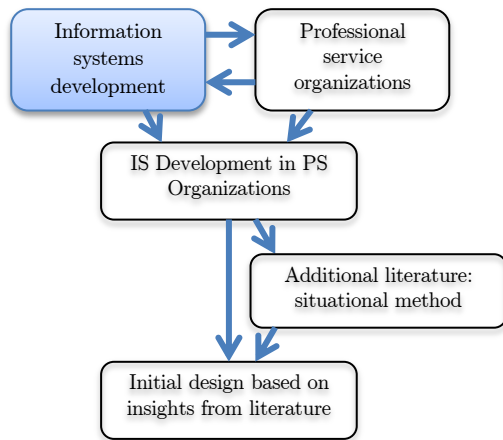


**Figure 3-3: Structure of literature review.**

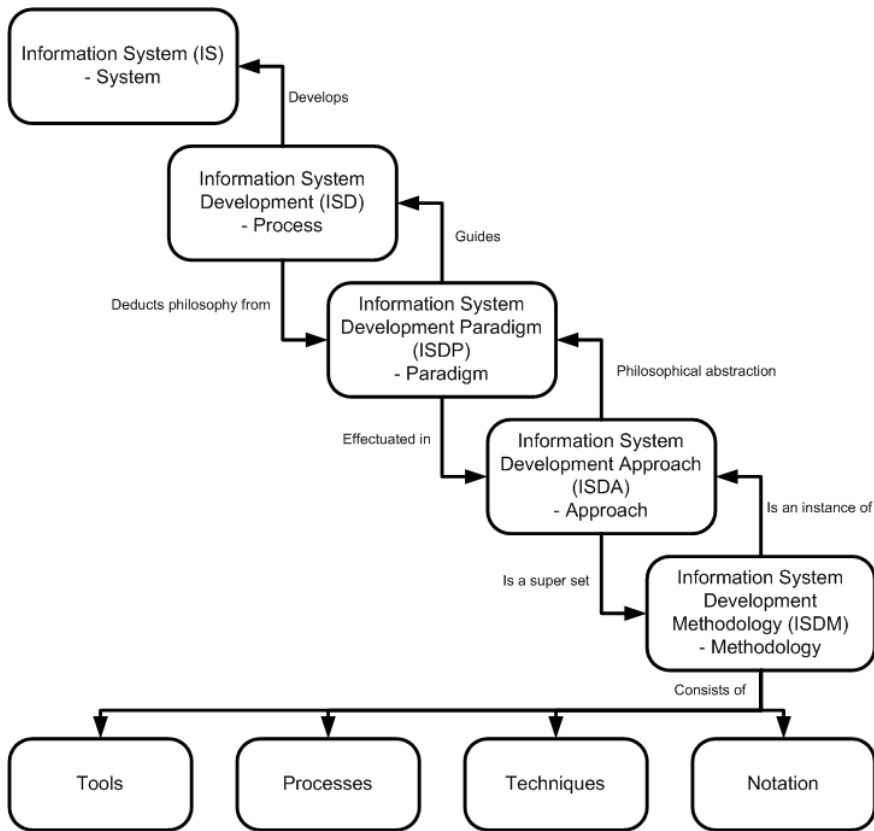
The actual literature review process and the structure of this chapter are asynchronous; because of the iterative, reciprocal nature of the literature review, insight of every element in Figure 3-3 influences the reviews in other elements in this figure. However to contribute to the readability of this chapter, the outline of this chapter depicts a very sequential process. To increase the visibility of the structure, Figure 3-3 is repeated at every paragraph to indicate the process flow.

### 3.4 Literature: Information systems development

In the problem statement the lacking methodological support for ISD is identified as one of the root problems of the ineffective use of information systems in professional organizations. Therefore the analysis in this paragraph of the information systems development field is used as a foundation for the development of the initial design of the research direction in paragraph 3.8.



ISDM can be perceived in light of the information system (IS) under-development, being implemented in an organization. Figure 3-4 provides a concept map of IS, in relation to the methodologies that are there to develop it. The figure presents five concepts that are closely connected; information systems (IS), information systems development (ISD), information systems development paradigm (ISDP), information system development approach (ISDA), and finally the information system development methodology (ISDM).



**Figure 3-4: Information systems development: concept map, source: (Hirschheim and Klein, 1989; Iivari et al., 1998; Kroenke, 2011).**

In this paragraph's sections an overview on ISD is created based on this concept map. The aim is to analyze and identify the potential fit between ISD and the professional service organization domain. The elements in the map are used to focus the research on specific elements from the broad set of methodologies that are currently available. This paragraph is structured according to the flow of Figure 3-4; each term is analyzed in depth in a separate section.

### ***3.4.1 Information systems***

An information system (IS) comprises the combination of technology and social activities; the information technology and the

people' s activities using the information technology. In essence an information system is a group of components that interact to produce information in an organization. There are five fundamental components of an information system; hardware, software, data, procedures, and people (Kroenke, 2011). In this research IS is defined as the system under development (SuD) that is implemented to support the processes in the professional service organizations.

According to literature, an IS is a representation of a business environment, since it is implemented to support business functions in the most efficient way (Carr, 2003). An information system is the artificial representation of a real world system, as perceived by somebody, built to enable information processing functions (Wand, 1996). In this research the real-world domain are professional service organizations, and the IS are artificial representations of that domain. Wand identifies three aspects of information systems: (1) the deep structure: the aspects of the IS that reflect the represented domain; (2) the surface structure: the user interface characteristics of the IS; and (3) the physical structure: the technical means that are employed in the implementation of the IS (Wand, 1996). In this research the focus is on the *deep structure* of the IS, hence the representation of business processes, workflows, hierarchies and alike.

### ***3.4.2 Information system development***

Information system development (ISD) is the process to design, implement and align an IS for a specific organization it is developed for. For this research ISD is defined as “ a change process taken with respect to object systems in a set of environments by a development group using tools and an organized collection of techniques collectively referred to as a method to achieve or maintain some objectives” (Lyytinen, 1987; Welke, 1981). Since information systems are understood to include both the social as the technical elements in the organization it supports, ISD includes both the development of the social as the technical elements of the SuD.

### 3.4.3 ISD Paradigm

Information system developers approach the ISD task with a number of explicit and implicit assumptions about the nature of the organization, the design task, and the expectation about the design task. Research demonstrates that the assumptions can be grouped into four paradigms that guide the ISD process. According to Hirschheim et al. the paradigms are: the functionalistic paradigm (all information systems are designed to contribute to specific ends); the social relativist paradigm (there is no single reality, only different perceptions about is: system objectives emerge as part of the organizational construction of reality); the radical structuralist paradigm (fundamental social conflict is endemic to society, yet there is an objective economic reality); and the neohumanist paradigm (information systems are developed to remove distorting influences and other barriers to rational discourse) (Hirschheim and Klein, 1989), see Figure 3-5.

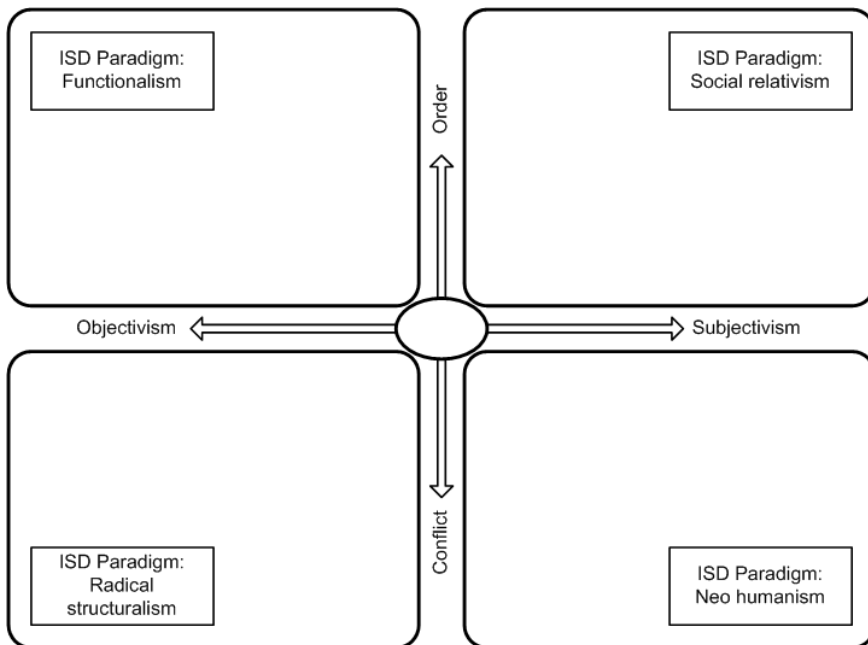


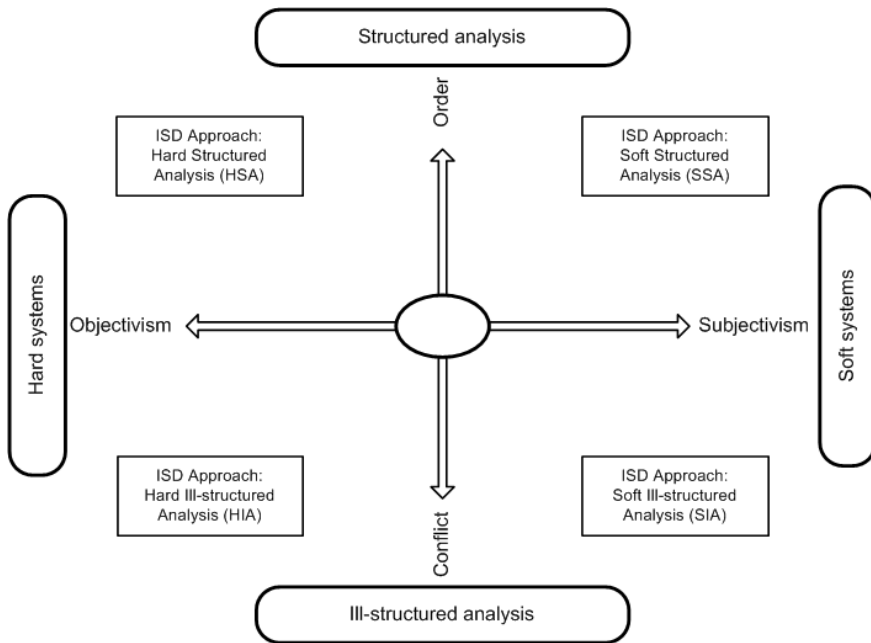
Figure 3-5: Paradigms in ISD, adapted from (Hirschheim and Klein, 1989).

The traditional paradigm for ISD is the functionalistic paradigm, where information systems are designed to solely contribute to ends they are developed for. As discussed in the previous paragraphs, the key characteristics of professional organizations induce that the organizational ends are not clearly defined, and therefore, the ends for information systems are often unclear. We aim to approach ISD as a means to define the ends for information systems and therewith develop effective systems that support the interests of the heterogeneous group of knowledge workers that are supposed to use the IS.

#### ***3.4.4 ISD Approach***

An ISD Approach (ISDA) is a set of goals, guiding principles, fundamental concepts, and principles for the ISD process that drive interpretations and actions in ISD (Iivari et al., 1998). Therefore, ISDA is an ISD paradigm effectuated in a process. An ISDA is an abstract class of ISD methodologies, which are analyzed in paragraph 3.4.5.

In Figure 3-6 four different ISDA' s are identified, based on the distinction of the fundamental paradigms underlying an ISDA. The two main determinants for ISDA are the extent to which an approach is structured, and the extent to which a problem is defined that is addressed by an ISDA. In other words in this research ISDA' s are differentiated based on the distinction between structured and ill-structured analysis, and the distinction between the well-defined and ill-defined problems that are addressed by ISDA' s. The latter distinction is often labeled as hard systems opposed to soft systems.



**Figure 3-6: ISD approaches, adapted from (Iivari et al., 1998).**

The first determinant, structured versus ill-structured analysis, differentiates between traditional and more modern ISDA's. Structured analysis is the traditional approach for ISD, originating from the 1960's and 1970's. It is characterized by a top-down approach in which the sequence of processes that will ultimately deliver the SuD is predefined, and thus the processes will be executed in that sequence (Boehm, 2006; Royce, 1987). Ill-structured analysis – more often labeled agile analysis – refutes the necessity of predefined structures in an ISD process flow, the need for formal notation and documentation. Ill-structured analysis aims for delivering functionality over process formality (Abrahamsson et al., 2003; Boehm, 2002, 2006).

The second determinant, hard systems versus soft systems, differentiates between well-defined and ill-defined problems an ISDA has to address. The hard systems approach assumes that the world is considered to be systemic and can be studied systemically. In this approach, the ISD process is well-defined, has a single, optimum



solution, and will be best executed using the scientific approach to problem solving. In hard systems thinking, systems are understood to be ontological entities; entities that are real in the world and which can be interpreted only in a single perspective (Checkland, 1978). The soft systems approach, the world is considered to be problematic, but can be studied systemically. In the soft systems approach systems are considered to be epistemological; a mental construct in a human understanding, resulting from an interpreting activity. Developing an IS according to soft systems thinking requires the insights of every stakeholder in the development process, because every stakeholder of the SuD interprets the SuD and its functionality in a different perspective. Depending on the perspective a very different understanding is created of the particular SuD (Checkland and Poulter, 2006; Wilson, 1990).

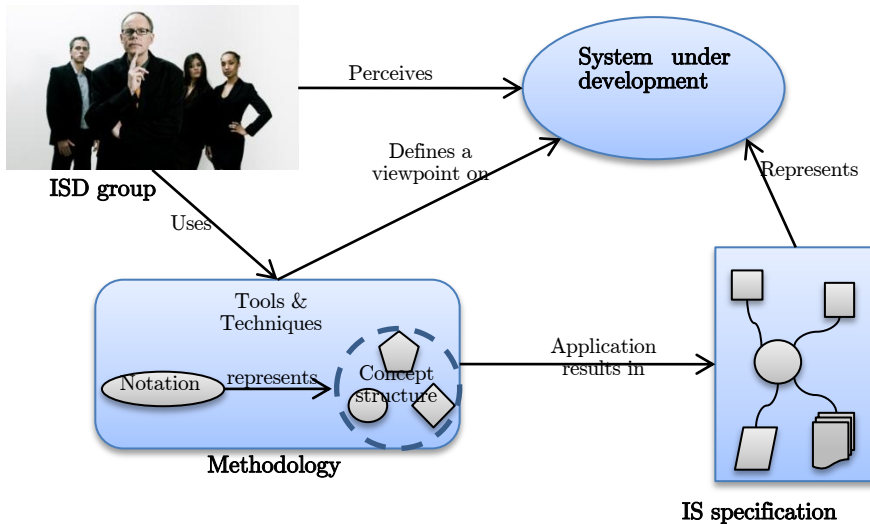
Based on the discriminating concepts underlying ISDA, four ISDA's are found, one in each quadrant: (1) Hard Structured Analysis (HSA), (2) Soft Structured analysis (SSA), (3) Hard Ill-structured Analysis (HIA), and (4) Soft Ill-structured Analysis (SIA). Each ISDA corresponds with an ISD paradigm (see paragraph 3.4.3). The functionalistic paradigm corresponds with the hard structured analysis approach, the social relativist paradigm with the soft structured analysis approach, the radical structuralist paradigm with the hard ill-structured analysis approach, and the neo-humanistic paradigm with the soft ill-structured analysis approach.

### ***3.4.5 ISD Methodology***

Before advancing in the presentation of the ISD methodology in this section, we make a brief stop here, to make a remark about the syntax used to describe the sets of tools, notations, techniques and languages that make up the topic of this section; methodology. As Brinkkemper noted there are two words in used to describe this concept: *method* en *methodology* (Brinkkemper, 1996). Where the word *method* comes from the Greek 'methodos', *methodology* adds the part 'logos' which is Greek for "logic of" to the first part, which means way of investigation. Therefore the term *methodology* would

imply something like an investigation into methods / the science of methods. In this research we use the term *methodology* interchangeably with *method*. The definition of *method* (and thus for *methodology* in this research) is presented by Brinkkemper:

*A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products* (Brinkkemper, 1996).

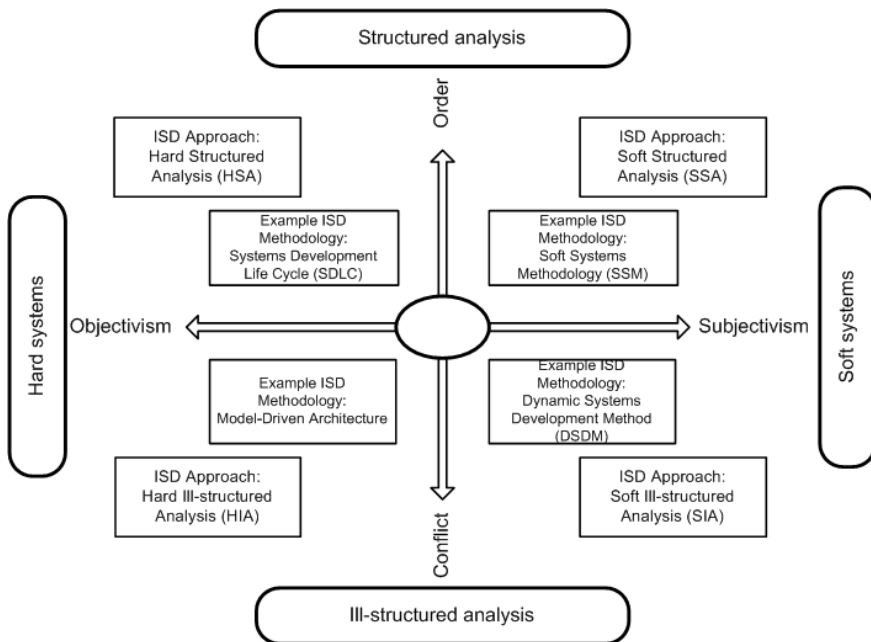


**Figure 3-7: The role of methodology in ISD, adapted from (Lyytinen, 1987).**

In the concept map in Figure 3-4, an ISD methodology is an instance of an ISD approach. An ISD Methodology can be interpreted as an organized collection of concepts, methods, beliefs, values and normative principles supported by material resources (Iivari et al., 1998). More specifically, “an ISDM is codified into a set of goal-oriented procedures that guide the work and cooperation of the various parties (stakeholders) involved in the building of an IS application. These procedures are usually supported by a set of preferred techniques and tools, and activities” (Hirschheim, 1996).

The aim of any ISD methodology is to specify, develop, and implement the SuD. The actors applying the ISD methodology, use it to define a viewpoint on the SuD. Schematically, the processes of specification and development of the SuD are schematically depicted in Figure 3-7. Next to these activities, the implementation is also an important element of an ISDM.

Different ISD methodologies include different types and sets of tools and techniques (including their conceptual structures and notations), and processes or workflows. Interrelations between techniques and tools can be defined differently even between methodologies which use the same techniques and tools, and the procedures for building and analyzing the results can be different. Although there is diversity among ISD methodologies, they include similarities, for instance the use of concepts and notations (Tolvanen, 1998).



**Figure 3-8: Examples of ISD Methodologies in the four perspectives.**

An ISDM is an instance of an abstract ISDA class, as is discussed in paragraph 3.4.4. Whereas an abstract ISDA in essence is an ISD

paradigm effectuated in a process, ISDM' s are concrete, practical methodologies that include all necessary features for developing an IS. Iivari states that the specific features of a specific ISDM can sometimes conflict with the abstract ISDA class the ISDM is part of (Iivari et al., 1998). In Figure 3-8 the same ISDA overview is presented as in paragraph 3.4.4, but now extended ISDM examples accompanying each ISDA class.

### **Evolution of ISD methodologies**

ISDMs became an issue in the 1950' s, when information systems were built as engineers construct buildings. The development process is only initiated after the design is carefully decided and calculated. When software became more complex in the 1970' s, the need for formality and structure gave rise to the well-known waterfall-model or life-cycle development model (Boehm, 1988; Royce, 1987). In this model, the development of software systems becomes requirements-driven confined to successive phases. This means that the next phase in the system development process is only initiated when all artifacts in the previous stage required careful verification and validation.

Over the past decades technology progressed and the demands of practitioners changed. Numerous ISD methodologies are introduced that deal with the changed requirements. In this section we discuss that recent methodologies for ISD that focus on business value and integration, and are more flexible and less formal are most suitable for the development of information systems in professional organizations.

Multiple authors have written about the evolution of methodologies for system development. In this section we use the work from Boehm and Avison to demonstrate the changing methodological approach to ISD (Avison and Fitzgerald, 2003; Boehm, 2006). In Figure 3-9 we present a graphical overview the evolution of ISD methodologies.

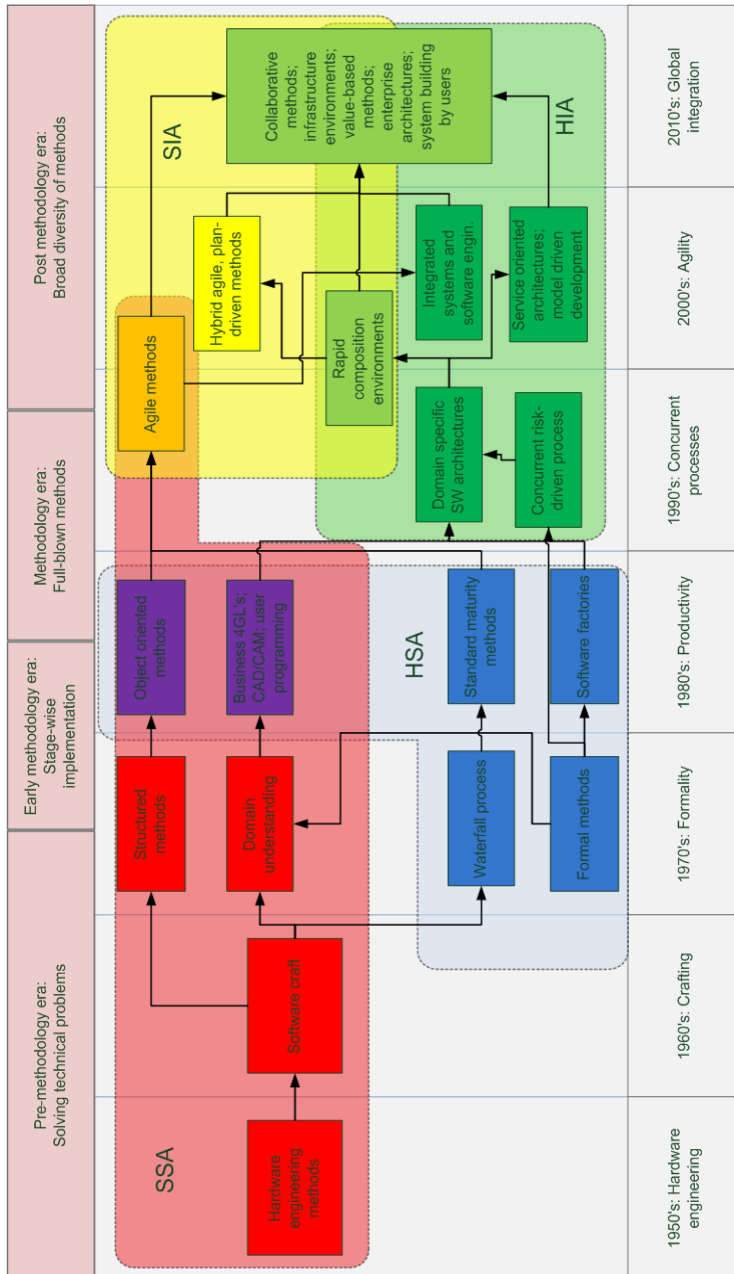


Figure 3-9: Evolution of ISD methodologies, adapted from (Avison and Fitzgerald 2006, Boehm 2006).

Both Avison and Boehm describe the evolution of ISD methodologies chronologically from the invention of the computer till the current times. Whereas Avison distinguishes four eras in the development of methodologies (pre-methodology, early methodology, methodology, and post-methodology era), Boehm groups development methodologies by decade, see Figure 3-9. In this paragraph four ISDM meta-groups are identified, according to the classification of ISDMs and ISDAs in Figure 3-8. Since ISDMs are practical, concrete instances of abstract ISDA classes and include all necessary practical features that can be conflicting with the ISDA class, some ISDMs are member of multiple ISDA classes. The remainder of this section describes each meta-group.

The first group is the set of groups are the Soft-Structured Analysis (SSA) ISDM' s. Literature identifies seven ISDM types, developed over six decades. Initially, this group of methodologies for ISD originates from a focus on solving technical problems. See in Figure 3-9 the Hardware Engineering Methods, and Software Craft Methods. However, these methodologies suffer from poor understanding of business requirements and problems arise because of the individualistic programming approach engineers apply (Avison and Fitzgerald, 2003). System development is basically perceived as a craft rather than an activity that can be planned, managed, and controlled. Programming information systems largely consists of *heroic debugging* (Boehm, 2006). Later advances as Structured Methods and Domain Understanding can be classified as *early methodologies* (Avison and Fitzgerald, 2006) aim to improve formality of the software development process (Boehm, 2006). In the following decade Business 4GL's / CAD/CAM / User Programming approaches are developed. These approached aim at productivity (Boehm, 2006), but cannot be considered full methodologies, unlike the Object Oriented methodologies. The latter group of ISDMs contains full blown methodologies that still aim at stage-wise development of software systems.

The second group is the set of the methodologies that is classified as Hard-Structured Analysis (HSA) ISDMs. Methodology sets in this

group is employed in development processes in which the world is considered to be systemic and can be studied systemically. In this approach, the ISD process is well-defined, has a single, optimum solution, and will be best executed using the scientific approach to problem solving. Early methodologies in this group are the Waterfall process or software life cycle design (SDLC) and formal methods approaches. Both these sets of methodologies aim for formality and implement the development process stage-wise (Royce, 1987) and a formal design for the system under development (Boehm, 2006). The development process is optimized for technical efficiency. The group of HSA methods that follow-up on these methodologies are aimed at improving productivity and streamline concurrent processes (Boehm, 2006). The methodologies in this group are firmly based on philosophical foundations. There are many methodologies developed that classify as member of this group. And, although they can be grouped in subsets, most methods are further extensions and improvements of the waterfall methodology, though there are many variations and different blends. We discussed the Object Oriented Methodologies and the Business 4GL's; CAD/CAM; and User Programming methodologies already in the SSA section. These methodologies also classify as HSA methodologies because they are full-blown methodologies: a collection of phases, procedures, rules, techniques, tools, documentation, management, and training required to develop a system (Avison and Fitzgerald, 2003). Other ISDMs in this group are the software maturity models. Maturity models, like Capability Maturity Model (CMM), which predict to deliver better end products, and better and more standardized development processes (Boehm, 2006). Next to these methodologies the Software Factories approach fits the HSA approach too. Software Factories is an organizational structure that specializes in producing computer software applications or software components according to specific, externally-defined end-user requirements through an assembly process. A software factory applies manufacturing techniques and principles to ISD to simulate the benefits of traditional manufacturing. Methodologies in this group too are criticized for not suiting the development process as well. For instance methodologies from this group are known to result in instable, inflexible,

dissatisfying, and poorly documented systems (Avison and Fitzgerald, 2006). But the main criticism on the methodologies in this group is that the predicted productivity gains are never met, methodologies are overly complex, and these methodologies are applied because of a *fetish for technique* (Wastell, 1996).

The Hard-Ill-Structured Analysis (HIA) ISDMs originate from the 1990 as evolutions of the HSA approaches. Methodologies such as the Domain Specific Software Architectures and Concurrent, Risk Driven Processes still are full-blown methods that aim at concurrent processes in the organizations. They are introduced because of a lack of scalability in the HSA ISDM's and refute the necessity of predefined structures in an ISD process flow and the need for formal notation (Avison and Fitzgerald, 2006). Successively they are followed up by Rapid Composition Environments, Service Oriented Architectures, Model Driven Development, and Integrated Systems and Software Design. These ISDMs are developed in the *post-methodology* era, in which a backlash against the traditional methodologies (SSA and HSA ISDMs) leads to less formal and more elective methodologies (Avison and Fitzgerald, 2003). The aim of these methodologies is to be able to develop information systems with agility by shortening the development process. These HIA ISDMs are developed to deal with rapid change in system's requirements because the ISD process is perceived as an ill-structured problem (Boehm, 2006). In Figure 3-9 HIA ISDMs overflow into the 2010 perception of ISDM; Collaborative Methods that aim at global systems integration. Main criticism on this group of ISDMs is the required skill set system developers need to have before they can apply the methodology.

The last group of ISDMs we discuss are the SIA methodologies, which aim at information systems' value and integration (Boehm, 2006). Many of methodologies in this group too build further upon earlier methodologies. A common denominator of these methodologies is that they are less formal, and more flexible (Avison and Fitzgerald, 2003). A well-known set of methodologies in this group are agile methods. Agile ISDMs, for instance Scrum, Extreme



programming, Feature-driven development, Adaptive software development, and Dynamic systems development method (DSDM), are designed to deal with rapidly changing market forces, systems requirements, implementation technology, and project staff (Cockburn and Highsmith, 2002). Furthermore, agile methods are aimed at the ability to implement every changing business condition real-time in the software development process (Abrahamsson et al., 2003). Methodologies that aim at global integration are often agile, hybrid methods, aimed at a collaborative approach with user involvement that develop overall architectures (Boehm, 2006). Agile ISDM's are a reaction to the traditional ways of developing information systems like the SDLC methodology (in the HSA approach). Since the traditional methods are documentation driven, heavyweight software development processes, the focus in these methods is on elicitation of a complete set of requirements, followed by architectural and high level design, development, and inspection (Cohen et al., 2004). There are two concepts underlying agile development: working code is a measure for project quality, and people working together like a well-oiled machine are most effective. Conboy carefully reconstructed the definition of agility based on multidisciplinary sources of literature and came to the following definition for agile development: "the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" (Conboy and Fitzgerald, 2004). Like the HIA approaches, SIA ISDMs are further developed and integrated with other methodologies to create Collaborative Methods that aim at global systems integration. Criticism on this group of ISDMs is predominantly the lack of scalability, and the unsupported parts of the ISD process, such as requirements specification, system test, and acceptance test (Abrahamsson et al., 2003). Conboy and Fitzgerald conclude that it is pointless to argue whether agile methods are superior to traditional methods. An agile method is only better when there is a need to be agile and the organization is capable of being

agile. What practitioners should do instead of just preferring agile, is making an assessment of the organizational capabilities and the agile needs before selecting a methodology (Conboy and Fitzgerald, 2004).

Most ISDM groups we discuss above are actively applied and further developed to date. In fact the majority of methodologies adopted by organizations are derived from ideas from the first two groups (SSA and HSA) we described above; from the early methodology era in Avison's overview, and the 1960's and 1970's in Boehms overview (Baskerville et al., 1992; Fitzgerald, 2000). These methods are designed when ISD was in its infancy and the very nature of these traditional methods reflects this point. Typical traditional methods take a rational, incremental approach to development with an emphasis on the ability to control each facet of development. Social-technical issues, flexibility, and direct relevance are less important (Kiely and Fitzgerald, 2005).

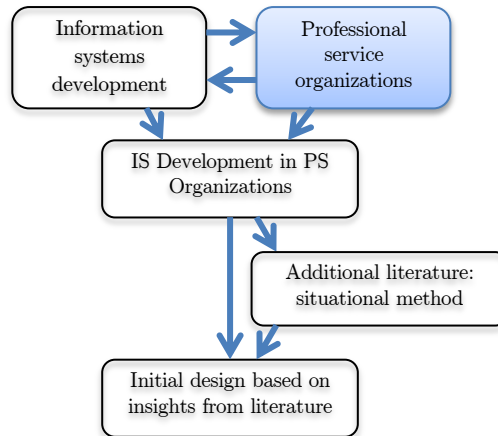
However literature shows that the traditional methodologies for ISD do not provide professional service organizations with the means to develop an effective information system (Brynjolfsson, 1993; Drucker, 2000). We therefore focus in this research on the SIA ISDM group, as we have selected to be the best fit for our research in paragraph 3.4.4. Members of this group are methodologies that are blends and hybrids of new and traditional methodologies, but are less formal and more flexible (Avison and Fitzgerald, 2006). These methodologies aim for improving business value and global integration (Boehm, 2006).

### **3.5 Literature: Professional service organizations**

This paragraph analyzes the organizational problem domain of this research. In the first chapter the lacking methodological support for information systems development in professional service organizations is identified as the main research problem. According to the action research perspective, the research's *problem domain* is professional service organization (the *technical solution space* is

information systems development methodologies, which is analyzed in the previous paragraph).

Professional service organizations (PSOs) employ predominantly professionals; employees characterized by their specific competence in a particular area (Bucher and Stelling, 1969). Organizational performance levels and the professionals'

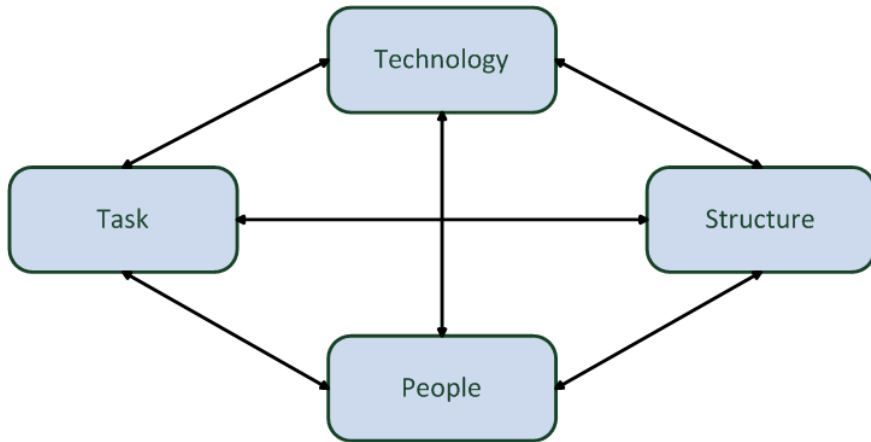


productivity are known to stagnate, relatively to other type of organizations (Brynjolfsson, 1993). Because of the ill-defined process flows (and sometimes even the lack of any process flows), a phenomenon prone to PSOs, problems arise when the professional employees have to co-operatively accomplish a task (Anderson and McDaniel Jr, 2000).

The aim of this paragraph is to determine the key characteristics of PSOs that are relevant to information systems development. This paragraph develops a typology of characteristics of PSOs and indicates which are relevant for ISD processes. The sections of this paragraph zoom in step-wise on the main goal, first by analyzing literature on organizational characteristics in general, then on organizational characteristics relevant for ISD, on PSO characteristics, and finally on PSO characteristics relevant for ISD in particular.

Characteristics of organizations is in the recent decades foremost discussed in literature in relation to specific areas of research, for instance in relation to MIS success (Raymond, 1985), inter-organizational networks (Burns and Wholey, 1993), job satisfaction (Adams and Bond, 2000), technology transfer (Håkanson and Nobel,

2001), generation of business ideas (Grandi and Grimaldi, 2005). In this literature a broad plethora of organizational characteristics is discussed, ranging from organizational culture, flexibility, and innovativeness to organizational structure.



**Figure 3-10: Diamond model for organizational characteristics, adapted from (Leavitt, 1978).**

Less recent literature provides us with a classification of the characteristics of organizations in general. Ginzberg describes a scheme of organizational characteristics that is based on Leavitt's work which originates from organizational psychology (Leavitt, 1978). Leavitt's diamond model consists of four major components of an organization; (1) the task, (2) the structure, (3) the people, and (4) the technology, which are interrelated and mutually adjusting, see Figure 3-10. In some later research the original model is expanded (Ginzberg, 1980; Yap, 1990), however these alterations of the model are not directly relevant to this research. Therefore Leavitt's model is used to characterize a professional service organization.

### ***3.5.1 Professionals versus complex systems***

Literature provides at least two options to regard a PSO: (1) organizations that are dominated by the presence of professionals or

knowledge employees; or (2) as complex (socio-technical) systems, a term first coined in organizational psychology. Both perspectives are analyzed in depth in this section.

First the perspective in which PSOs are characterized by the employment of professionals is analyzed. Bucher provides a list of characteristics in which PSOs are different from other organization types (Bucher and Stelling, 1969). This list is built around the notion that professional or knowledge employees that populate PSOs have a large degree of freedom. According to Bucher a PSO differs from other organizational types concerning the routineness of tasks that are executed by professionals. Because professionals tend to move through the organization, spontaneous internal differentiation occurs. Tasks are therefore often redefined. Furthermore, the professional builds its own role instead of fitting into a present role. Because of the diversity of tasks and roles, there is in a PSO a constant competition for resources. Bucher notes that integration in PSO is more a political process. The various professionals engage in a number of activities which are more appropriately described using the language of politics than the language of social structure. The last issue that defines a PSO from other organizational types is the distribution of power; since professionals move through the organization the power distribution shifts correspondingly.

The second perspective with which the characteristics of PSO can be depicted is the complex systems / complex socio-technical systems view (Beyer et al., 2004; Fujimoto et al., 2000). *Complex systems* is a term that originates from the statistical physics, information theory, and non-linear dynamics fields. It represents the organized but unpredictable behavior of systems of nature that are considered to be fundamentally complex (Gell-Mann and Lloyd, 1996; Waldrop, 1993). Implemented in the organizational science field, the term complex adaptive systems is often coined (Dooley, 1997; Holland, 1996; Lansing, 2003). Complex adaptive systems are complex in that they are dynamic networks of interactions and relationships; not aggregations of static entities. They are adaptive in that their individual and collective behavior changes as a result of experience

(Juarrero, 2002). *Socio-technical* refers to the interrelatedness of social and technical aspects of an organization. It is founded on two main principles; (1) the interaction of social and technical factors creates the conditions for successful organizational performance; and (2) optimization of each aspect alone (socio or technical) tends to increase not only the quantity of unpredictable, *un-designed* relationships, but those relationships that are injurious to the system's performance (de Sitter et al., 1997). Literature describes the characteristics of complex socio-technical systems in terms of the professional autonomy that is owned by the employees in an organization, which shifts the power from top level management to the operations (Ketchum and Trist, 1992; Trist and Bamforth, 1951); the adaptability of employees and teams to deal with changing requirements and process flows caused by the internal structure and environmental uncertainty (de Sitter et al., 1997), processes in complex socio-technical systems can be characterized by the high stochasticity, variability, heterogeneity, complexity, and the limited availability of resources (Harper and Shahani, 2002; Vandaele and De Boeck, 2003).

### 3.5.2 *PSO characteristics*

The aim of this section is to provide variables to characterize PSOs using Leavitt's model. This paragraph is structured according to the components of the diamond model for ISD, as is depicted in Figure 3-10. Therefore the four components that are likely to be relevant for ISD in PSO's are: (information) technology; tasks; people; and structure. Table 3-2 summarizes the characteristics that are discussed per component in detail in the remainder of this section.

<b>Information technology:</b> <ul style="list-style-type: none"> <li>• Means to coordinate in complex structures</li> </ul>	<b>Tasks:</b> <ul style="list-style-type: none"> <li>• Ill-defined processes</li> <li>• Conflicting goals</li> <li>• Unpredictable process flow</li> </ul>
<b>People:</b> <ul style="list-style-type: none"> <li>• Self-responsible professionals</li> </ul>	<b>Structure:</b> <ul style="list-style-type: none"> <li>• Split governance structure</li> </ul>

<ul style="list-style-type: none"> <li>• Professional autonomy</li> <li>• Productivity: quality is more important than quantity</li> <li>• Knowledge is an asset rather than a cost</li> </ul>	<ul style="list-style-type: none"> <li>• Low output efficiency</li> <li>• Value shop organization</li> <li>• Complex processes</li> </ul>
--	---

**Table 3-2: Summary PSO characteristics.**

### **(Information) Technology**

The use of information technology in PSOs is predominantly characterized as a means to cope with the complex coordination structures (Ren et al., 2008). Coordinating in PSOs is a challenging process (Kobayashi et al., 2005; Seagull et al., 2003). Of all highly complex organizations, PSOs like hospitals can be ranked as one of the most complex (Hafferty and Castellani, 2010). Operating in this complex environment is tough, and the result expressed as efficiency is hardly ever satisfying (Cutler, 2002; Feldstein, 1974; Newhouse, 1993). Key elements in operational efficiency are the means and success of *coordination* (Malone and Smith, 1988). Information technology is the main means for this.

In industry, ERP is introduced as IT solution for optimizing the complete value chain, and decrease inefficiency. Nevertheless, in specific PSO environments, ERP does not provide a suitable solution because ERP is (1) based on a bill of materials, instead of bill of resources (Roth and Dierdonck, 1995), (2) based on a deterministic planning, versus a stochastic and variable hospital environment (van Merode et al., 2004), (3) ERP makes it hard to measure throughput time, while costs of PSO is mostly based on consumed time (Roth and Dierdonck, 1995). ERP in the PSO domain though, is useful for strategic planning and master planning.

### **Tasks**

Tasks in PSOs have two distinct characteristics; ill-defined process flows and a high degree of conflicting goals within a process.

The process flow in professional organizations is very different of process flows often encountered in other organizational types, like manufacturing organizations. Often a standard process flow is hard to identify, and therefore organizational performance is hard to improve because of the ill-structured processes that can be identified as job-shop processes (Cutler, 2002; Harris, 1977). Processes in PSOs are characterized by their high stochasticity, variability, heterogeneity (Vandaele and De Boeck, 2003), complexity (Harper and Shahani, 2002), and their lack of resources. Because of deviation caused by the process characteristics – let alone human influences – a high efficiency level is not within reach.

Generally speaking two distinct planning approaches exist to deal with stochasticity and variability. (1) dynamic *first come, first served approach* combined with ad hoc incident coordination, and (2) static long term planning. As a rule, the former assumes the possibility to plan *production* at the very last moment, having all options at disposal till the point of decision. The latter alternative, the static long term planning, is intended as a rule of thumb with which the staff itself is up to use their creativity and initiative to fill in the best planning at the moment of execution. It should be noted however that heterogeneity, complexity and the lack of resources will remain obstacles, which are not solved with an improved planning technique.

Planning and goal setting are significant determinant of organizational performance (Smith et al., 1990). Individual employees combine forces in organizations for the purpose of attaining goals (Peters and Waterman, 2004; Simon, 1964). Planning is a fundamental element of organizational development (Schendel and Hofer, 1979). In essence planning is a rather straight forward activity that can be conducted on various levels, from operational to strategic. Planning is the organizational process of creating and maintaining a plan in order to achieve a goal.

Goals are objectives that the system should achieve through cooperation in the internal or external environment (van Lamsweerde, 2000) or, as Anton states it, goals are high-level



objectives of the business, organization or system, they capture the reasons why a system is needed and guide decisions at various levels within the enterprise (Antón et al., 1994). Goals may be formulated at different levels of abstraction ranging from high-level strategic concerns to low-level technical concerns. Goals also cover different types of concerns, functional and quality concerns.

Planning and goal setting in PSOs is a complex procedure because of the a lacking standardized process flow (Cutler, 2002), and upfront undefined goals (Drucker, 1999).

## People

Knowledge workers or professionals are the main employees of PSOs. Knowledge workers are characterized by their high level of professional autonomy; they identify, plan, and execute their supposed task according to their own best insights. Problematic in such an environment is the measurement of productivity. The requisite people skills for professionals in the context of ISD for the organization predominantly is determined by the measurement and management of the employees productivity.

From an economic perspective employee productivity is determined as added value per labor hour. But in professional organizations other elements next to this pure economic perspective play a role.

Drucker states that the PSO output majorly is determined by the individual employee's productivity (Drucker, 1999). A professional employee makes claims to competences in particular areas. "*He claims that he, uniquely, possesses the knowledge and skills to define problems, set means for solving them, and judge the success of particular courses of action within his area of competence*" (Bucher and Stelling, 1969). Professionals can be characterized as *knowledge workers* (staff purely involved with handling information) and *technologists* (staff applying highly specialized information to manual processes). Drucker identifies a number of factors influencing this productivity:

- The knowledge worker's task needs to be determined upfront (this is not self-evident)
- Knowledge workers are self-responsible for their productivity. This implies that they have to manage themselves. This means knowledge workers are required to have autonomy.
- Productivity of knowledge workers is partly determined by the output quantity, but foremost by the quality of their output.
- The knowledge, knowledge workers need to fulfill their task is considered to be an asset rather than a cost.

These issues are quite the opposite of what is needed to improve the productivity of manual workers. The approach to the design and implementation of supporting systems and management in professional organizations therefore should be based on a different approach, focusing on the key element of these organizations and their professional employee.

### **Structure**

*Governance.* PSOs are characterized by a difficult governance structure, in which managers and the board of directors have limited influence on the operations, and the knowledge workers only marginally have interest in the financial administration and the financial wellbeing of the organization (Glouberman and Mintzberg, 2001a). The governance structure is setup to protect the knowledge worker from behaving as an economic man (Harris, 1977). This results in a fundamental setting of conflicting goals (Cutler, 2002). Management and board try to optimize on output performance measures, knowledge workers aim to improve the in-process performance measures. The aim of management could result in an overemphasis on KPIs (Propper et al., 2008), the aim of the knowledge workers could result in a flat line optimization where extra investments do not result in extra quality or output (Enthoven, 1980).

*Effectiveness.* The organizational structure of PSOs is characterized predominantly as a means to attain a high level of efficiency. The

most cited problem in professional organizations evidently is the lack of efficiency, or at least the need for improved efficiency, enforced by, among other things, financial restrictions. An ongoing challenge in professional organizations is the aim to improve the organizational performance (Anderson and McDaniel Jr, 2000; Bucher and Stelling, 1969; Greenwood et al., 2005). Professional organizations' inefficiency can be measured by: (increase of) through put time, (lack of) quality improvement, or (low) percentage of resource utilization (Stiglic and Kokol, 2005b). Traditional approaches to deal with these process characteristics are batching, buffering, and sequencing the process tasks.

Although it is clearly a construct of central importance, organizational effectiveness is hard to define. Quinn et al. identified this problem (coining the term *effectiveness* where we use the term *performance*): “ *While effectiveness is clearly a construct of central importance, it is not without problems. One of the major problems pertains to the elusiveness of a definition*”. To cope with this problem Quinn designed an empirical derived spatial framework for the various definitions for organizational effectiveness or organizational performance. In this model two dimensions are identified; the organizational focus that can be internally or externally directed, and the organizational structure that can be aimed at stability or flexibility (Quinn and Rohrbaugh, 1983). The result of this approach is a framework with four quadrants, each representing an organizational value, see Figure 3-11. In each quadrant the criteria for organizational effectiveness are listed; (1) the human relations model in which cohesion and morale are means to attain human resource development; (2) the open system model in which flexibility and readiness are means to attain growth and resource acquisition; (3) the internal process model in which information management and communication are means to attain stability and control; and (4) the rational goal model in which planning and goal setting are means to attain productivity and efficiency (Quinn and Rohrbaugh, 1983).

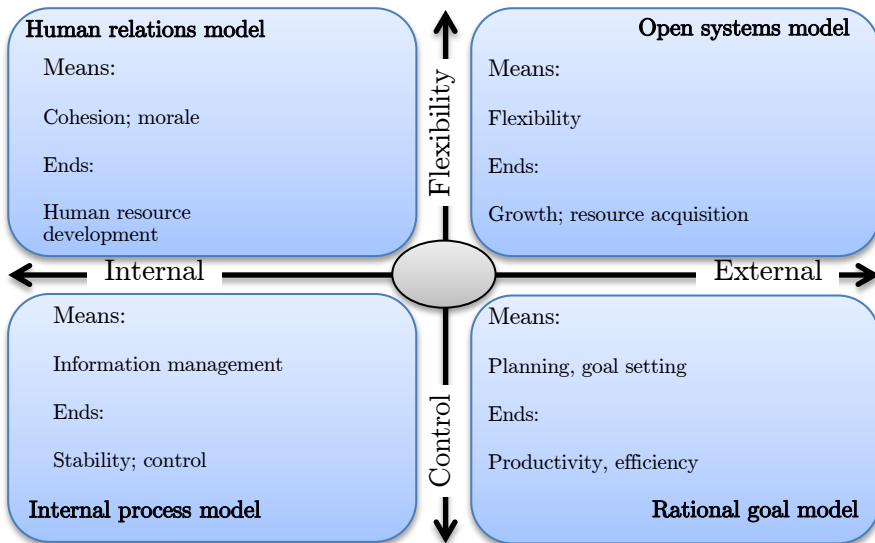


Figure 3-11: Spatial model of effectiveness criteria, adapted from (Quinn and Rohrbaugh, 1983).

An overemphasis on any of the models can result in dysfunctional organizations. An overemphasis on the human relations model, for example, can lead to an "irresponsible country club." Likewise, an overemphasis on the internal process model can lead to a "frozen bureaucracy"; overemphasis on the rational goal model can lead to an "oppressive sweat shop"; overemphasis on the open system model can lead to a "tumultuous anarchy" (Gifford et al., 2002).

Research has demonstrated that productivity and effectiveness are lacking in professional organizations relatively to manufacturing organizations (Brynjolfsson, 1993). A study of universities shows that the internal process and rational goal models were negatively related to trust, morale, equity of rewards, and leader credibility, and positively related to conflict, scapegoating, and resistance to change (Zammuto and Krakower, 1991). Other literature shows that a higher job satisfaction and more work promotions in organizations with greater human relations and open systems model (Quinn and Spreitzer, 1991). Yet another study finds that PSOs that adopt human relations and open system model primarily lead to the

successful implementation of total quality management (Shortell et al., 1995).

*Organization:* A common flaw in the analysis of PSOs is that these organizations are regarded as value-chains. Value chain analysis is perfectly suited for the analysis manufacturing organizations, in which processes are to large extent standardized, repetitive and predictable, but certainly less appropriate for health care organizations, which could be better characterized as value-shops. This is one of the reasons that health care organizations are for this research considered as professional service organizations. Value shops are organizations in which value is created by dynamically mobilizing resources and activities to resolve a (health) problem of a particular customer (in this case, a patient). The core activities of value-shop organizations are (1) problem finding and acquisition; (2) problem solving; (3) choice (4) solution execution; and (5) result evaluation (see Figure 3-12).

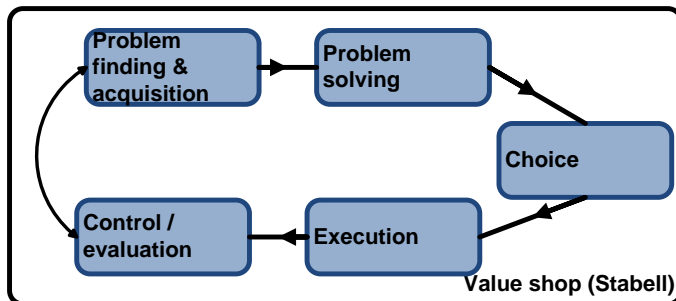


Figure 3-12. Value shop model

These activities could be iterative, cyclic, and applied in an interruptible manner. Furthermore, in most cases there is a significant sequential and reciprocal interdependence between activities and process instances, due to the various resources that have to be shared and/or allocated. whereas multiple disciplines and specialties are active in spiraling activity cycles (Stabell and Fjeldstad, 1998). Thus, a specific task can be regarded as a process instance, i.e., as a chain of value shop activities. In such a setting central coordination models hardly ever lead to optimal schedules

that remain valid, anyway, only for a very short time span before becoming obsolete.

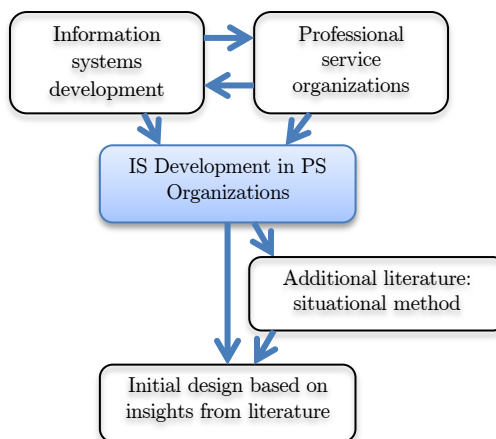
The chains of value shop processes, or as defined by Ren, process trajectories (Ren et al., 2008), are prone to be heavily interdependent. When these interdependencies are not synchronized, coordination breakdowns may occur. The information system in this case does not support proper coordination. The resources and staff in a PSO can be sequentially interdependent, reciprocally interdependent, or pooled interdependent. For a PSO value shop chain to work effectively, safe, timely, and with high quality, the different trajectories need to interweave and intersect at specific times (Ren et al., 2008). To accomplish this, the planning has to be adapted to last minute changes in the overall process. These changes have to be coordinated with multiple groups (which, in many PSOs, have no intentions to do so) (Ren et al., 2008).

*Process characteristics:* Next to the prior discussed organizational structure, the process structure also causes much complexity for a PSO. PSO processes are characterized by their high stochasticity, variability, heterogeneity (Vandaele and De Boeck, 2003), complexity (Harper and Shahani, 2002), and the scarcity of resources. Because of deviation caused by the PSO process characteristics – let alone human influences – a high efficiency level is not easily within reach. In fact, the most cited problem in PSOs evidently is the lack of efficiency, or at least the need for improved efficiency. When we look from the viewpoint of the PSO itself, process inefficiency can be, amongst others, measured by: (increase of) waiting time, (lack of) quality improvement, or (low) percentage of resource utilization (Stiglic and Kokol, 2005a). Traditional approaches to deal with these process characteristics are batching, buffering, and sequencing the operations. However, these measures are effectuated in departmental budgeting, deterministic planning, and control. Therefore, significant resources are required to communicate planned and last-minute activities. Operating in this complex environment is tough, and the result expressed as efficiency

is hardly ever satisfying (Cutler, 2002; Feldstein, 1974; Newhouse, 1993).

### 3.6 IS Development in PS Organizations

Both literature and empirical analysis reveal that the soft systems – ill-structured analysis is the best fit with the PSO domain. This section summarizes the positioning of ISD in PSO according to the model of ISD derived from literature that is developed in this paragraph.



#### Paradigm:

In light of the professional organization domain, in this research the *neohumanist* paradigm on systems development is adopted. Hirschheim states that in this paradigm “ information systems are developed to remove distorting influences and other barriers to rational discourse” .

ISD is governed by three knowledge interests. (1) The technical knowledge interest directs the developer to be sensitive to issues associated with effective and efficient management of the system project. (2) The interest in mutual understanding directs the developer to apply the principles of hermeneutics, which examine the rules of language use and other practices by which we improve comprehensibility and mutual understanding, remove misunderstandings, and disagreement or other obstacles to human communication. (3) The knowledge interest in emancipation directs

the developer to structure system development to reflect the principles of rational discourse (Hirschheim and Klein, 1989).

The neohumanist paradigm provides us with the perspective to perceive ISD as a means to resolve the state of perpetual conflicts in professional organization.

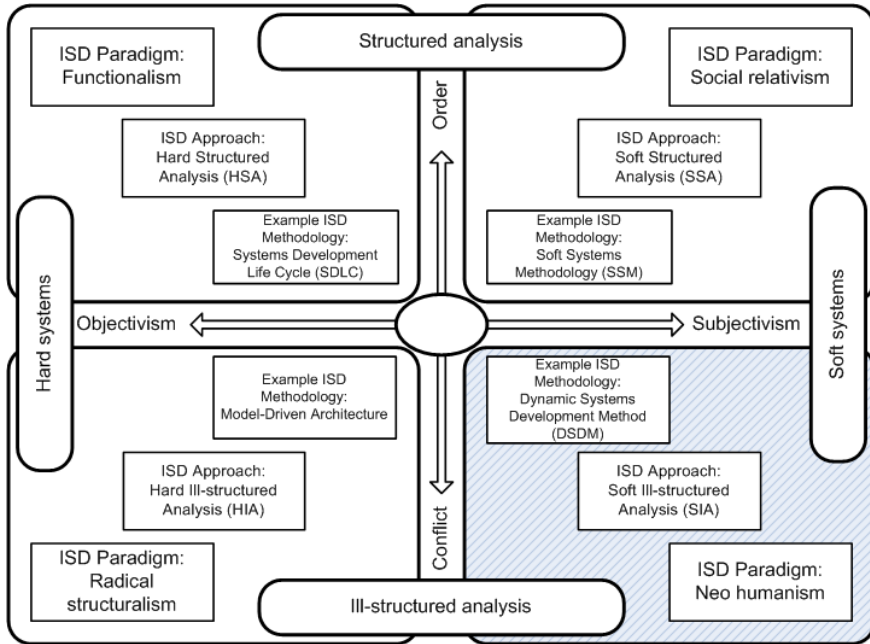


Figure 3-13: Research positioning based on ISD paradigm, ISD approach, and ISD methodology.

### ISD Approach

ISD in professional service organizations is a combination of ill-structured analysis, and problematic development processes, and because of the characteristics of professional service organizations (complex coordination, process characteristics, conflicting goals, and knowledge workers autonomy) this research is focused on the SIA ISDA. This ISDA frames the development of an IS as an ill-defined problem that is addressed using an ill-structured process that aims for functionality over formality.



## ISD Methodology

The positioning of this research is based on the ISD paradigm, ISD approach, ISD methodology that are identified to be most suitable for the PSO domain. The selected approach implies that this research perceives the development of IS in professional service organizations as ill-structured problems, that can be best addressed by a soft-systems methodology.

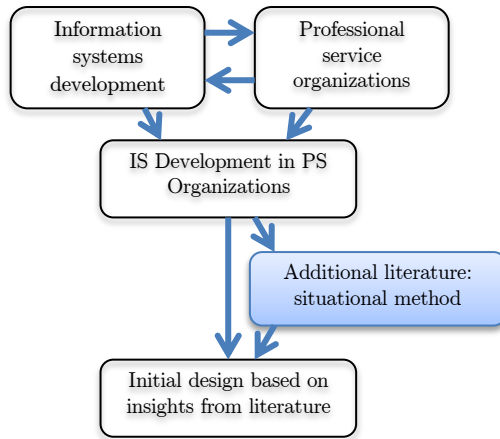
Because literature demonstrates that full-blown methodologies do not live up to the demands of realistic ISD projects, ideas from situational method engineering are applied as a guideline to develop method chunks that fill the gap in ISD for professional service organizations.

This approach aligns with literature, since most available methodologies for ISD have some gaps in them or, if not complete gaps, they have areas that are treated much less thoroughly than others (Avison and Fitzgerald, 2006). Especially SIA ISDMs are prone to incomplete coverage of the development process, as is demonstrated in the previous sections. Partially this is caused by a focus on methodological flexibility (Abrahamsson et al., 2003), and partially because of backlash against over formal, extensive methodologies that occurred as a reaction on the many over-formal methodologies in use in the 1990s (Wastell, 1996).

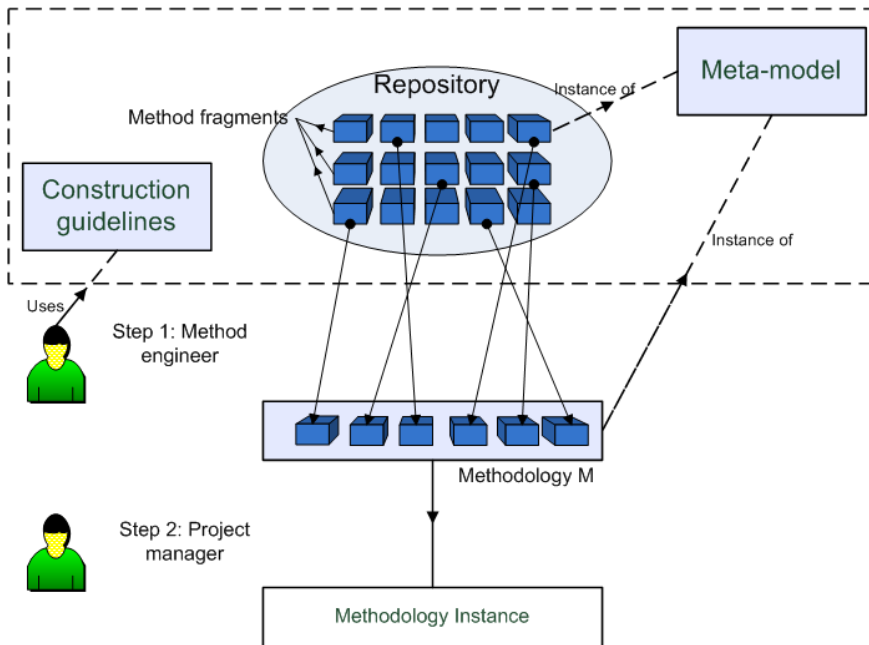
### 3.7 Literature: Method engineering

This paragraph discusses the implications of contingency theory, the primary being that ISD methodologies are to be adapted to organization specific conditions. Next the implications of contingency theory are implemented using (situational) method engineering ((S)ME). ME is an approach to select or configure a most applicable methodology for a specific ISD project.

In the IS field, the notion of a one-size-fits-all methodology has to be hard to implement, and as a result the concept of Method Engineering (ME), or Methodology Engineering, is developed. A state-of-the-art review conducted in 2010 provides insight in this particular field of research



(Henderson-Sellers and Ralyté, 2010). ME is defined as the engineering discipline to design, construct, and adapt methods, techniques and tools for systems development. The approach taken in ME is to select methods or parts of methods, which are already stored in a method base, and combine them to build a complete development methodology. When ME is applied to develop in-house a methodological approach that is organizational specific or project specific we define this as Situational Method Engineering (SME). SME is a major component of ME, encompassing all aspects of creating a development method for a specific situation. It is indicated to be a solution for the problem of selecting the most appropriate method for an ISD project (Henderson-Sellers and Ralyté, 2010). In the SME thinking, a methodology can be defined as an approach to perform an ISD process, based on a specific way of thinking, consisting of direction and rules, structured in a systematic way in development activities with corresponding development products (Brinkkemper, 1996). A very simplistic graphical overview of SME is presented in Figure 3-14. Method fragments and method chunks are stored in a method base. Method fragments and method chunks are selected based on project characteristics and assembled to engineer a situational methodology for ISD.



**Figure 3-14: Multi-level hierarchy of (situational) method engineering. Adapted from (Cossentino et al., 2006).**

The concept of engineering a methodology based on situational dependencies strongly relates in literature to contingency theory (Donaldson, 2001; Shenhar, 2001). As is discussed in paragraph 3.1.2, contingency theory concludes that the ISD process, the IS, the IS performance, and the organizational performance can be characterized with a set of different variables, therefore the method should be contingent upon various internal and external variables that influence the process outcomes.

### ***3.7.1 Situational method composition***

The situational method composition is not aimed at configuring one single base method, but at combining and aggregating several method chunks in order to develop a new methodology (Brinkkemper, 1996; Punter and Lemmen, 1996; van Slooten and Hodes, 1996). The composition process is subdivided into three phases (Bucher et al., 2007): (1) identifying the situational

characteristics: the characteristics that can be used to characterizing specific development project types; (2) decomposing generic artifacts into artifact fragments: the artifact fragments, which are generic artifacts specified according to the situational characteristics from step 1, are used to fill the method base; and (3) composition of the artifact fragments into a situational method: the situational method is composed by application of a well-defined process that follows construction and composition principles to fit the situational characteristics of the development project.

In this research the first stage of the situational method composition process is of high relevance. However, literature indicates that this stage is not well-researched. The problem is that in literature the concepts *context* and *project type* are not clearly defined, and therefore often used interchangeable (Bucher et al., 2007). One approach taken in literature is based on a list of seventeen contingency factors whose values influence the project approach (van Slooten and Hodes, 1996). According to Bucher, this list comprises characteristics that are primarily external to the method application. This implies that the contingency factors describe the environment to which the method is adapted and in which it is deployed. Bucher proposes to include both *project type* and *context* as factors that describe the organizational characteristics in which the methodology under-development will be applied (Bucher et al., 2007). In the meta-model for SME that is described in section 3.7.2 both elements are included.

### ***3.7.2 Method engineering meta-model***

For analyzing ME from a more abstract perspective, the ME meta-model is relevant to include. The meta-model presented in Figure 3-16, is a result of an elaborate discussion in literature. Initially five constituent elements of a method are classified based on a review of different approaches to method construction and method implementation. These elements are *design activities*, *documents specifying design results*, *roles*, *techniques*, and an *information model* for the method. This list that can be used to describe a generic

model, is validated by analyzing scientific contributions to the field of ME (Braun et al., 2005). Therefore, these elements are the core of the ME meta-model. To extend this ME model to the SME domain, Bucher adds the concepts *adaptation mechanism*, *context*, and *project type* (Bucher et al., 2007). The latter concepts regard situation as a combination of *context* and *project type*. In Figure 3-15 *method fragment* is introduced to connect the *adaptation mechanism* concept in the model. As will be discussed in section 3.7.3, a combination of the concepts *design activity* and *technique* is denoted as a *method fragment*. Design activities describe the task that have to be executed (the *what*), and techniques provide options to achieve the results (the *how*) (Bucher et al., 2007).

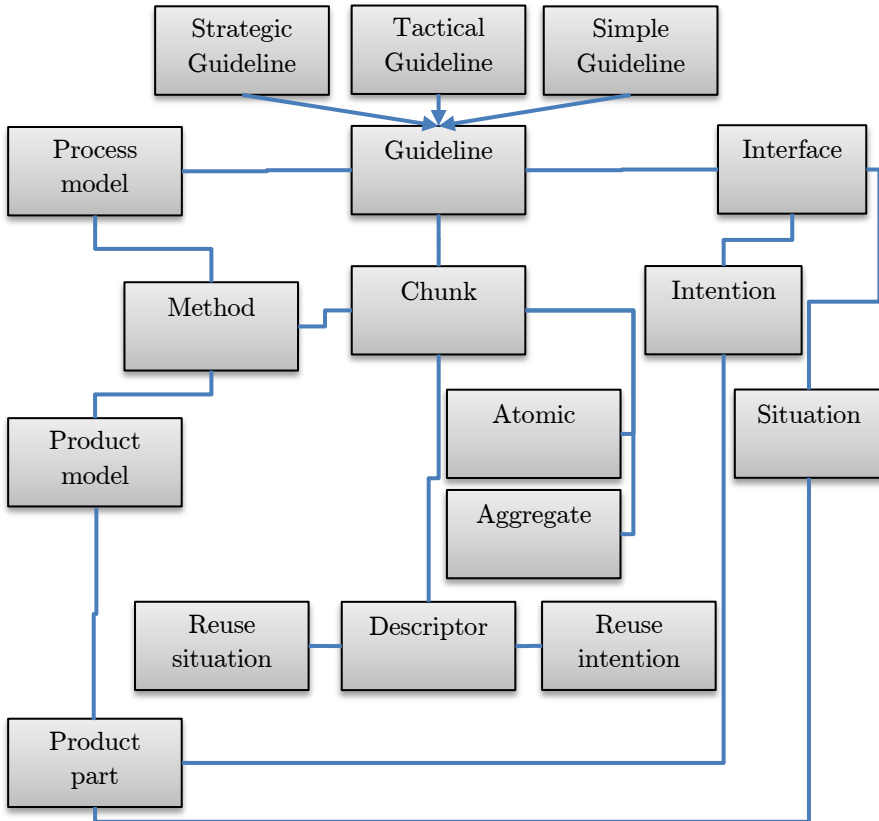


Figure 3-15: Method engineering meta-model. Adapted from (Cossentino et al., 2006)

Since this research follows up on SME to design a methodology dedicated for professional service organizations, the meta-model for SME is the foundation for the IT artifact. Therefore the meta-model for SME is referred to in paragraph 3.6.

### ***3.7.3 Method chunk and method fragment***

SME aims to develop a methodology that is constructed from smaller components, called method fragments or method chunks. There is no clear consensus in literature about the distinction between method chunks and method fragments (Ågerfalk et al., 2007). For this research we adopt two definitions:

*A method fragment is an atomic methodology element, either product or process oriented.*

*A method chunk is a combination of at least two method fragments, one of which product oriented and one process oriented.*

In this paragraph both concepts are discussed and the differences are pointed out.

#### **Fragment**

It is often mentioned in literature that a method fragment can be regarded an atomic element of a methodology, whereas a method chunk is a combination of multiple fragments (Mirbel and Ralyté, 2006; Ralyté, 2004; Rolland and Prakash, 1996). A method fragment is defined as a description of an IS engineering method, or any coherent part thereof (Harmsen, 1997). The term method fragment is introduced in (Harmsen et al., 1994b). A method fragment can be either process oriented (describes the stage, activities and tasks), or product oriented (deliverables, diagrams, etc) (Brinkkemper, 1996).

There are many views on the meta-level of method fragments, see (Cossentino et al., 2006) for an elaborate discussion. For this

research the meta-model for method fragments as depicted in Figure 3-16 is selected. The important elements of a fragment are the product and the process aspects, the presence of a stakeholder, and the possibility of reuse, which is enabled with the guidance (Cossentino et al., 2006). A method fragment therefore consists of the elements *design activity*, *technique* (or guidance), *role* (or actor), and *design result* (or artifact) (Bucher et al., 2007).

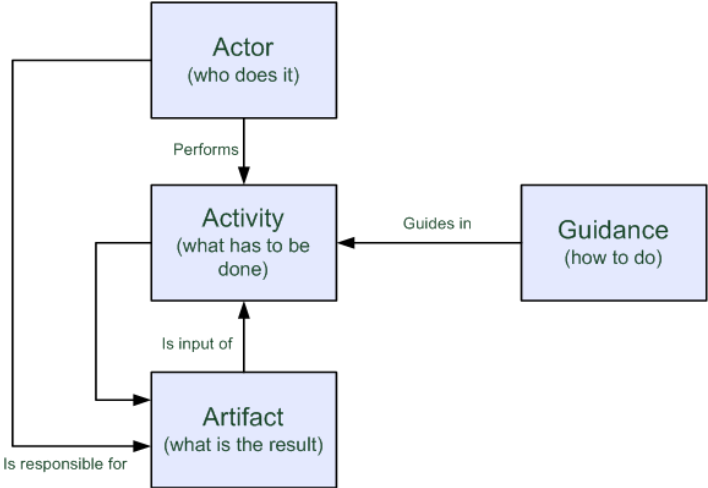


Figure 3-16: Method fragment meta-model. Adapted from(Cossentino et al., 2006).

The meta-model for method fragments is a subset of the meta-model for SME as depicted in Figure 3-15. In the meta-model for fragments elements like *context* and *project type* are left out.

**Chunk**

Since an overall methodology has to interdependent aspects: product and process (Rolland et al., 1999), a method chunks is a combination of, at least, one product oriented fragment and a process oriented fragment, the process part being a guideline for the product focused part (Henderson-Sellers and Ralyté, 2010). A method chunk is a tightly coupled representation of a process and product fragment (Ralyté and Rolland, 2001a). The product part of the method chunk includes all product elements necessary for the process part

execution: the input and output product elements (Henderson-Sellers and Ralyté, 2010).

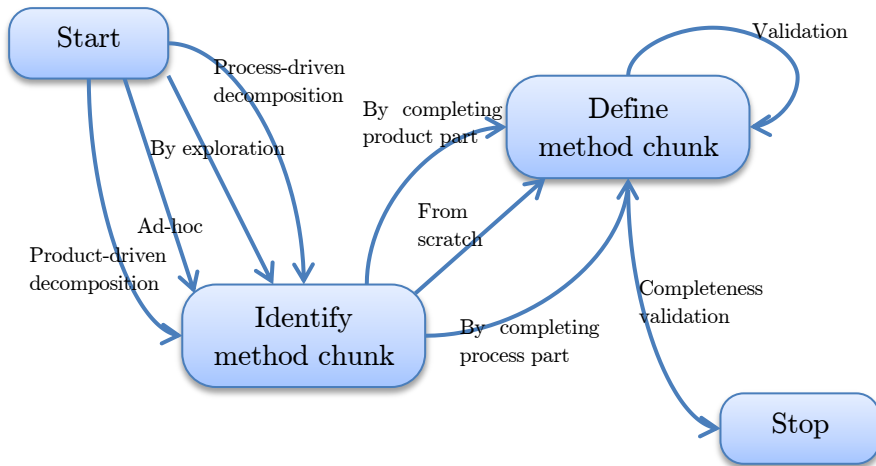
It is important for both method fragments as for method chunks to include knowledge about the context of use of the method fragment/chunk in a formal way within the method base. *Context* represents both the situation (i.e. project type and organizational context, see paragraph 3.7.2) and decision. This implies that the knowledge about the situation in which it is relevant and the associated decision that can be made in such a situation (Rolland and Prakash, 1996).

#### ***3.7.4 Developing method fragments and chunks***

Both the development of a reference process or even a method for methodology construction as well as for method fragments is one of the main challenges of the SME field (Bucher et al., 2007; Henderson-Sellers and Ralyté, 2010). Many authors propose to extract method fragments from existing methodologies (Brinkkemper et al., 1998; Harmsen, 1997; Ralyté and Rolland, 2001a), but offer no solution in how to do that (Henderson-Sellers and Ralyté, 2010).

This research focuses on the development of method fragments in particular for the PSO domain. Henderson et al. list in an extensive literature review the various ways to identify or construct method fragments (Henderson-Sellers and Ralyté, 2010). There are two mainstream approaches to identify or construct a method fragment. The first one is to create method fragments by reverse-engineer existing methodologies (either process-driven, or product-driven), the second approach is to create method fragments from scratch. In Figure 3-17 both strategies are mapped in a process model for the development of method fragments. The third option – by exploration – is a rather theoretic approach, hardly ever applied (Henderson-Sellers and Ralyté, 2010).





**Figure 3-17: Process model for method fragment / chunk construction. Adapted from (Ralyté, 2004).**

### Method fragments from existing methodologies

A major part of SME literature focuses on the identification of method fragments from existing methodologies (Ralyté, 2004). For the sake of reverse-engineering existing methodologies, according to literature the methodologies can either be considered from a process perspective or a product perspective. The former defining a methodology as a sequence of processes, the latter as a list of deliverables or products (Ralyté and Rolland, 2001a; Rolland et al., 1999). Since the process-driven view is more likely and more powerful, the process perspective is the most discussed view in literature (Ralyté, 2004).

There are basically two approaches to reverse-engineer a methodology into method fragments: (1) decomposition (process driven or product driven): modularization of the complete methodology; and (2) exploration: analyzing how to use the same model in different ways. Both approaches are depicted in Figure 3-17. The third approach (ad-hoc) is discussed in the next section.

Since most ISD methodologies are not modular built, reverse-engineering these methodologies can be a problem (Ralyté, 2004).

### **Method fragments from scratch**

Construction method chunks and fragment from scratch (or *ad hoc* as in Figure 3-17) is primarily useful for supporting new technologies or new domains (Henderson-Sellers, 2005; Henderson-Sellers et al., 2002). In case of constructing fragments and chunks from scratch, theory and best practice permits the initial identification of fragments and chunks. After the initial identification literature dictates that the newly constructed fragments are evaluated in practice, refined, quality assessed in an iterative fashion until quality standards are met (Henderson-Sellers and Ralyté, 2010). This approach is, of course, similar to the ADR approach that is applied in this research, which makes it highly useful and applicable during this research.

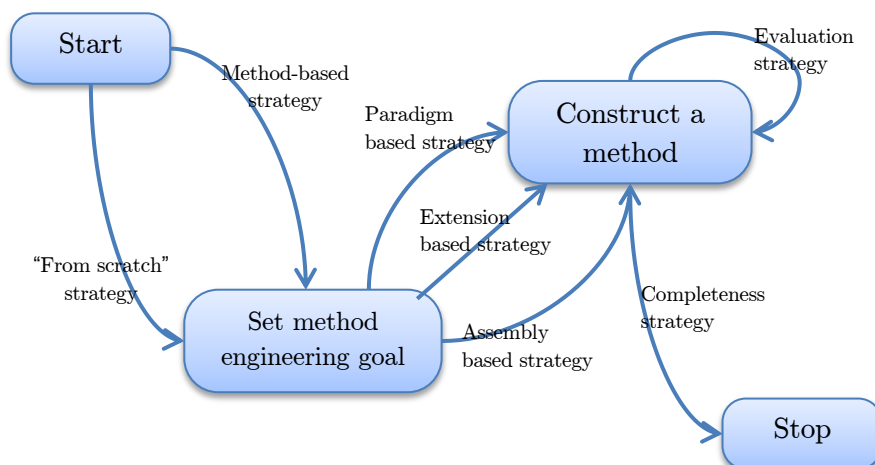
#### ***3.7.5 Creating a modular methodology from fragments***

The goal of this research is not to develop an overall methodology for ISD in PSOs; this research is aimed at developing a method to construct method chunks and method fragments that could be part of an overall methodology for ISD in PSOs. Nonetheless it is important to define the ultimate goal of the fragments and chunks that can be developed with the *IT artifact* that is the subject of this research.

The creation of a methodology based on the SME approach can be done using various approaches - corresponding to the stages of the SME spectrum, see (Harmsen et al., 1994a) - however in this research the focus is on modular SME. Therefore only the approaches to construct a methodology from fragments, which fit this stage of the spectrum, are discussed in this paragraph.

The generic process model for SME is rather straight forward: a methodology is an instance of the SME meta-model, and consists of several method chunks (Ralyté and Rolland, 2001b). First the

*method engineering* goal is set. This can be done either based on other methodologies (*method-based strategy*) or a completely new methodology (*from scratch strategy*). The second stage in the process model is to *construct a method*. There are three strategies identified for this stage (Ralyté et al., 2003): (1) *Assembly-based*: using method fragments and chunks that are stored in a method base to develop the methodology (Kornysheva et al., 2007); (2) *Extension-based*: using patterns applied in existing methodologies to develop the methodology (Deneckere et al., 2008); and (3) *Paradigm-based*: either abstraction-based, instantiated from a meta-model or adapted. Figure 3-18 depicts the three alternatives.



**Figure 3-18:** Generic process model for SME. Adapted from (Ralyté, 2004; Ralyté et al., 2003).

Only one strategy for the construction of a methodology fits this research: the *paradigm-based* approach. The *extension-based* approach and the *assembly-based* approach do not align with this research because these approaches assume an already filled method-base (Ralyté et al., 2003).

In the paradigm-based approach next to the set of method fragments and chunks there is also a set of *route map* elements included in the development of the methodology with SME (Henderson-Sellers and

Ralyté , 2010). The route map elements include strategies, activities, products, and project management elements. Characterizing for the paradigm-based approach to SME is that the method fragments are predominantly created rather than abstracted from existing methodologies, as is more usual in the assembly-driven approach (Ralyté et al., 2003). The method fragments that are assembled in the paradigm-based approach are generated based on a meta-model (Henderson-Sellers and Ralyté , 2010). This approach is a better fit for the introduction of new technologies or organizational domains for which pre-existing fragments in the method base are insufficient to construct a methodology (Han et al., 2008). In this case fragments that are required need to be created from a meta-model. The process model for the paradigm-based strategy to *construct a method* is depicted in Figure 3-19.

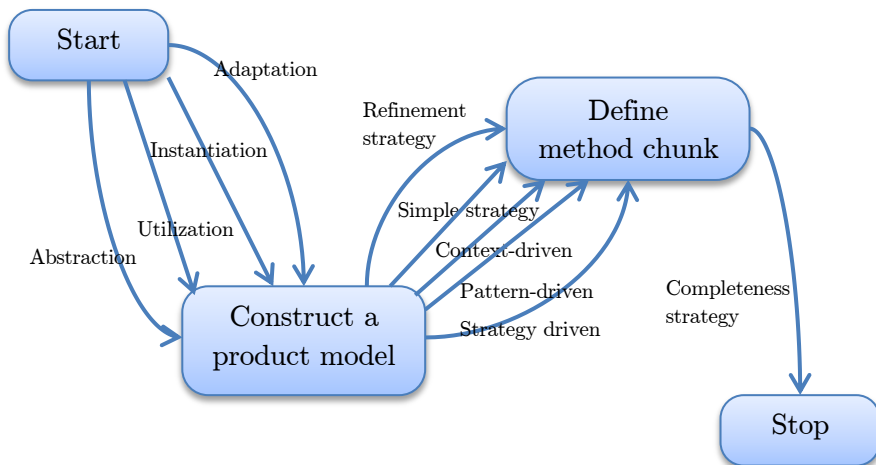


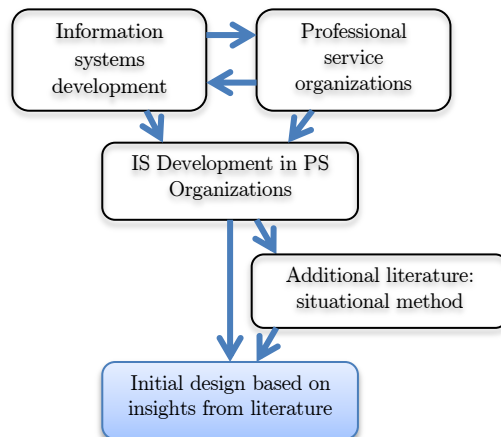
Figure 3-19: Process model Paradigm-based approach to SME. Adapted from (Ralyté et al., 2003).

In this approach predominantly method *chunks*, rather than fragments, are used to construct a methodology. Since chunks are either product oriented or process oriented (see paragraph 3.7.3) both the process model and the product model of the methodology has to be developed to construct a methodology. Therefore the construction of a methodology using the paradigm-based approach to

SME consists of two stages: the construction of a product model and the construction of a process model, see Figure 3-19. After the product and the process model are developed, both models are integrated into the complete methodology. Because this approach is the best fit for this research, the strategies for developing the process and product model are discussed in more detail in paragraph 3.6. The approach to develop the product and the process model namely highly influence the development of the method chunks and fragments.

### 3.8 IT artifact: Initial design

In the first part of this chapter the various theoretical backgrounds of this PhD research are addressed. In this paragraph the initial IT artifact, as is discussed in chapter 2, is developed based on the literature that is discussed in this chapter.



In the following chapters (chapter 4 and beyond) the initial design is further developed in correspondence to the ADR approach. In the first section of this paragraph the outcomes of the literature review are discussed that are relevant to the artifact development. In this paragraph's next section an initial design for a domain specific approach to develop method chunks is presented. In the final section of this paragraph design principles are listed that should aid when applying the initial design in the building, intervention and evaluation stages in chapter 4 and 5.

### 3.8.1 Literature review outcome

In the previous paragraphs it has become clear that PSOs are very characteristic with regard to information technology, tasks, structure, and people. Because the professionals that employ PSOs have a professional autonomy, encounter tasks that are hard to define, and value quality more than quantity, efficiency is hard to attain. Furthermore, processes are ill-defined, highly stochastic, variable and complex, which makes coordination and communication complex. Information systems are the means to cope with this complex organizational structure.

Information systems development is a complex process for which the approaches over the years have been evolving. Although distinct paradigms, approaches, and methodology directions can be clearly identified, still one of the major criticisms on methodologies for ISD is the lacking adaptability for specific projects and domains. Furthermore the broad coverage and structured approach of many methodologies has led to a *backlash* against methodologies in organizational domains.

Contingency theorists stress the need to align contingency variables (organizational characteristics such as structure and size) with IS variables (such as management and implementation). These variables ought to be aligned with IS performance and finally with organizational performance. If an information system is well implemented, according to contingency theory, this will pay-off in organizational performance. To ensure the quality of implementation of IS, the implementation methods are to be aligned with the organizational characteristics.

To cope with the criticisms on ISD methodologies, (situational) method engineering is introduced. In this approach, methodologies are adapted to specific organizational conditions. In this chapter the most radical variety of SME is selected as the most suitable for this research. In this approach methodologies are modular constructed, using method fragments and chunks. Both method fragments / chunks and methodologies that consist of an assembly of chunks, are

based on a meta-model. SME research hiatus are in approaches to construct method fragments, and the construction of domain specific applications of meta-model based methodologies.

Action design research tries to find solutions for organization specific problems. By creating solutions to these problems ADR also tries to find a solution for the class of problems the instances are part of. In the ADR paradigm problems in specific organizational settings are instances of a larger class of problems. In the following chapters of this research two ISD projects in PSOs are analyzed to construct method chunks, tailored to the specific needs of the organization. These two cases are instances of the class of problems. The class of problems is the development of method chunks for PSOs. Since this research is conducted according to the ADR approach, the problem statement and corresponding solution design is detailed further during the course of the research.

### ***3.8.2 Domain specific approach to construct method chunks***

This section describes the initial design of the IT artifacts. The IT artifact that is proposed in this section is a method to develop method chunks. These chunks address domain specific / organizational specific needs when developing an IS. As is described in paragraph 3.6, IS development in a professional service organization most probably will have to deal with ill-defined problem statement, with a process that is ill-structured. Prebuilt methodologies that cover every aspect of IS development are therefore less applicable.

Figure 3-20 presents a general process flow of the initial design of the method to develop method chunks. Basically the initial design of the method, that is developed in this research to address IS development in professional service organizations, consists of four major activities: (1) the identification of the contingency variables; (2) the definition of the method chunk's scope; (3) selection of method fragments and assembly hereof into a method chunk; and (4) implementation of the method chunk into the overall methodology. Table 3-3 presents a

slightly more detailed list of activities related to the four main stages in the process.

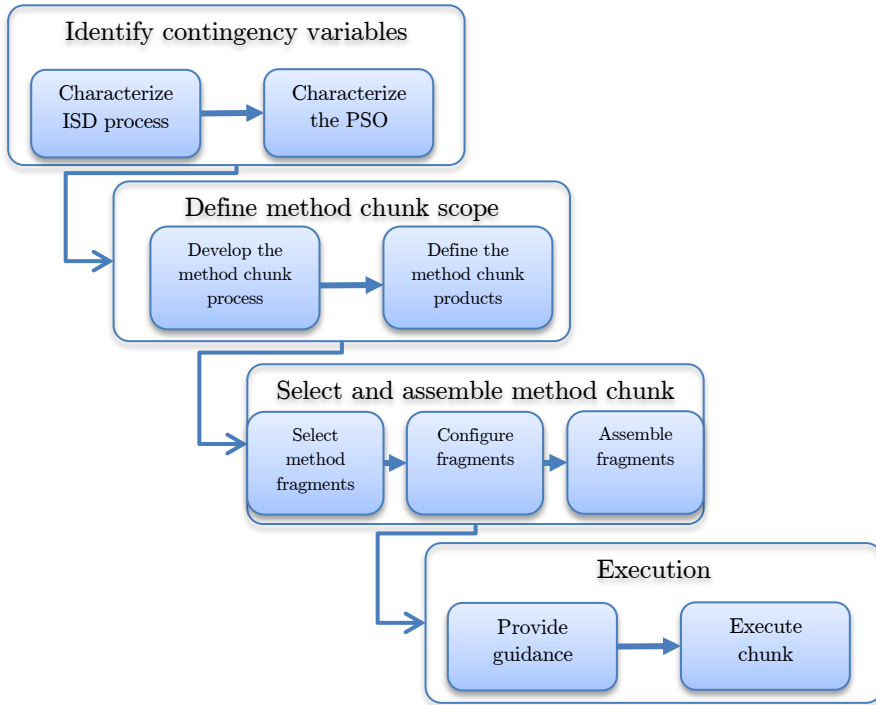


Figure 3-20: Approach to construct method chunks; process model.

Identify contingency variables	Characterize ISD process
	Characterize PSO under-review
Define method chunk's scope	Develop the process of the method chunk
	Define the products of the method chunk
Select and assemble method chunk	Select fragments
	Configure and assemble fragments into chunk
Execution	Provide guidance on how to use the method fragment
	Implement the method chunk

Table 3-3: Stages in the initial design to develop method chunks.



The approach to develop IS for the PSO domain that is developed in this research is based on contingency theory, as is discussed in paragraph 3.1.2.

Situation: Intention Origin Objective Type Aggregates	
Problem discription	
Role definition	
Process description	Product description

**Figure 3-21: Outline of a method chunk description. Adapted from (Deneckere et al., 2008; Ralyté and Rolland, 2001a).**

Using contingency theory as a foundation implies that for this research there is an assumed relation between the contingency variables, IS variables, IS performance, and organizational performance. Therefore to create a method chunk that successfully addresses the specific needs in the IS development process, the method chunk characteristics should be aligned with the organizational characteristics.

There are some general concepts in literature of the outline of a method chunk; Figure 3-21 depicts the outline adopted for this research.

### ***3.8.3 Design principles***

Brinkkemper described a list of rules for the construction of methodologies using the SME approach (Brinkkemper et al., 1998).

A subset of these rules are relevant for the construction of method fragments and method chunks. Table 3-4 displays the rules that apply to method chunk construction.

Rule	Description
New concept	There should at least be one concept newly introduced in each method fragment
Linking concept	There should at least be one concept linking the two fragments to be assembled
Combination	Any technical method fragment should be supported by a conceptual method fragment

**Table 3-4: Rules for constructing method fragments and chunks. Adapted from (Brinkkemper et al., 1998).**

Based on the literature review in this chapter, specifically in the perspective of the analysis of professional service organizations, some design principles can be added to the construction process of method chunks. Table 3-5 depicts the design principles and the consequences for method chunks.

Design principle	Consequences
End-user involvement	<ul style="list-style-type: none"> <li>- Chunks cannot be developed from a single perspective (expert view)</li> <li>- Development and implementation of a chunk only starts when users confirm the need for it</li> <li>- Regular interaction with end-users to evaluate development activities</li> </ul>
Upfront task definition	<ul style="list-style-type: none"> <li>- Before a chunk is developed based on the need for it, the contribution, process, and products of the chunk need to be defined</li> </ul>
Interest integration	<ul style="list-style-type: none"> <li>- Different stakeholder groups require individual consulting</li> <li>- Viewpoints of stakeholder groups need to be integrated during discussion sessions</li> </ul>

- Risk of the development of a chunk that, by addressing every group's interests addresses no interest at all

**Table 3-5: Design principles for a method to construct method chunks.**

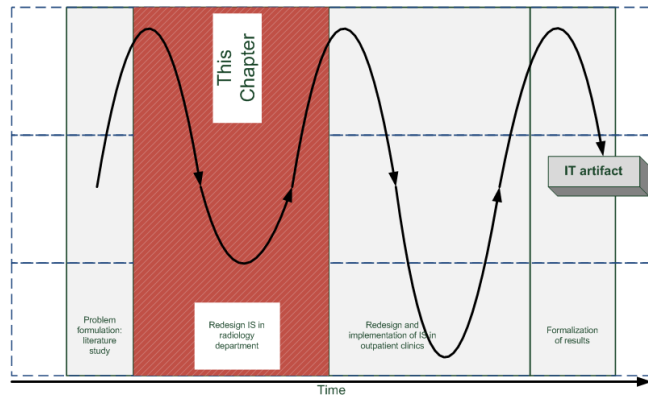
The literature review conducted in this chapter on the several topics that are relevant to this research reveals that successfully developing method chunks, which is the goal of this research, is effective when end-users / professionals are involved, tasks are defined before the process is initiated, and the various interests in the PSO are integrated in the method chunk development process.

In the next chapter the initial to method chunk construction is implemented using the defined design principles. The ARD stage is the BIE process (building, implementation and evaluation stage) in which the design is further enhanced by means of interaction with the organization.



## 4 Building: Redesign IS in radiology department

This chapter describes the second stage of the ADR approach, the building, intervention and evaluation (BI&E) stage. In this research the BI&E stage



is conducted twice, as depicted in the figure. In the second stage of the action design research the results from the first stage are used to further develop the initial design of the IT artifact (Cole et al., 2005; Hevner et al., 2004). Because this research adopts the IT-dominant approach to ADR (see chapter 2), the *IT-artifact* (organizational specific method chunk) in the first BI&E iteration only serves as a light weight intervention in a limited organizational context. The emerging artifact and the theory ingrained in it are continuously instantiated and validated through interventions into participating organizational members' assumptions, expectations and knowledge (Sein et al., 2010). In chapter 5 a method chunk is developed in a wider organizational setting.

In this chapter feedback from the intervention on the IT-artifact will be used to improve the initial design. The ultimate goal of the *building, intervention and evaluation* stage is to support an iterative process at the intersection of the IT artifact and the organizational environment. The aim of this stage is the realization of an embedded artifact. Characteristic of this stage is the continual design and evaluation as a result of emerging theory and design approaches for the class of problems under review (Cole et al., 2005).

## 4.1 Research methods

In the BI&E stage three principles are important: reciprocal shaping, mutually influential roles, and authentic and concurrent evaluation. In the following section the application of these principles in this research and the methodology that accompanies the principles is discussed. This stage applies a number of methods that are summarized in Table 4-1.

Method	Goal
Participation	Record the effects of the artifact under development and intervene with the artifact in the organization at the same time
Observational research: Unstructured interviews in natural setting	Get a better understanding of specific issues in organizational problem domain

**Table 4-1: Applied methods in the building, intervention, and evaluation stage.**

### Principle 3: Reciprocal shaping

Stretched even further as in design science research, when developing an IT artifact using the action design research approach, the IT artifact and the organizational context are mutually influencing. This implies that the environment in which the artifact is developed influences the artifact under development, and vice versa; the environment is being influenced by the development of the artifact. The action design research team, consisting of the researcher(s) and practitioner(s), engage in recursive cycles to create and improve the IT artifact under development. As Sein et al. state; the development team “*may use its chosen design constructs to shape its interpretation of the organizational environment, use this increasing understanding of the organizational environment to influence the selection of design constructs, and/or interleave the two*” (Sein et al., 2010).

In the first BI&E iteration the IT-artifact is only embedded in a narrow part of the organization, therefore the end-users are not involved in the current iteration. The aim of this research is to

develop a method chunk that fits the PSO context specifically. When developing the method chunk the ADR intention is to simultaneously use the chunk to test its applicability. Therefore, the chunk under-construction is used to intervene in the development process and the results from the development process are used in short feedback loops to improve the chunk.

#### **Principle 4: Mutually influential roles**

The research team interacts with the practitioners during the development of the method chunk for ISD to improve the chunk under-development. We, as researchers, bring our knowledge of theory and technological advances. The practitioners are expected to bring practical knowledge of the professional organizational domain. The aim of this approach is synergy while bringing together both the sources of knowledge.

To gain initial understanding about the problem domain the practitioners' insights are derived by observation using unstructured surveys in a natural setting, thereby getting a thorough understanding of the specific problems in this case setting.

#### **Principle 5: Authentic and concurrent evaluation**

In action design research, evaluation is not a separate phase, which can be initialized after the building phase. Design and evaluation are interwoven in this research. Evaluation is therefore conducted at the same time as the building of the IT artifact, and the intervention in the organizational domain.

Proper methods or defined metrics combined with appropriate data are required to evaluate the extent to which the design activity has been successful,. An array of various measures can be applied to evaluate the design success, such as completeness, functionality, or performance. The methods and measures selected for analyzing the extent to which the artifact under design satisfies the initial problem statement are thus as well applied in the iterative design process. During the design phase the measures and methods can provide

feedback to the artifact under design, as well as to the design process.

Hevner et al. provide an extensive overview of design evaluation methods available for design research (Hevner et al., 2004). In this research participation and behavioral observation are applied. The next sections explain these research methods.

#### ***4.1.1 Method: Participation***

In participant observation the researcher actively participates in the ADR team to develop the IT artifact under review. Participant observation makes a dual demand on the observer. Recording can interfere with participation, and participation can interfere with observation. Participation research is used less in research because of this concern (Blumberg et al., 2005), but in ADR, it is for this reason one of the best suited methods available (Sein et al., 2010). In paragraph 4.4 the researcher participation is described.

#### ***4.1.2 Method: Behavioural observation: unstructured interviews***

Much of what we know comes from observation. Observation qualifies as scientific inquiry when it is conducted specifically to answer a research question, is systematically planned and executed, uses proper controls, and provides a reliable and valid account of what happened (Blumberg et al., 2005). In this research the main participants are interviewed to gain better initial understanding of the processes, information systems and organizational settings. Because, in this stage, in a narrow organizational setting the first concern is to get a rough understanding of the situation as it is when before the research starts. Therefore the participants are interviewed in unstructured interviews in a natural setting.



## 4.2 Factsheet

Figure 4-1 presents a snapshot of the process-data model as is operationalized in this chapter. As in the previous stage, the problem formulation stage, presented in chapter 3, the input that is used and output that is delivered in this chapter is depicted. Below the figure the inputs and outputs are discussed.

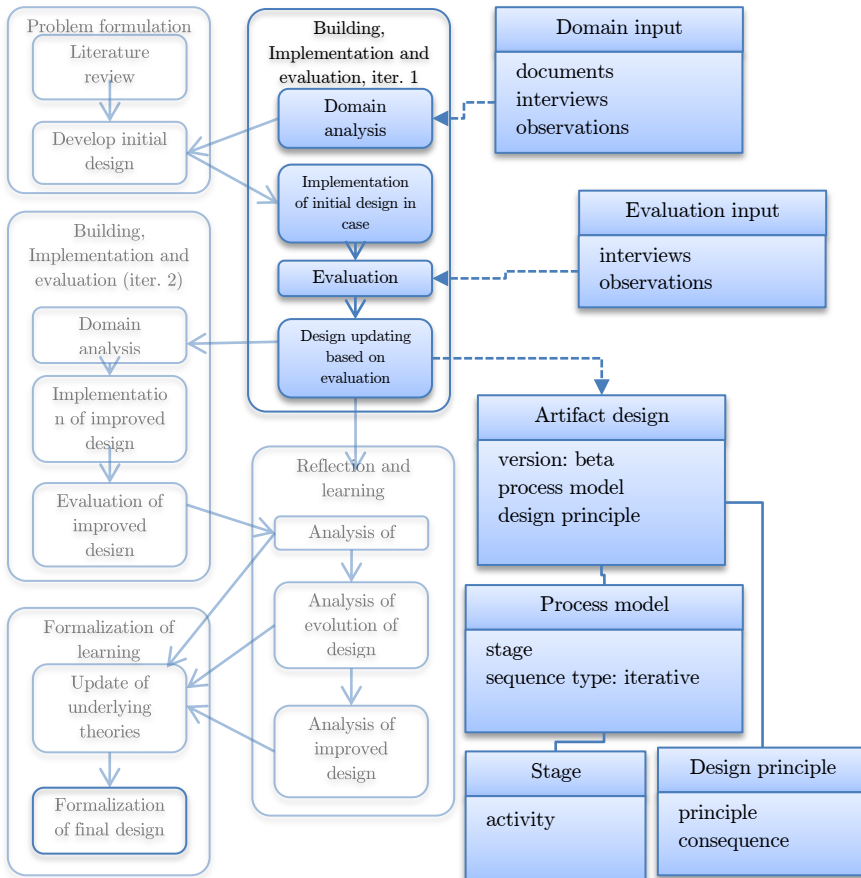


Figure 4-1: Process-data diagram: Snapshot first BI&E stage.

Domain input:

Type	Description
documents	Every available document in regards to the IS development project in both the radiology department and the neurosurgery department is analyzed. Process descriptions are analyzed and updated (if necessary), ISD documentation from previous projects is read, information in regards to patient numbers, waiting lists, strategy, manuals, previous conducted questionnaires, requirements analysis, and requests for proposal for software vendors is analyzed.
observations	Observations are done during collaboration and project meetings with the ISD project team (weekly scheduled over the course of three months), analysis of the primary operations during multiple full-day visits on the outpatient clinics and radiology department.
interviews	Conclusions from document analysis and observations, summarized in a report, are checked during unstructured interviews: <ul style="list-style-type: none"> <li>• Three interviews with ISD project team members</li> <li>• Two interviews with medical staff</li> </ul>

Evaluation input

Type	Description
observations	The outcomes of this BI&E stage are summarized in a report for the ISD project team, and presented during an ISD project meeting. During this meeting observations are done.
interviews	The final report that presents the findings of this BI&E stage is summarizes the developed artifact. This document is used as input for an unstructured interview with ISD project team manager.

Output: Artifact design

Type	Description
version	The initial design of the artifact is upgraded to a beta-version.
process model	The process model is adopted to findings in this case. Most prominent change in this BI&E stage is the adoption of the process model to an iterative/reciprocal sequence; allowing feedback loops and parallel development of multiple stages.
design principle	The design principles in this stage are not updated.

### 4.3 Building

As discussed in chapter 2 there are two particular cases selected in two distinct PSOs. This first BI&E stage tries to develop a method chunk for the IS and process redesign in a radiology department of a mid-sized hospital (MST) (Iacob et al., 2009).

In this case the ADR team consists of a researcher and practitioners from the hospital. The researcher also participates as a project team member that addresses the issues in this case. The practitioners are employed as full time project leaders in this organization. Their role is the development and implementation of the information system in the radiology department. Since this BI&E stage is the first iteration in the IT-dominant ADR cycle, the end-users are not actively involved in the development process. However the ADR team pictures the end-users to be the medical and administrative staff that is going to work with the newly developed IS. The end-users participate actively in the development process by formulation requirements, providing domain knowledge and participate in the redesign process.

The BI&E stage that is described in this chapter follows the approach to develop method chunks as is described in chapter 3. In this chapter the initial approach for method chunk development is

designed based on a broad literature review on relevant topics. The development approach consists of four parts: (1) the identification of contingency variables; (2) the definition of the method chunk's scope; (3) the selection and assembly of method fragments; and (4) the implementation. Stages 1, 2, and 3 are discussed in this paragraph, stage 4 is discussed in paragraph 4.4.

#### **4.3.1 Case description**

For this BI&E stage, a stereotype PSO process in the hospital is selected; the treatment process of a spinal disk herniation. This kind of treatment processes are often termed value shops, or job shops (Pham and Klinkert, 2008; Stabell and Fjeldstad, 1998). See Figure 3-12, on page 77 for a detailed description of value shops. In the case process, several specialists work together, each with his/her own value shop activities, in order to fulfill the complete patient treatment. Coordinating and collaborating in this process is a major challenge, considering the different measures for optimal performance (e.g. patient satisfaction, costs, treatment quality, and through put time).

The first application of the IT-artifact is a specific clinical trial in a mid-size 'top-clinical' hospital located in the east of the Netherlands. This hospital supplies approximately 820 beds for the region, divided over five locations. The patient treatment process is observed via interviews, participation, and group discussions, over a period of 3 months.

In this case study we focus on an information system that is supporting various processes from multiple departments, one of which is the Hernia nuclei pulposi (HNP) or *spinal disk herniation* diagnosis and treatment. This process is based on a standardized clinical trial. A clinical trial is an instrument used to organize the multi-disciplinary care process for the patient, in order to manage and improve the quality of the treatment process.

Of all patients diagnosed with HNP, only 5 to 6 percent receive medical treatment. Yearly 600 patients are treated in this hospital.

The average treatment time (or throughput time) varies between four to twelve months before full recovery, depending on the medical background, and the physical state of the patient. The actual time spend by the patient within the hospital is limited to approximately three fulltime days. As already stated, the HNP treatment process in this hospital department is standardized according to a uniform clinical trial. This standard trial is multi-disciplinary care plan including all key interventions (= interventions that are applied to 80% of all patients) and targets. In the HNP treatment, disciplines involved are, next to general practitioners, physiotherapists, anesthetists, neurosurgeons, radiologist, lab-assistants, administrative staff, and nursery staff. Each discipline can be considered a value shop in itself.

The care plan is set up based on a time schedule in which per day all the necessary activities are listed for the complete clinical treatment. The care in a clinical trial is addressed concerning goals and actions, instead of problems. Thus, the goal of a clinical trial is to operate goal oriented, instead of problem oriented. Although with the introduction of the care plan, this process appears to be not typical anymore for a PSO organization; processes are structured, and actions are conducted according to standardized decision models. However, in practice the outcome of the process is still highly stochastic; the inflow in the process is heterogeneous, which makes forecasting the outflow very complex. There are many professionals involved in each process step, all following the value shop process flow: (problem finding and acquisition, problem solving, choice, execution, and control / evaluation).

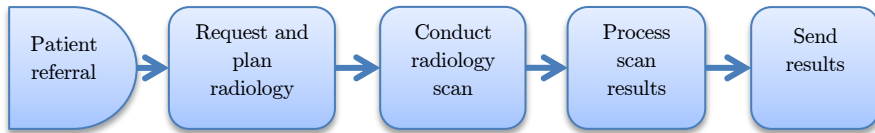
For this case we zoom in on the IS that facilitates two intertwined processes, operated by two specialized departments in the HNP treatment; the neurosurgery department, and radiology department. Both these departments can be considered value shops, with a typical value shop process. Since the development of the IS for this particular part of the process turns out to be complex, no standardized ISDM supports this process.

In order to increase the efficiency of the treatment process a new IS is about to be selected and implemented. Because processes and departments in hospitals are often interwoven, the IS of the radiology department is a main determinant of the efficiency of the treatment process.

The main department involved with this HNP procedure is the neurosurgery department. This department is responsible for initiating the clinical trial and the administrative settlement. In this case study we focus on the operations of both the neurosurgery department and the radiology department. The issue in this case study is the automation and improvement concerning the ordering, requesting, and planning of the medical diagnostics as provided here by the radiology department, for a treating specialism (here the neurosurgery department).

The current flow of the basic process is initiated by either a general practitioner, or a specialist in a hospital, diagnosing a patient HNP. After this diagnosis, a complex treating process is started to heal the patient from his injury. At a certain point in the process the neurologist forwards the patient to the radiologist for a scan of the internals of the patient. This operation can comprise several scans, ranging from a simple “bucky” scan (Rontgen scan) to the more complex CT scans or MRI scans. A high level view of the radiology process is depicted in Figure 4-2, a more detail provided at the Appendix A: Process description medical diagnostics, MST case. Here the patient is send to the radiology secretary to make an appointment for the advised scan. The response for this request for a scan can be given directly (as a scheduled appointment), or with an acknowledgment for the request (upon which the scheduled appointment is communicated later with the patient). On the day of the scan, the patient registers at the central radiology desk, and is scanned after a short period of time. The radiologist checks the quality of the scan, and the scan is being processed into the system. After the radiologist concludes with a final diagnosis, the results are send to the requesting department (i.e. the neurosurgery department). Now the neurosurgeon can finalize his diagnosis, and

decide to perform medical surgery. After the surgery, the patient is put up for post-surgery treatment, and outpatient recovery. At this point the patient should be fully recovered.



**Figure 4-2: Radiology process (high level view).**

For the usage and implementation of the IT artifact, the focus is on the development of the IS for the radiology department. Because of the poor current IS-support several problems are encountered in this current process flow:

- Sub-optimal usage of the radiology resources
- Long waiting lists for radiology scans
- Difficulties for requesting and scheduling patient treatment
- Unknown costs per patient treatment
- No transparency and flexibility in radiology planning process

Both the neurosurgery process as well as the radiology process that is supported by the IS are described in more detail in Appendix A, on page 245.

#### ***4.3.2 Contingency variables: case analysis***

The dominant approach that aims to enhance the attainable level of efficiency in health care that is currently taken is a traditional one consisting of a central directed planning system that is used to schedule resources, staff, and patient treatments. The organization in this case is no exception. However both literature, as discussed in paragraph 3.5, and participation in the IS development project in this organization demonstrate two major issues that prevent this approach from being successful:

The first one concerns the *internal process characteristics*. The stochastic, complex, dynamic and heterogenic process characteristics impose limitations on the efficiency of a health care organization. The only known measures for preventing extreme low efficiency is to apply batching / lot sizing, buffering, and sequencing techniques to the process demand. In this case, these techniques are also encountered.

The second problem is the *complexity of inter- and intra-organizational communication and coordination*. In this case there are many conflicting goals for the neurosurgery and the radiology department. These conflicting goals concern the specific sequence in which patients are treated, the authority over the departmental planning, and the scheduling of feedback sessions.

### **ISD perspective**

In chapter 3 ISD is described from the body of knowledge perspective. In this section the insights from literature are applied to gain better understanding of the ISD process in this case.

In this ISD project the aim of the information system under-development is to deal with inefficiency and lack of transparency in the radiology department. With the development of the new IS the hospital aims to improve throughput time, not only within the radiology department, but also in departments that share a part of their process with the radiology department. In this study we zoom in on the neurosurgery department. When this perspective is mapped on the Hirschheim's four paradigms it becomes evident that in this case the *functionalistic* paradigm is dominant. What can be perceived from the case is that this information system is developed to contribute to the efficiency of a group of departments in this hospital.

In chapter 3 the functionalistic paradigm is aligned with the Hard-Structured Analysis approach for ISD. In this top-down approach the development process is structured sequential. Furthermore this approach assumes that problems are well-defined and can be solved



accordingly. This matches exactly what is perceived in this case study. The problem is well-defined (lack of efficiency) and the process is structured in a very traditional sequence (process analysis, process redesign, requirements analysis, vendor selection, and implementation). The ISD methodologies that are identified to match this ISD approach are the traditional methodologies like systems development life cycle overlap with the structured ISD process as designed in this case.

As can be concluded from the gap between literature analysis and case analysis. Literature demonstrates that as the key characteristics of professional organizations induce that the organizational ends are not uniformly defined. Therefore, the ends for information systems are often diverse. In this organization a clear choice is made to improve efficiency of the radiology department. Other aspects as for instance patient treatment quality, the staff's ability to maintain autonomy, safety, and process integration are less important. And although the currently applied ISD methodology matches the functionalistic paradigm, the research team questions the success factor of this approach.

### **PSO perspective**

In chapter 3 we predefined a set of PSO characteristics that are influencing the ISD process. In this section these characteristics are further specified to match this particular case. The four variables from Leavitt's model are used as references in this analysis.

*Information technology.* For a PSO information technology is identified as a means to coordinate in complex structure like processes, workflows, and business rules. From literature it becomes evident that IS that are based on ERP thinking do not work well in complex organizations. In this case however an ERP based system is adapted to work throughout the overall organization. The IS that is developed for the radiology department is to be integrated in the overall ERP system. The ERP system thinking limits the radiology in developing an IS that fits their requirements.

*Tasks:* The ill-defined processes with conflicting goals and an unpredictable process flow that is characterized as typical for PSOs does not apply to the radiology department itself. However when perceived as a node in the outpatient processes, the typical characterizations do apply to this case. Especially the conflicting goals in the several processes cause complex implementation decisions.

*People:* Staff in a PSO is typically self-responsible with professional autonomy. This fully matches the case under review; the medical staff makes their own decisions in each decision point in a process. Quality is characterized as the most important indicator for productivity according to literature. However in this particular case quantity is the dominating indicator for productivity; throughput time, number of patients waiting, and access time (time it takes before a patient is treated) are most important.

*Structure:* The split-responsibility organization structure is not clearly evident in this case. The surgeons in the neurosurgery department are self-employed (in Dutch: maatschap), and are therefore self-responsible for their produced output and income. As is noted in the previous section is quantity an important factor in the radiology department, in the neurosurgery department is the combination quality – quantity the measure that is optimized. Furthermore is the neurosurgery department clearly organized as a value-shop with the iterative loops to identify the best treatment. The radiology department in contrast can be better characterized as a traditional manufacturing organization; keeping throughput as high as possible by standardizing as much as possible.

### **Hiatus in ISD process**

After a series of interviews with the practitioners involved in the research team, observations done in the organization during both project meetings and end-user observations, and analysis of documents available some critical aspects in the ISD process are identified. In the ISD process in this case the main problem is the conflicting goals that complicate the decision making process about

the type of IS to be implemented. Several departments have different stakes in a single process because of the interwoven. Furthermore, the solution direction concerning a new IS is not yet defined. There are various solutions, but many of them have a large impact on the flow of the processes. The last issue that is characterizing this case is the professional autonomy that the specialists from the various departments highly value. The traditional ISD approach taken in this case omits these aspects. Therefore the ADR team decides to develop a chunk that addresses this issue. Table 4-2 depicts the issues that are not covered by the ISD methodology that is currently used to design and implement the IS for the radiology and adjacent department. The requirements that flow from the problems are used as input for the chunk development process.

<b>Problem</b>	<b>Requirements for new method fragments</b>
<b>Conflicting goals in the process that is to be supported by the new to-be developed IS</b>	<ul style="list-style-type: none"> <li>- Ability to deal with multiple very different interests in a single process that is to be supported by an IS</li> <li>- Analyze individual / departmental goals for shared interests</li> </ul>
<b>Solution direction for new IS is not yet defined</b>	<ul style="list-style-type: none"> <li>- Ability to explore solution directions</li> <li>- Define solutions based on shared interests</li> </ul>
<b>Professional autonomy of the staff makes a solution oriented discussion nearly impossible</b>	<ul style="list-style-type: none"> <li>- Ability to evaluate individual requirements for the new IS</li> <li>- Ability to integrate the individual requirements into a shared perspective</li> <li>- Means to discuss outcomes</li> </ul>

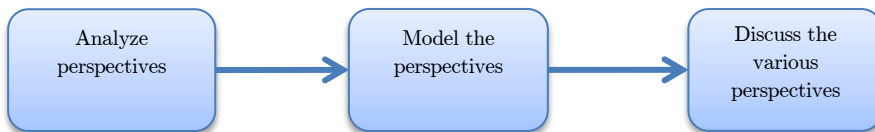
Table 4-2: Requirements based on MST case analysis.

### ***4.3.3 Method chunk's scope***

The second stage in the development of method chunks, is the development of the chunk's process and the definition of its products. The chunk's process and products are to be based on the contingency

variables that are listed in the previous paragraph. This section covers these activities.

*Chunk process:* The ISD process the method chunk under-development is to be part of, is traditionally structured, similar to the systems development life cycle. Therefore is the process of the chunk under-development is structured accordingly. The main hiatus in the ISD process is found to be a setting of conflicting goals, where several departments have different perspectives about the means to attain high efficiency. The chunk that is to be developed should therefore offer the ability to first analyze the current perspectives, model the perspectives, and finally use the model as a discussion platform. The process for the chunk is herewith defined to have at least these three stages, as depicted in Figure 4-3.



**Figure 4-3: Initial design method chunk's process flow.**

*Chunk products:* There are two main products identified that are to be results of the method chunk under development. The first is an intermediate product that provides an overview of the different goals that are held by the involved departments. This overview is planned to be the input for the second stage of the method chunk in which the distinct goals are analyzed in order to develop a shared perspective on the exact direction the overall ISD project should be developed towards. Since this organization is identified to be quantity driven in regard to its effectiveness, the product should be able to include this factor as a means to analyze the current situation and develop the new perspective.

#### ***4.3.4 Selection, configuration and assembly***

The scope for the method chunk under-development is set, based on the analysis of the contingency variables in this case. In this

paragraph fragments are selected and assembled into a method chunk.

### ***Selection***

Two method fragments are to be selected that fulfill the requirements that are listed in the case analysis. In this section the *goal modeling* fragment and the *collaborative design* fragment are selected.

### **Goal modeling**

A method fragment that immediately comes to mind when analyzing the contingency variables and scope in this case, is the *goal modeling* fragment (Anton, 1996; Dardenne et al., 1993a). Goal modeling is applied as the *product-oriented* method fragment in this development process. Because of the process interdependencies, goal analysis can produce more effective systems since goals play an important part in several respects. Van Lamsweerde (van Lamsweerde, 2000) argues why goals are critical in any ISD process:

- Goals provide an exact criterion for sufficient completeness of a requirements specification, the specification is complete with respect to a set of goals if all the goals can be proved to be achieved from the specification and the properties known about the domain considered (van Lamsweerde, 2000; Yu, 2004)
- Explaining requirements to stakeholders is an important issue; goals provide the rationale for requirements. A requirement appears because of some underlying goal which provides a base for it. A goal refinement tree provides traceability links from high-level strategic objectives to low-level technical requirements (Lapouchnian, 2005; Mostow, 1985; van Lamsweerde, 2000)
- Goal refinement provides a natural mechanism for structuring complex requirements documents for increased readability
- Goal refinements provide the right level of abstraction at which decision makers can be involved for validating choices being made or suggesting other alternatives overlooked so far

- Goals have been recognized to provide the roots for detecting conflicts among requirements and for resolving them eventually
- Goals drive the identification of requirements to support them, together with scenarios they are the driving forces for a systematic requirements elaboration process
- A goal refinement tree provides traceability links from high-level strategic objectives to low-level technical requirements
- Goal models provide a way to communicate requirements to customers. Goal refinements offer the right level of abstraction to involve decision makers for validating choices being made among alternatives and for suggesting other alternatives

There are good reasons for modeling, and modeling goals in particular in organizations. As Kueng states the obvious reason: “Managers, systems analysts and their ilk, all those who use business models of one sort or another, are concerned with managing complexity” (Kueng and Kawalek, 1997). Curtis elaborates some more on the basic uses of organizational modeling. In his opinion, models of organizations are used to facilitate the human understanding and communication, to support the process improvement and management, and to automate the process guidance and execution support (Curtis et al., 1992). But, as Holland, Lansing and many others already noted, organizations, processes, and structure are prone to continuously evolve (Holland, 1996; Lansing, 2003). In contrast to the processes and structure within organizations, goals in organizations are relatively stable (Anton, 1996). Thus by emphasizing organizational structure and processes by modeling these – as opposed to behavior and goals – the flexibility and creativity within an organization will be limited to the boundaries of the produced models. In this report we use goals as a basis for producing models of operations which allow flexibility in an organization.

### **Collaborative design**

The *process-oriented* method fragment selected for this development process is collaborative design.

The collaborative design is described as the collective activities of diagnosis, problem analysis and initiation of creative ideas to realize a different organization of work. Participation is argued to be essential, but is often realized solely by involving (external) expert designers. Traditionally the design process is sequentially organized using a step-by-step approach, using expert knowledge to design and guide tools and interventions (Katsma, 2008). Sociotechnical systems theorists react to the traditional approach to organizational design and propose a more interpretative deployment (Berggren, 1993; Van der Zwaan and De Vries, 2000). In this latter approach organizational participants are involved from different backgrounds. This approach assumes that both the design content and process are difficult to predict in advance. Therefore the design content is defined and detailed by sharing different perspectives and social interactions between the expert and organizational designers. This represents a bottom-up formulation of the design content.

Participation, during collaborative design, fundamentally contributes to knowledge exchange and competence development between the actors, because it creates the opportunity to really assess the organizational problem by the involved actors themselves and act in the design process (Ketchum and Trist, 1992). This is often contrary to reality in which organizational members perform according to predefined role patterns or are even only able to approve of another person's solution without being heard. Collaborative design offers the involved participant an opportunity to play with and experience different concepts and try out *design alternatives* that represent future organizational formats.

The integration of learning and playing with gaming and simulation shows an enhanced experience for participants, improved social interactions during the process and it also boosts the quality of the design outcomes from these sessions (Barreteau and Abrami, 2007; Feinstein et al., 2002). In collaborative design the participants not only design, but also investigate the mutual reactions to these produced designs. Literature in simulation-based gaming -and design show how these methods are able to increase the sense of reality

(Lindley, 2005), stimulate knowledge exchange and even after the implementation process these simulations support the organizational members since they give additional information and functionality compared to the business process documentation (Wenzler and Chartier, 1999).

### *Configuration*

At this point in the process the method fragments are selected and need to be configured in order to be effectively usable in the ISD process. Literature on the two method fragment, goal modeling and collaborative design, that are selected offer a broad array of possible configurations for the actual configuration. In this paragraph design options are discussed and the fragments are configured. Since this BI&E stage does not include end-user involvement, predominantly the configuration of the goal modeling fragment is discussed in this section. Concerning goal modeling, in this paragraph five concepts are discussed that need to be configured to use the method fragment in the ISD process. These are goals, stakeholder analysis, goal elicitation, goal modeling, and goal analysis. Figure 4-4 and Figure 4-5 depict a general configuration of the method fragments that are discussed in this paragraph. Goal elicitation, modeling and analysis are often critical and complex processes. Business people need therefore ways to express these goals as clearly and precisely as possible, both for their own understanding and for communication with other stakeholders, such as requirements engineers, process designers, business strategists and software developers. To date, there is no de facto standard notation/language for describing goals, and they are often described in myriad of ways going from informal and highly abstract pictures or text that lack a well-defined meaning to very precise predicate logic formulae. This leads to misunderstandings, and makes it very difficult to provide tools for visualization and analysis of goals. It also makes for organization impossible to trace back to business processes the causes of success or failure in reaching organizational goals. In this paragraph's last section the use of the goal modeling fragment in design sessions is discussed.



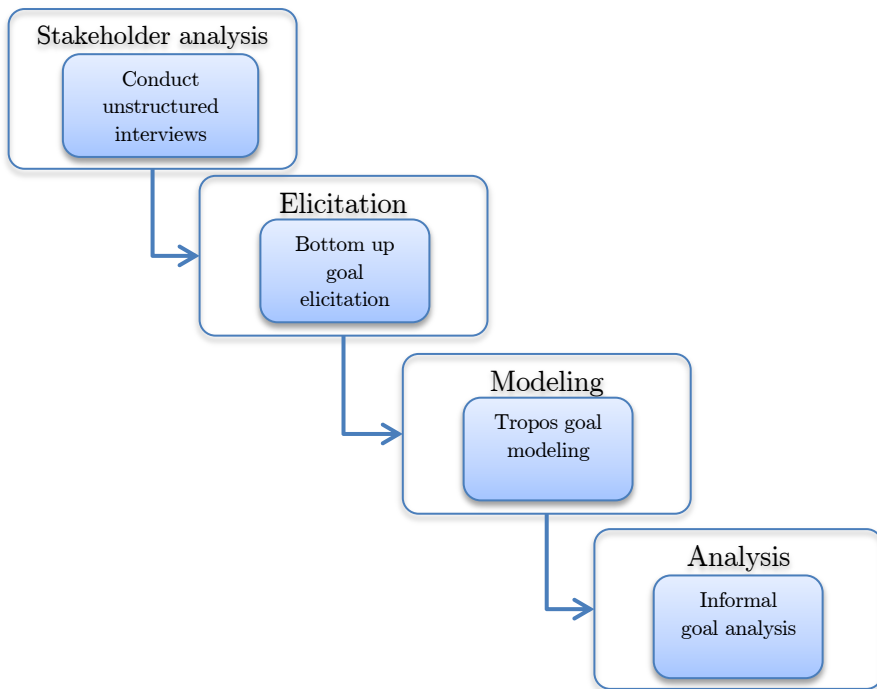


Figure 4-4: Goal modeling method fragment - general configuration.



Figure 4-5: Collaborative design method fragment - general configuration.

### Goal modeling

*Goals*: can be formulated at different levels of abstraction ranging from high-level strategic concerns to low-level technical concerns. Goals also cover different types of concerns, functional and quality concerns. A basic tree structure is usually not sufficient to display goals. First, a distinction between soft goals and hard goals needs to be made. Soft-goals are goals that cannot be easily satisfied. Abstract, high-level goals are usually represented as soft-goals. Hard-

goals are goals that are directly measurable and usually equate to concrete requirements that have a KPI attached. Hard-goals usually achieve soft-goals. Another extension is to include the actor in the goal modeling framework as representing the entity (e.g., person, organization or IS) who achieves goals. In information systems development, business process modeling has become a major focus of attention. The focal point in process modeling are the structures and operations (the *what*, *who*, *where*, and *how*) in the organization. The goal (the *why*) of the organization, the motivation to carry out a process or operation, is implicitly modeled within a business process. Although goals are included in nearly all descriptions of business process modeling methods (e.g. “a process is a set of partially ordered steps intended to reach a goal” (Feiler and Humphrey, 1992)), the notion of the *goal* itself has received relatively little attention in the literature on modeling (Soffer and Wand, 2004). Goals in organizations are considered to be abstract concepts, like the high level objectives of the business, organization, or system (Anton, 1996). Since human action primarily is driven by goals (Scherer and Zölch, 1995), the concept of goal is a key issue in most process modeling methods. However, there are two problems with goals in process models: (1) most of the time goals are not explicitly modeled, and (2) there is a considerable controversy about the concrete goal taxonomy or classification. This expresses in the various definitions about goals and goal modeling:

- A goal is a non-operational objective to be achieved by the composite system (Dardenne et al., 1993b)
- A goal is a dependency between two enterprise stakeholders, one stakeholder depending on another to fulfill an agreement (Giorgini et al., 2008)
- A goal is a strategic interest of an actor (Bresciani et al., 2004)
- Goal is a state which the system desires to achieve. It explains why the system is built (Yang and Zhang, 2003)
- Goals are the process actor's viewpoint on the process (Antón et al., 1994)

- Goals represent the strategic interests of actors (Giunchiglia et al.)

The definition that *a goal is the actor's viewpoint on a process* is adopted for this research. This definition of goals still does not imply a clear concept. It can be interpreted as an abstract high level concept such as *improvement of efficiency*, to low level concrete concepts as *register client*. Clearly the idea of a goal implies a broad variety of concepts. In the literature on goals, goal-based requirements engineering, and goal oriented analysis, many distinctions or classifications are made. Also the different approach on the identification of goals directs to diverse viewpoints on goal modeling. Several authors have written about different classifications, and apparently two axes are important in this analysis. The first axis is the goal classification or taxonomy. Classifications here are made on basis of functional or non-functional goals (Keller et al., 1990), hard goals or soft goals (Bresciani et al., 2004; Soffer and Wand, 2005), temporal goals (Dardenne et al., 1993b), and inter or intra actor goals. The second axis focuses on the direction in the organization (top-down versus bottom-up) (Anton, 1996).

Functional / non-functional goals: The first classification to be discussed here is the distinction between functional goals, and non-functional goals. This distinction is proposed both by Keller et al., and by Kueng et al. (Keller et al., 1990; Kueng and Kawalek, 1997). Functional goals are services a system delivers such as the registration of a patient, the sending of an invoice, or the assignment of a resource to a specific operation in a process. In most cases, functional goals can be assigned to a specific process, actor, or organizational structure. Non-functional goals do not apply to a specific process, actor, or structure. This type of goals is of importance for the whole organization. Examples are security, performance, or scalability. Both the functional and the non-functional goals can in principle be clearly defined, and thus are measurable.

Hard and soft goals: Another goal classification is the distinction between hard goals and soft goals (Mylopoulos et al., 1999), or between operational goals and strategic goals (Soffer and Wand, 2005). An operational goal can be accomplished by a given process. The strategic goal, however, cannot be achieved by a single process. As Soffer et al. state, soft-goals only refer to a part of the domain, not directly to a given process. Soft-goals are used to specify (at a qualitative level) not sharply-cut objectives. Hard-goals clearly define a state/target, an actor desires to achieve. Some author equate this classification to the functional versus non-functional distinction (see (Bresciani et al., 2004; Dardenne et al., 1993b)). However we assume in this taxonomy that hard goals are goals which can be determine by a measure, whereas soft goals have no clear-cut criteria as to whether they are satisfied. In the functional and non-functional classification, both types of goals are measurable. An example of a hard goal is the increase of a margin on sales, and of a soft goal is the increase in the loyalty of the customers. In the hard and soft goal classification provided by Mylopoulos et al. the goals are always modeled as interdependencies between two actors or systems in an organization (Mylopoulos et al., 1999). Because soft goals only relate to a domain, not a specific process, the analysis of soft goals cannot be addressed in the same manner as the analysis of hard goals.

Temporal goals: A third classification is in terms of the temporal events influencing the satisfaction of goals. Dardenne et al. define goals as non-operational objectives to be achieved by the composite system. Here, non-operational implies that the goal cannot be established *through appropriate state transitions under control of one of the agents* (Dardenne et al., 1993b). An example here would be the goal of a library to have each request for a book eventually satisfied. This goal cannot be achieved by the action of one agent, actor, or system itself within the

organization, thus this goal is defined as non-operational. These goals can be classified according to their temporal pattern. Dardenne et al. identified five patterns with which goals can be classified. These patterns are *achieve*, *cease*, *maintain*, *avoid*, and *optimize*. These patterns have an impact on the set of possible behaviors of the system. Where *achieve* and *cease* patterns generate behavior, restrict *maintain* and *avoid* behavior. *Optimize* pattern goals compare behavior.

Inter/intra actor goals: The fourth important classification is on the level of goal assignment. Goals can be assigned to a system or organization, to an actor in the system or organization, or to the relationship between two or more actors in the system or organization. Dardenne et al. define this distinction as system goals and private goals (Dardenne et al., 1993b), Mylopoulos et al. define goals between two or more actors in the system, and Keller et al. most often identify goals on the system level (Keller et al., 1990).

Because it is of high importance to have a clear understanding of the goals and the goal requirements before using these requirements for modeling and analysis, emphasis is needed on the goal acquisition phase. Most important in this phase is the knowledge of the problem domain, because clients are not capable to formulate the relevant requirements explicitly and precisely (Dardenne et al., 1993b). We argue that the problem domain knowledge can be acquired by means of, e.g., stakeholder analysis, document mining and scenarios, and has to be combined with a certain strategy for goal identification and elicitation. These two aspects are further addressed in the following two sections.

*Stakeholder analysis:* Aims to develop and understand stakeholders from the perspective of an organization, or to determine their relevance to a project or policy (Brugha and Varvasovszky, 2000). Conducting a stakeholder analysis involves questions about the positions, interests, influences, interrelations, networks and other characteristics of the stakeholder, with reference to their past,

present, and future positions (Freeman, 1984). Stakeholder analysis encompasses a range of different methodologies for analyzing stakeholder interests and is not a single tool. The purpose, time scale of interest, the context, and the degree to which an issue has been clearly defined, bear on how a stakeholder analysis is carried out (Crosby, 1992). Varvasovszky et al. developed a framework for conducting a stakeholder analysis (Varvasovszky and Brugha, 2000). In this framework first three preliminary questions are asked. These questions concern, in accordance to Crosby:

- The aim and the time dimension of the analysis. Being clear about the aim helps to identify the scope and time dimensions of most interests; past, present, or future. Depending on the aim of the analysis and the resources available, the analysis may be conducted over a various time period.
- The context of the stakeholder analysis. Understanding the culture and the context is necessary for deciding how to interact with the stakeholders, collect and analyze the data.
- The level at which the analysis will take place. The analysis can take place at one or more levels; local, regional, national and international. This influences the means of collecting data, and the target persons to be considered a stakeholder.

After these preliminary questions, the stakeholders can be identified. Stakeholders can be defined as actors who have an interest in the issue under consideration, who are affected by the issue, or who – because of their position – have or could have an active or passive influence on the decision making and implementation processes (Varvasovszky and Brugha, 2000). Multiple *methods for data collection* can be applied in conducting a stakeholder analysis; face-to-face interviews using checklists, semi-structured interviews, and structured questionnaires can all be used for collecting data from primary sources. Usually these all are individual respondents. Though groups of stakeholders may also be interviewed – e.g. by means of focus groups, or informal group discussions (Varvasovszky and Brugha, 2000). Other data sources may include published and

unpublished documents, reports, policy statements, or internal regulations of the organization. Stakeholder analysis has been used as a management and strategic tool for identifying the optimal strategies for managing other stakeholders, identifying current and future opportunities and threats and how best to handle them (Blair et al., 1990). Because stakeholders are used to express their requirements in processes or actions, rather than in goals, it is more effective to analyze the interviews and documents for action words. Each stakeholder has different and sometimes conflicting requirements, goals, and priorities. (Antón et al., 1994). The goals for a desired system are sometimes unclear from an outset. Thus they must be extracted from diverse sources of information. As Anton states, is it unreasonable to expect a complete set of goals for a system from only one information source, but the combination of goals extracted from various information sources does produce a more complete set of goals.

*Goal elicitation:* Goal identification is not a trivial endeavor. Goals may be explicitly stated by the stakeholders but it may also be necessary to extract them from various sources of information available to requirements engineers. Therefore, when goals are implicit an elicitation process must take place. A preliminary analysis of the current system/organization and a *stakeholder analysis* are important sources of goals. This analysis can result in a list of problems and deficiencies that can be formulated. Goals can also be elicited from available documents, interview transcripts, process descriptions, etc., by searching for intentional keywords in the documents (van Lamsweerde, 2000). Anton (Anton, 1996) mentions that using process descriptions is very useful to identify an initial set of goals, although it is not wise to only use process descriptions. Other sources of information, such as *interviews*, diagrams should be used to supplement the goal identification process. Stakeholders express their requirements in terms of operations or actions, rather than goals. It makes sense to look for action words such as schedule or reserve when gathering requirements for a meeting scheduler system. In an organization goals can be specified in a *bottom-up* or *top-down* fashion. Top-down specification of goals can be regarded as

refining and decomposing the system's goals. The top-down approach usually starts from analyzing the corporate strategy and translating it into a set of soft goals, which are gradually refined and decomposed into operational level goals (Soffer and Wand, 2005). Bottom-up analysis of goals starts with the analysis of the individual autonomous actors/agents, identifies their "private" operational goals and aggregates them into more abstract and higher level goals that concern the whole organization/system. The former approach appears to require guess work and inventiveness, since there is no systematic way for refining high level goals into concrete quantifiable goals. Furthermore, high level enterprise goals do not always imply what should be the goal, for example, of a concrete low level process (Anton, 1996). In the latter approach, the challenge resides in the fact that goals do not always comply with one another. Each stakeholder has different requirements and priorities. Very often these interests are conflicting. While the first type of approach is typical / suitable for organization models in which the control is imposed into a centralized and hierarchical fashion, the second approach is more suitable for organizational models in which the control is distributed over several units/departments. Nevertheless, common to both approaches is the fact that in the end both approaches result into a detailed decomposition of goals in the form of a goal tree in which leafs' granularity is sufficiently fine to allow their operationalization. The same top-down/bottom-up distinction holds for software system development technologies/paradigms. More precisely, we see the top-down approach as more appropriate for workflow-driven development approaches (e.g., SOA) and the bottom-up approach for distributed approaches (e.g., multi-agent systems). Both strategies for goal elicitation - top-down and bottom-up -are incorporated in Goal-Oriented Requirements Engineering (GORE) methodologies. The top-down approach for example is implemented in the KAOS methodology (Dardenne et al., 1993b), whereas the bottom-up approach is implemented in the TROPOS method (Mylopoulos et al., 1999). GORE methodologies are explained in the next section. To both approaches there are some advantages and disadvantages attached, see Table 4-3.



	Top-down	Bottom-up
Advantages	Uniform set of goals, applied to the whole organization Limited effort needed for applying method and analyzing results	Easy to apply starting with low level operational processes and actors All goals from different stakeholders can be included with relative little effort
Disadvantages	Difficulties cascading high level abstract goals to lower operational level Omitting individual or subgroup goals on lower organizational levels.	Analyzing and applying results of this method can lead to sub optimization Approach might end up in many conflicting goals amongst actors High work load involving the analysis of many actors and many processes

**Table 4-3: Top-down and bottom-up goal modeling approaches compared.**

Three types of relations essentially form the basis of the top-down and bottom-up approaches: decomposition, abstraction and operationalization. These relationships are essential mechanisms through which during goal specification one can go, on one hand, from one level of abstraction to the following one (decomposition and abstraction), and, on the other hand, from goals specification to design specifications (operationalization).

*Goal modeling*: has received a large amount of attention over the past years and its popularity has increased ever since. The main reason for this is the inadequacy and inability of traditional systemic approaches (e.g., structured or object analysis), to capture the rationale for the systems. At the requirements level these approaches treat requirements as consisting only of data and behavior / processes and disregard the high-level concerns in the problem domain (van Lamsweerde, 2005). According to van Lamsweerde Goal modeling ends where most traditional specification techniques would start, it focuses on the activities that precede the formulation of

system requirements (van Lamsweerde and Letier, 2004). Some of the most popular goal modeling tools are KAOS (Dardenne et al., 1993b) and I\*/ Tropos (Giunchiglia et al., 2002; Yu, 1995; Yu and Mylopoulos, 1994a).

KAOS is a goal modeling approach, with a large amount of formal analysis techniques; it stands for Knowledge Acquisition in automated Specification. It is a multi-paradigm framework that allows combining different levels of reasoning. KAOS is focused on goal satisfaction and the systematic building of complete, conflict-free goal based requirements modeling (van Lamsweerde, 2005). KAOS is semi-formal for structuring and modeling goals, qualitative for selection among the alternatives and formal for accurate reasoning. In KAOS a goal is a prescriptive statement of intent about some system whose satisfaction in general requires the cooperation of some of the agents forming that system (Dardenne et al., 1993b). Goals in KAOS can be both functional and non-functional (soft goals). Goal refinement ends when every sub-goal can be realized by some agent. These end-goals (terminal goals) become the requirements of the system. For more information on KAOS we refer to the following sources: (Dardenne et al., 1993a; Dardenne et al., 1991; Dardenne et al., 1993b; van Lamsweerde, 2000, 2004, 2005; van Lamsweerde and Letier, 2004; van Lamsweerde and Willemet, 1998).

I\* (Yu and Mylopoulos, 1994a) is a technique that focuses on modeling and reasoning support for early phase requirements engineering. It tries to capture the understanding of the organizational context and rationales that lead up to systems requirements. It consists of two main modeling components. The Strategic Dependency (SD) model is used to describe the dependency relationships among various actors in an organizational context. The Strategic Rationale (SR) model is used to describe stakeholder interests and concerns, and how they might be addressed by various configurations of systems

and environments (Yu, 1995; Yu and Mylopoulos, 1994a; Yu and Mylopoulos, 1994b).  $I^*$  can be used for both early and late phases of requirements engineering. During the early requirements phase  $I^*$  is used to model the environment of the system to be, it facilitates the analysis of the domain by allowing the modeler to diagrammatically represent the stakeholders of the system, their objectives and their relationships (Mylopoulos et al., 1999). During the late phases the  $I^*$ -models are used to propose the new system and the new processes and evaluate them on how well they meet the functional and non-functional needs of the users. Tropos is an extension framework for  $I^*$ , it is a requirements driven agent oriented development methodology (Bresciani et al., 2004; Mylopoulos et al., 1999). The aim of Tropos software methodology is to guide the development of agent-based software systems (Mylopoulos et al., 1999). Tropos incorporates knowledge level concepts (such as agents, goals, plans) through the phases of software development. A pivotal role is assigned to requirements analysis when the environment and the system-to-be are analyzed.

*Goal analysis.* Van Lamsweerde mentions that although goal based reasoning is valuable in IS development, goals are hard to understand for some stakeholders (van Lamsweerde, 2000). Therefore *scenario-based elicitation* is a good alternative for stakeholders to discuss the system. Therefore some techniques have been developed to elicit goals using scenarios. Potts also suggests that it is unwise to apply goal based requirements methods in isolation and says that they should be complemented with scenarios (Potts, 1997; Rolland et al., 1998). There are two approaches to goal analysis: (1) formalizing the constraints into predicate logic in order to facilitate reasoning; and (2) discuss the results of the modeling exercise with focused feedback groups.

Formal reasoning has been designed to work at the architectural and design level. These are the later phases of software development. As a result the application of formal

methods to early requirements analysis is by no means trivial; there is a mismatch between the concepts used for early requirements specification (e.g. actors, goals) and the constructs of formal specification languages such as Z (Fuxman et al., 2004). Fuxman et al. developed a specification language for formal analysis to early requirements specification. This language is based on the primitive concepts of I\* such as actors and goals (Yu, 1995) and enriched with a temporal specification language based on KAOS (Dardenne et al., 1993b).

Feedback sessions can be used to discuss the results of the modeling exercise. This approach provides, in contrast to the formal approach, no formal analysis of the requirements. However, the goal models used during feedback sessions provide excellent ground to discuss and refine the results of the individual opinions, expressed during the (non-) structured interviews. Used like this, the goal modeling technique is a means to overcome dead-lock situations when goals really conflict. We found this approach more valuable in the current case where the goal modeling method fragment is applied. Therefore the ADR team used this variant of goal analysis in the collaborative design sessions.

### **Goal modeling summary**

Goal modeling is a means to identify, during the IS development process, the rationale for the development of the IS. In goal modeling the various interests of every stakeholder of the IS under-development is mapped to identify the best IS alignment with the organization. Various types of goals can be modeled: functional and non-functional goals, hard and soft goals, temporal goals, and inter/intra actor goals. The goal modeling process, as identified for the development of the method chunk in this chapter consists of four stages: stakeholder analysis, in which the stakeholders are identified; goal elicitation, in which the goals per stakeholder are identified, goal modeling, in which the elected goals from every stakeholder are grouped in one goal model; and goal analysis, in which the developed

goal model is analyzed in order to be used as a input for decision making.

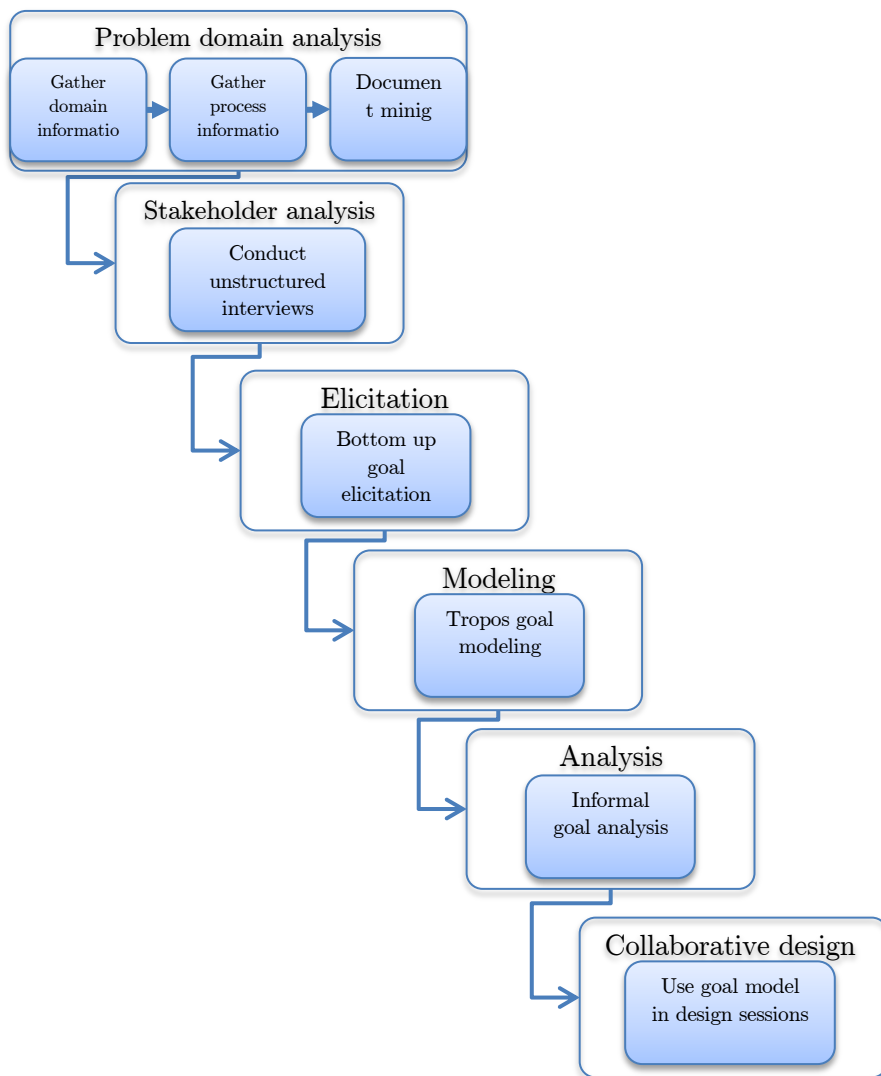


Figure 4-6: Goal modeling - collaborative design process flow.

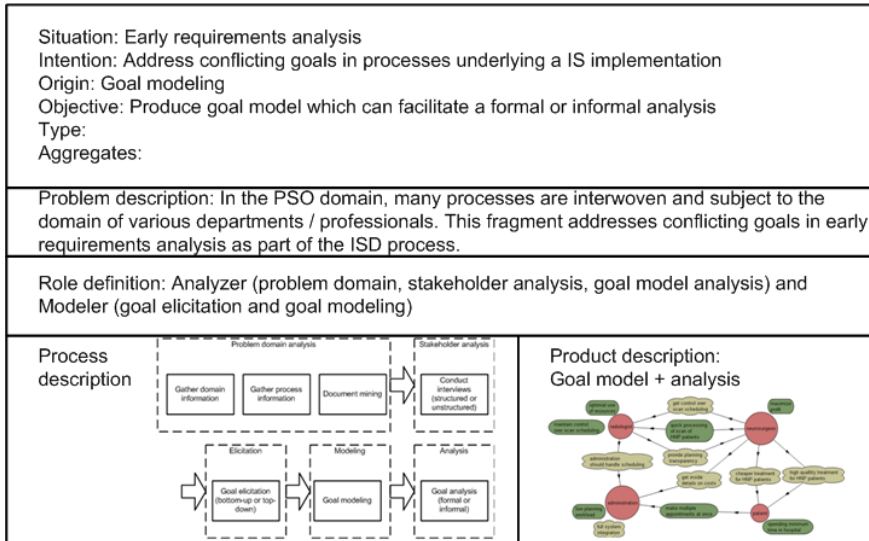
*Assembly*

In the previous section the configuration of the method fragments is discussed. In this paragraph both fragments are assembled into the

goal modeling - collaborative design method chunk. In this paragraph the method chunk's process and the products are discussed. Figure 4-6 depicts the process flow of the assembled method chunk.

Process: The process of the goal modeling – collaborative design method chunk is presented in Figure 4-6. The process consists of six stages: problem domain analysis, stakeholder analysis, elicitation, modeling, analysis, and collaborative design.

In the problem domain analysis domain information is gathered by analyzing the organization and the organizational domain it resides in. In the PSO under-review the hospital domain is analyzed by reading literature. Process information is assembled by mining documents, interviewing process participants, and observing. Furthermore documents are mined to get a better perspective on the ISD project, previous IS developments and future plans. When the organizational domain is analyzed, stakeholders are analyzed. By conducting unstructured interviews relevant stakeholders are identified as well as their stake in the ISD project. The next step is to elicit goals from the relevant stakeholders a bottom-up approach is selected. The ADR choses this approach because of the many conflicting goals in the ISD project. By eliciting goals bottom-up the underlying motivations and requirements are clearly identified. When the goals from all the stakeholders are modeled in the next stage, the models can be analyzed for conflicts. In this method chunk the ADR team choses to analyze the model informal to create an understandable analysis, instead of a more rigid but less comprehensible analysis, because of the use of the model and analysis in collaborative design sessions. In the latter sessions stakeholders are requested to participate in a collaborative design based on the insights that the goal models and analysis based on these models provide.



**Figure 4-7: Method fragment description: goal modeling.**

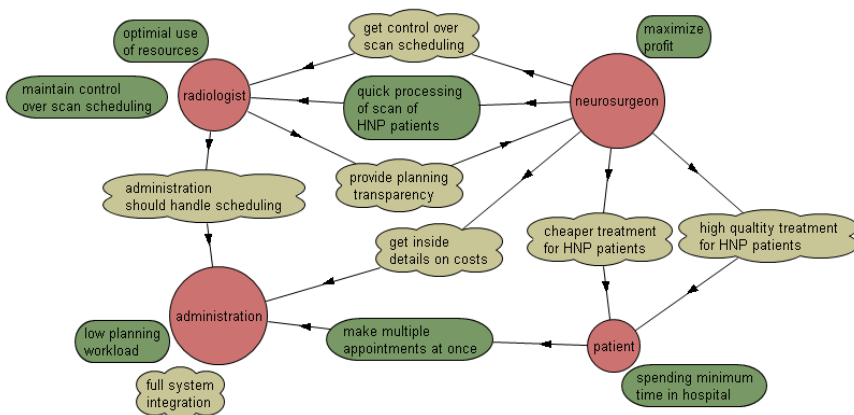
*Products:* The outcome of this method fragment is (1) a goal model depicting the various interests of the stakeholders in the process / organization that is supported by the new to be developed IS, and (2) an analysis of the goal model, either formal (drawing conclusions) or informal (facilitating a discussion). Both outputs are valuable contributions in a requirements analysis stage of an ISD process. Figure 4-7 depicts a description of the method chunk that is now assembled.

#### 4.4 Intervention

This paragraph describes the first application of the developed method chunk. The chunk is constructed in a PSO environment where an ISD process is ongoing. Since this is the first BI&E stage in the IT-dominant ADR process, *the intervention with end-users is not executed to a full extend*. To present a notion of the superficial intervention that is conducted with the constructed method chunk, this paragraph summarizes a prototype analysis that is delivered with the method chunk.

When conducting the goal elicitation in a bottom-up approach, first the stakeholders are to be identified. In the case under consideration, the stakeholders identified are the neurosurgeon, the radiologist, the radiology administration and the patient.

The goal analysis is performed via interviews with the relevant stakeholders and group sessions discussing the results from the individual interviews. Since this is the first iteration in the action design research setting, the end result, as depicted below, is not used in discussions with the end-users. In the first iteration only an alpha-version is developed. The beta-version of the method under-development will be implemented in a much broader organizational setting, involving also the end-users. In chapter 5 the second iteration of the action design research cycle is presented.



**Figure 4-8: Goal modeling – collaborative design method chunk in use: Tropos goal model.**

Figure 4-8 displays an overview of bottom-up goal model with the identified stakeholders along with their private and composite (inter-actor and intra-actor) goals. This graphical model is constructed with the aid of a software tool in which the Tropos methodology is embedded; TAOM4E (Morandini et al., 2008; Penserini et al., 2007; Perini and Susi, 2004).



In this model, the circles represent the identified actors, the rounded rectangles the hard goals, and the cloud shapes represent the soft goals. When a goal is situated between two actors, this means actor A depends on actor B for attaining this goal. A goal is a condition or state of affairs in the world that the actor would like to achieve. A soft goal is typically a non-functional condition, with no clear-cut achievement criteria.

In the HNP case study, several goals can be identified. The efficiency of the HNP patient scan planning and scheduling at the radiology department is critical for the performance of the radiology department. The goal of the radiologists is to maximize the utilization of the resources (like MRI scanners), while guaranteeing best quality to their patients. At the same time the neurosurgeons want their patients diagnosed as fast as possible. For this they want insights and editing rights in the radiology schedule. Next to this, it is important for the neurosurgery department to have clear view on the costs attached to treating patients, and service their patients with the highest level of quality at low costs (thus maximize its profits). The administration prefers a fully integrated system (the radiology system, the neurosurgery system, and the hospital information system are not fully integrated), and a work load evenly spread over the whole day. The goal of the patient is to make multiple appointment (e.g. lab tests, radiology scans) at the same time, and have them succeeding.

## 4.5 Evaluation

This paragraph reviews the goal modeling – collaborative design method chunk that is developed in the first BI&E stage of the action design research cycle. Since this is the first iteration in the IT-dominated ADR approach, the evaluation is only conducted on the ADR team level; the researcher and practitioner level. End users are involved in the second BI&E iteration that is discussed in chapter 5.

In Table 4-4 the original requirements as presented in Table 4-2 are validated against the developed method chunk.

### Requirements for new Validation method fragments

Ability to deal with multiple very different interests in a single process that is to be supported by an IS	The goal modeling – collaborative design method chunk supports the input of every stakeholder involved in the ISD process
Analyze individual / departmental goals for shared interests	Input from every stakeholder is modeled in the method chunk
Ability to explore solution directions	The resulting goal model from the goal modeling fragment is used as discussion platform in the collaborative design sessions
Define solutions based on shared interests	During the collaborative design sessions the goal model is used to discuss multiple solutions that fulfill the goals of the stakeholders
Ability to evaluate individual requirements for the new IS	Since the input from every stakeholder is modeled in the goal model, the eventual resulting decision can be checked against the original input
Ability to integrate the individual requirements into a shared perspective	The goal modeling method fragment that is part of the method chunk, strongly supports the analysis and combination of individual goals into shared goals
Means to discuss outcomes	The goal model that is produced provides a very low entrance platform to discuss the (conflicting) goals that are shared by the individual stakeholders

**Table 4-4: Validation of developed method chunk against original requirements.**

With the implementation of the IT artifact in the case as described in paragraph 4.3.1 the outline of the method fragment was still unclear. Several solution directions, like process modeling oriented

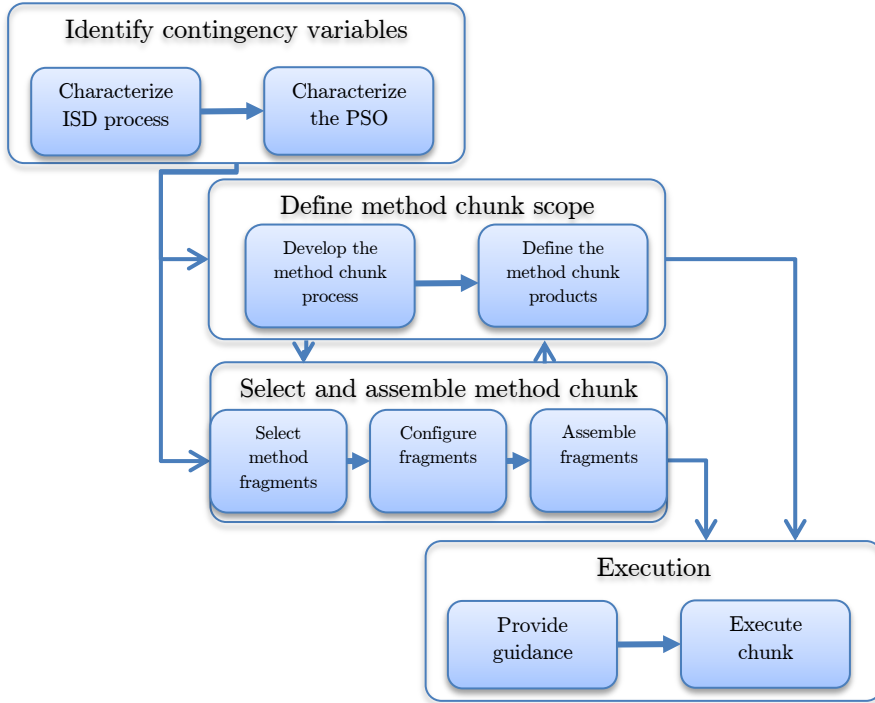
requirements analysis, are explored before the choice is made to select goal modeling as the best fit. Predominantly the technological skills of the professionals in the case organization proved to be one of the main factors to select a more low-technology level method fragment. Initial attempts to model the processes in UML are received with little understanding (researchers were even asked by the professionals to *just write down the process in prose text, so we can understand it*). This can be explained by the complexity of UML diagrams when they are applied to processes in organizations that are heavily interwoven (Moody and Van Hillegersberg, 2009). Because of the identified level of technological skills, the very low-tech goal modeling approach was found to be extremely useful for modeling the conflicts in the processes that were identified to prevent any process improvements.

One of the key issues that immediately caught notion is the lack of common understanding of interests of the other staff working in different departments. Therefore it turns out to be almost impossible to define a list of key issues for the new to be developed IS. Even with the extensive efforts to map processes, business rules, workflows, and procedures, there still remain a *void in the mutual understanding*.

With the selection of goal modeling as a technique, the research increases its focus on ways to *cope with the setting of conflicting goals* in the departments that are the future users of the information systems. Traditional methodologies do not put emphasis on the detection, analysis, and solutions to different goals in an organization.

The goal modeling fragment turns out to be a valuable add-on in the ISD process in this particular case. *The fragment fills a gap in the ISD methodology* for the development of the overall IS that supports both the radiology department as well as adjacent departments. By the use of the goal modeling fragment typical opposing interests in the process are clearly mapped out. In this case, the informal variant of the goal analysis stage is selected. This approach produces a

foundation for discussions about solutions to the opposing interests, before any implementation choices for the IS are made.



**Figure 4-9: Updated approach to develop method chunks.**

*The applicability of the method chunk in this organizational domain is the main positive aspect* of the approach developed so far. However during the superficial implementation and actual usage of the method chunk in a limited organizational environment some flaws draw attention. The main issue is that the ISD project, as is initiated in this organization, is indicated to be following the functionalistic paradigm, according to Hirschheim's classification. In this perspective IS are developed to cope with well-defined problems and follow a well-structured process flow. The goal modeling method chunk that is developed in this chapter does not comply with this paradigm. The method chunk is developed in the social relativism paradigm, for the process is still structured, but it assumes ill-defined

problems that are solved following a subjective approach. Although this method chunk is not fully implemented in the organization, chances are that, because of the ISD perspective mismatch, the chunk will not contribute to the development process.

The second issue that draws attention is the *match between the approach taken in this research to develop method chunks and ISD projects that are ill-structured*. The approach we draw in this and previous chapters is structured in a sequential, traditional process flow.

Chances are that the applicability of this approach is low when encountered with the radical structuralism or the neo humanism paradigm for ISD as described by Hirschheim, referred to in chapter 3. These paradigms for ISD suggest a setting of conflict, approached by ill-structured analysis. Since the first approach to develop method chunks that is applied in this chapter is not developed to deal with ill-structured development processes, the results can be sub-optimal. Therefore we propose an updated approach for the development of method chunks as is depicted in Figure 4-9. In this process flow the stage that selects, configure, and assemble method fragments and the stage in which the method chunk's scope is defined are conducted in a reciprocal style.

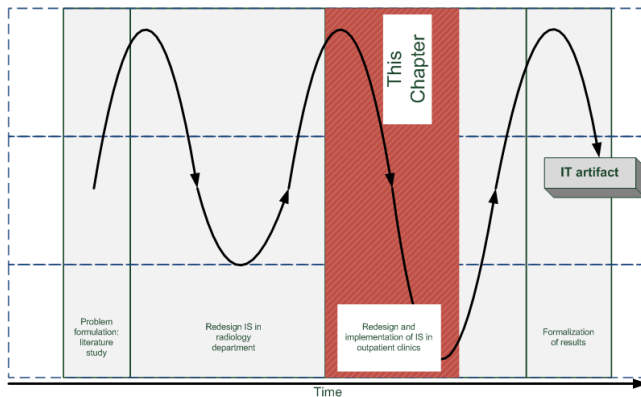
The final main issue that is encountered in this BI&E stage is the *mismatch between the proposed structure of the initial design of the approach to develop method chunks and the operationalization in this BI&E stage*. The process for developing method chunks, as described in the initial design is well-structured. By involving practitioners in the development process, as is prescribed by the ADR paradigm, immediate feedback is received during the development stage. For instance the professionals had a strong preference for the omission of initial analysis of the situation, and were reluctant to use UML type diagrams. In line with the ADR approach, the researchers had to react direct on this feedback. As a result the originally well-structured process to develop method chunks is relaxed. This resulted in a more flexible, but harder to

manage process, which was not beneficial to the project progress. Based on this result the second BI&E stage, discussed in chapter 5, will be organized based on agile principles; allowing both flexibility and a manageable structure.

In the next BI&E stage that is discussed in chapter 5, the lessons learned in this chapter are implemented in an organizational wide method chunk development.

## 5 Building: Redesign of IS in outpatient clinics

In this chapter the second building, intervention and evaluation (BIE) iteration in the ADR framework is described. Just as in chapter 4 a method chunk is



constructed that is useful in a particular organizational case. In contrast to chapter 4 however, this BIE stage penetrates the organization to more extend because of the involvement of the *end-users* in the process. End-users are involved only in the second iteration of the ADR process because this research is labeled as IT-dominant research, see chapter 2.

As with every chapter, at the beginning of this chapter an overview is presented of the research methods that are applied. To avoid overlap with chapter 4, only the extra methods are discussed that are required in light of the end-user involvement.

The same principles apply as are discussed in chapter 4. In the building, intervention and evaluation stage three principles are important: reciprocal shaping, mutually influential roles, and authentic and concurrent evaluation. In the following sections the methodologies that accompany the principles and that are new for this chapter are analyzed. The methods applied in this stage are summarized in Table 4-1.

Method	Goal
Participation ( <i>same as in chapter 4</i> )	Record the effects of the artifact under development and intervene with the artifact in the organization at the same time
Observational research: Unstructured interviews in natural setting ( <i>same as in chapter 4</i> )	Get a better understanding of specific issues in organizational problem domain
Non behavioral observation: record analysis	To find the problems and weaknesses in the current organizational, process, and system characteristics
Behavioral observation: presentation sessions	To align our development process with the organizational domain; to ensure continuous alignment to new insight concerning the requirements

**Table 5-1: Applied methods in the building, intervention, and evaluation stage.**

### ***5.1.1 Method: Non-behavioural observation: record analysis***

A prevalent form of observational research is record analysis. We apply this method to get insights into the current situation; to find the problems and weaknesses in the current organizational, process, and system characteristics. In our research it involves the analysis of historical operational data; process models, patient treatment data, occupancy rates, staff scheduling; but also project documents containing previous systems development projects, process redesign projects and organizational change projects.

### ***5.1.2 Method: Behavioural observation: workshop sessions***

To get initial feedback on the IT artifact under development, the findings are presented in a setting with direct involved participants. The aim of this method is to align our development process with the organizational domain; to ensure continuous alignment to new insight concerning the requirements. Because the aim of this stage of the ADR process is to involve end-users in the development of the IT-artifact, presentation sessions are an effective means to reach the end-users on a large scale at once.



## 5.2 Factsheet

This paragraph presents the inputs used for the second BI&E stage of this research, similar to chapter 3 and chapter 4. Figure 5-1 presents a snapshot of the process-data model that focusses on the second BI&E stage. Below the figure, the inputs and outputs are discussed in more detail.

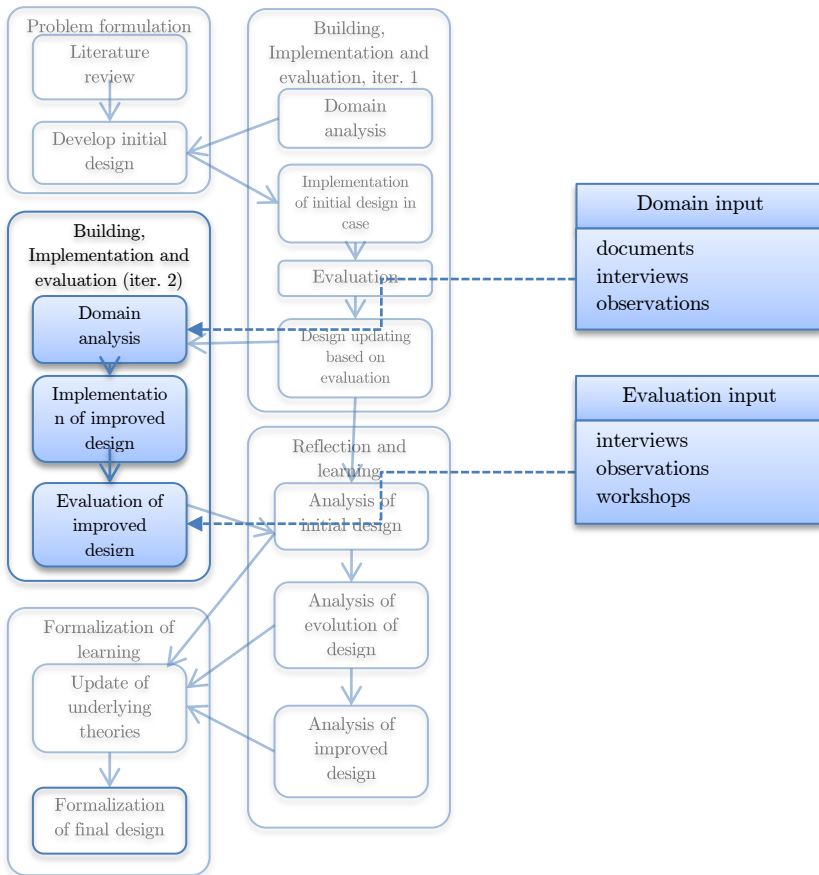


Figure 5-1: Process-data model: Snapshot second BI&E stage

Domain input:

Type	Description
documents	Every available document in regards to the IS development project in the outpatient clinics is analyzed. Process descriptions are analyzed and updated (if necessary), ISD documentation from previous projects is read, information in regards to patient numbers, waiting lists, strategy, manuals, previous conducted questionnaires and research, requirements analysis, and chain integration projects is analyzed.
observations	Observations are done during collaboration and project meetings with the ISD project team, meetings with different stakeholders in the ISD project (administrative and medical), analysis of the primary operations during multiple full-day visits at most outpatient clinics (see Appendix B: Process description outpatient clinics: Gelre case for an overview of the outpatient clinics).
interviews	<p>Many interviews are conducted:</p> <ul style="list-style-type: none"> <li>• Structured interviews with at least one stakeholder of every outpatient clinic involved in this ISD project</li> <li>• Unstructured interviews with : HR project team members, IT department, in-house logistics experts, facility management, and medical staff in other hospitals that operate the same model</li> <li>• Structured patient surveys (n=380)</li> </ul> <p>The structured interviews are processed in a transcript, for every unstructured interview minutes of meetings are produced. Patient surveys are statistically analyzed.</p>

Evaluation input:

Type	Description
workshops	<p>During the course of this research (one and a half year) six workshop types are organized, some of which are repeated multiple times with different groups:</p> <p>WS: initial design presentation and evaluation (audience: approx. 20 administrative &amp; medical management)</p> <p>WS: presentation of initial analysis and model validation (audience: approx. 30 administrative and medical staff)</p> <p>WS: presentation data analysis, further validation of model (audience: approx. 10 administrative management)</p> <p>WS: collaborative design with other project teams (audience: approx. 10 project employees)</p> <p>WS: presentation of final analysis and conclusions (audience: approx. 40 medical and administrative staff and management)</p> <p>WS: application of method chunk as part of ISD project (audience: approx. 30 medical staff, workshop repeated 4 times with different medical groups)</p> <p>For every workshop is a transcript is produced, and during the workshop notes are carefully taken.</p>
observations	<p>Observations are done during project meetings with the ISD project team and other project teams, presentations sessions and workshops, meetings with different stakeholders in the ISD project (administrative and medical), and analysis of the primary operations during multiple full-day visits at most outpatient clinics (see Appendix B: Process description outpatient clinics: Gelre case for an overview of the outpatient clinics).</p>
interviews	<p>The project is finalized in a report. This report is used</p>

to discuss the outcomes with several key stakeholders in this project. This is done in unstructured interviews:

- Two interviews with heads of outpatient clinics
- One interview with administrative management

### 5.3 Building

The organization selected for the second BI&E iteration is an implementation of a new information system for the outpatient clinics in a mid-sized hospital in the Netherlands (Gelre) (Rothengatter et al., 2010). In this BI&E stage a method chunk is developed that integrates with the overall ISD methodology in place.

In chapter 4 a method chunk is constructed based on an initial approach for method chunks construction. This approach is developed based on a literature review. During the development of the method chunk, the approach to construct this chunk evolved to a more mature design in accordance to the experiences of applying it in the organization. This paragraph describes the second method chunk construction process in the ADR process. The result of this stage is therefore again a fully developed method chunk.

The ADR team in this research consists of two researchers and practitioners employed by the hospital. The researchers participate as experts on ISD in the project and as team members of this project team. The practitioners are either employed as full time project leaders or as executive staff in the organization. In this second BI&E iteration end-users are actively involved in the development process. The end-users in this case include medical specialist, medical support staff, and administrative staff. The end-users participate actively in the development and implementation of the method chunk.

This paragraph is structured according to the approach that is developed to construct method chunks, as is initially developed in chapter 3, and thereafter updated in chapter 4. In the evaluation of the goal modeling – collaborative design method chunk in chapter 4,

we concluded that a very structured and sequentially (and traditionally) organized process for the construction of method chunks is not very feasible in dynamic PSO environments. Therefore a more reciprocal process is developed at the end of chapter 4. The process consists of the same stages as the original approach and consists of four stages; (1) the identification of contingency variables; (2) the definition of the method chunk's scope; (3) the selection and assembly of method fragments into a chunk; and (4) the implementation of the chunk in the organization. In the research in this chapter the stages are executed in a reciprocal flow, for visualization see Figure 4-9 on page 140. For readability purposes this paragraph is structured in the traditional sequence.

### ***5.3.1 Case description***

The second iteration of the ADR approach to develop a method chunk is carried out in the outpatient clinics of a medium sized hospital (250 beds) in the Netherlands. Hospitals are characteristic instances of the PSO domain, as is discussed in chapter 3. This research is conducted in the timespan from November 2008 to February 2010. The research project focuses on the outpatient clinics of this hospital with over 210,000 patient treatments yearly. The hospital, aiming at higher efficiency and patient centric care, is planning and implementing major changes in the information systems, the organizational structure, and the care and administrative processes. The changes are triggered by the construction of an entirely new facility that is constructed during the period mid-2008 to fall 2010.

The hospital organization faces major uncertainties because of the substantial changes in delivering health care service to its patients that are effects of the new facility. In the new building physical allocation is organized differently, which influence processes, workflows, and communications. The hospital staff is confronted with questions like: *will we be able to deliver a continuous and stable output in patient treatments?*, *Does the future building layout provide enough capacity and potential to expand production?*, *When*

*do we need to register a patient, or will a patient register him/herself?, and Who is responsible and accountable for patient throughput, and the utilization rate of shared resources?.*

The hospital staff's main concern is caused by the future transfer to a brand new facility. This facility is based on the *21st century airport operations* management concept for the design on the outpatient clinics (Vos et al., 2007). Capacity in this new facility is limited in terms of available space. The limitation of space is the reason that each medical specialism has to rethink its utilization of available rooms, waiting areas, and the use of shared resources. These issues, in their turn, lead to rethinking the workflows, patient logistics, and scheduling. Naturally the workflow, logistics and scheduling are all facilitated by IS. To induce the transfer from the current to the future situation, numerous projects are initiated. One project in particular is the patient logistics project for outpatient treatments. In this project the proper allocation of physical space for patients spending their time awaiting their treatment is designed. Because of limitations on available square meters, waiting areas have to be shared amongst multiple departments. To accommodate longer stays of patients before or during multi-stage treatments, a comfortable central waiting area is situated near the entrance of the hospital. This design is as stated very similar to 21st century airport layouts. As will be discussed below, the future layout will have an enormous impact on the hospital's information systems, processes, and organization structure. The staff will have to rethink, and redesign the very basics of appointment scheduling, last-minute scheduling, and communication with other departments.

To support the future processes and organization the current IS is to be replaced by a new IS that is to be developed and implemented in this organization. When the researchers are initially involved in the ISD project in this organization no definite decision is made about the type of IS to be implemented, nor about the processes and organizational structure that is supported by the new IS.

A detailed view on the processes and patient numbers in this hospital is presented in Appendix B, on page 252.

The research project in which the method chunk is developed that is presented in this chapter, is part of a much bigger organizational development project. Many smaller projects are initiated to cope with specific aspects of the transition the hospital is in. For instance some projects focus on human resource development, the configuration of the newly to be implemented ERP system, the right layout of the outpatient clinics' specialists rooms, the development of indication about routing for the patients in the new building, and many more. The project that is discussed in this chapter focusses on the patient waiting system that is connected to the newly to be implemented ERP system.

### ***5.3.2 Contingency variables: case analysis***

This case focuses on the redesign of the IS that is tailor made to the hospital processes and aimed to increase the hospital's efficiency and quality. As is discussed in chapter 3, IS can be analyzed on three layers; (1) the deep structure; (2) the surface structure; and (3) the physical structure, see paragraph 3.2.1. This ISD process focuses on the deep structure of the IS, meaning the aspects of the IS that reflect the represented domain (Wand, 1996). In the hospital domain processes are usually complex, multi-disciplinary, and knowledge intensive. Traditional redesign approaches encounter difficulties coping with the complex hospital environment because of their one dimensional perspective on the problems in healthcare. Traditional methods often only address a single facet of organizational processes, by which solutions are designed for problems in one area and creating issues in other areas.

In the research we analyze all processes, systems, and employee roles in the outpatient clinics in a hospital based on data about actual operations and realistic patient treatment information in order to reengineer the processes. In short we analyze knowledge intensive processes in a complex, multidisciplinary organization in order to

improve quality and efficiency. Because of the action design research approach workshop sessions are included in which every stakeholder group in the hospital is represented.

### **ISD perspective**

In chapter 3 a characterization is drawn on perspectives on information systems development. According to the Hirschheim's paradigms, the ISD project in this case classifies as *neo humanism*; the information system is developed to remove distorting influences and other barriers to rational discourse (Hirschheim and Klein, 1989). The ISD approach taken in this organization is the soft – ill-structured analysis; the ISD process is not structured according to predefined structures, nor formally documented. The soft systems perspective is applied, in which the insights of various stakeholders are required, see paragraph 3.2.4. At the start of the research project the development direction for the IS under-review is not defined yet, nor is the approach to execute the project.

ISD methodologies that therefore could classify as appropriate are the agile methods like dynamic systems development method (DSDM). However in this organization there is no specific methodology or methodology class adopted to deal with the transition. Next steps in the development project are taken ad-hoc.

### **PSO characteristics**

Of all highly complex PSOs, hospitals can be ranked as one of the most complex (Hafferty and Castellani, 2010). There are two main ingredients that create the complexity in hospital organizations: the organizational structure, and the process structure. The difficult organizational structure is widely discussed in literature (Glouberman and Mintzberg, 2001a). In this section the organization under-review is analyzed using the four characteristic areas of PSO that are modeled in Leavitt's model, as describe in chapter 3. The four characteristic areas are information technology, tasks, people, and structure.



*Information technology:* Information technology is identified as the means for a PSO to coordinate its complexity. This statement holds for this organization; all the outpatient clinics share some resources, like the radiology department, surgery rooms and a blood test room. In the future situation more resources are going to be shared, like consultation rooms, waiting areas, and even the services of the supporting medical and administrative staff. The information technology in place is not designed to cope with this complexity. In the future a new IS is to be implemented that is based on the enterprise resource (ERP) principles. Although literature does not advise EPR based systems to be used in hospitals, since it is based on a bill of materials rather than a bill of resources, the hospital in this case is planning on adopting this system. The particular characteristic processes of the outpatient clinics have to be partially adapted to work with the systems, partially the system needs to be adapted to fit the processes.

*Tasks:* Processes in this organization are defined but predominantly in the minds of the staff working in the clinics. During the course of this research many interviews are conducted to precisely model each process. It turns out that processes are not systematically modeled and there is no awareness across the clinics boundaries about the exact process configuration. The process flows for every clinic seem pretty predictable. The number of emergency treatments and no-shows can be predicted on a daily basis to some extent, besides the regular patient stream that is consistent over the various days of the week. There is however no proper insight in the various different treatments that are conducted during the course of one day. Techniques like sequencing and batching are not introduced.

*People:* As in every hospital, the management and board on the one hand try to optimize on output performance measures, while doctors and nurses on the other, aim to improve the in-process performance measures. The aim of management could result in an overemphasis on KPI's (Propper et al., 2008), the aim of the medical staff could result in a *flat of the curve medicine*, where spending on medical care increases even though the additional gains from such spending are

very low or non-existent (Enthoven, 1980). In this organization the medical staff is partially self-employed, partially employee of the hospital. This induces that the main characteristics of PSOs, as identified in chapter 3, hold for this organization; the staff is self-responsible, prefers professional autonomy, and quality is a more important process measure than quantity.

*Structure.* Management in a hospital is not homogeneous, but can be perceived as several distinct groups. Glouberman and Mintzberg identified four worlds of the general hospital: The Community (board of directors), the Control (administrative management), the Cure (doctors) and the Care (nurses) (Glouberman and Mintzberg, 2001b). The organizational structure in the medical domain is set up to protect the doctor from behaving as an economic man (Harris, 1977), however it can result in a fundamental setting of conflicting goals (Cutler, 2002). This organization is characterized by very low output efficiency; average resource utilization on some resources like consultation rooms is as low as 30% on average. As discussed are the processes complex and interwoven with many other departments. The problem in this organization is the lack of awareness about the actual resource utilization, the process configuration, and the connection of processes to the processes of other departments.

### **Hiatus in ISD process**

In paragraph 5.3.1 we describe the IS development in this organization. As is discussed in that paragraph, the current research project is part of a much larger organization development project, involving ERP system implementation, human resource development, and many other projects. This research project deals predominantly with the development of the patient waiting system as that is connected to the newly to be implemented ERP system. The dynamics of this project are therefore tightly connected to the overall project portfolio. The traditionally hard systems approach that is chosen to deal with the development of IS in structured organizations is not applied in this organization, since this approach often results in a transfer of the local problems to an adjacent department and a lack of attention to support participation of

various stakeholders. This organization chooses to adopt the soft systems approach, with which the importance is emphasized of involving every participant in the development of a solution for the problems that are described in the case description. This implies that in the method fragment every staff member of the outpatient clinics needs to be involved in the development of a new organizational structure, care processes, and information systems.

Similar as the analysis of the case under-review in the previous chapter are the voids in the ISD process in this organization predominantly determined by observation: working in the project team that is developing the new IS, and observation of the end-users in this organization. Furthermore are documents in regards to the ISD project in this organization reviewed and stakeholders (professionals as well as end-users) are interviewed. The voids in the current ISD process are the lack of a shared perspective on the future organization of IT and processes. Processes need to be improved before they are automated. However, staff members have no accurate information about actual flows of process-chains. Since the treatment of a patient generally passes through multiple departments, an overview needs to be created in order to optimize processes before automating them with the means of an IS. Because there is no solution direction chosen yet in the ISD process, resistance against the ISD project grows. The current ISDM in the project does not provide any platform to facilitate the analysis (with every stakeholder) the feasibility of development options.

Problem Requirements for new method fragments

Lack of definition and awareness of current processes that are to be supported by the newly developed IS	<ul style="list-style-type: none"> <li>- Ability to model processes</li> <li>- Ability to present process models to other stakeholders in a comprehensible way</li> </ul>
Current processes are sub-optimal. There is a lot of room for	<ul style="list-style-type: none"> <li>- Method chunk should facilitate process improvement analysis</li> </ul>

improvement, especially with the new IS.	<ul style="list-style-type: none"> <li>- Ability to analyze influences of new IS on processes</li> <li>- Means to discuss outcomes</li> <li>- Means to co-design future processes</li> </ul>
Solution direction for new IS is not yet defined	<ul style="list-style-type: none"> <li>- Ability to explore solution directions</li> <li>- Define solutions based on shared interests</li> </ul>

**Table 5-2: Requirements based on Gelre case analysis.**

The contribution of the method fragment under-development should be to facilitate in the ISD process a redesign and transformation of the hospital processes and organization to increase efficiency. For this research efficiency is defined as patient waiting time, and percentage of shared resources utilization.

### ***5.3.3 Method chunk's scope***

The next stage in the development of the method chunk that suits the requirements in this organization is the definition of its process and products. As is concluded in chapter 4, the best approach to develop method chunks is to conduct the stage *define scope* and the stage *select and assemble method chunk* reciprocally. Therefore the method chunk's scope is defined parallel to the definition of the selection and assembly stage that is described in the next paragraph. In the current paragraph the outline for the method chunk's process and the products it should deliver are defined.

### **Method fragment process**

The begin-state of the method chunks process is the situation in which no decisions are made about the direction the ISD process will be further developed. The end-state of the process is the definition of the solution direction of the overall ISD process. In the method chunk's process it is important to include every stakeholder in the development of the IS. Because any exploration process is rather unstructured, the method chunk's process should be also open to an

unstructured approach. Depicted in Figure 5-2 there are three stages in the process of the method chunk; (1) the development of a platform; (2) the analysis and optimization of current processes and IS; and (3) the development of a shared vision on the direction taken in the further IS development process.

### Method fragment products

The method chunk under-development should produce a platform to review current processes in light of the new building facility and IS. The platform should facilitate the analysis, optimization, and discussion about possible solution directions available. The use of the platform itself could be a means to deal with resistance in the organization, because much resistance to the ISD process is caused by staff being not aware of developments in the ISD process.

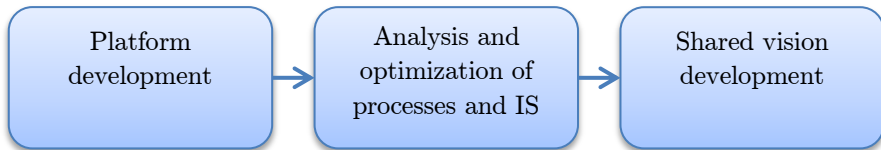


Figure 5-2: Initial design method chunk's process flow.

#### 5.3.4 Selection, configuration, and assembly

In this paragraph the method chunk is further developed based on the contingency variables and subsequent requirements and the scope set in previous paragraphs. Three topics are discussed in this paragraph; the selection of method fragments that match the requirements, the configuration of the method fragments to fit the organization characteristics, and the assembly of the fragments into one method chunk.

##### *Selection*

There are two fragments selected for this method chunk: simulation based analysis of processes, and collaborative design. The fragments address the problems that are described in the case description as a result of the multi-disciplinary, complex, and knowledge intensive

organization- and process structure. Since collaborative design is discussed in depth in chapter 4, only the simulation fragment is treated in this section.

## **Simulation**

In this method fragment simulation as a tool is selected to support the development of transition to the future setup. In this case visual 2D and 3D simulation of primary processes is selected because of:

- Analyze various scenarios. The ability to simulate a set of alternative solutions. In major transitions, like the case under review, many process, structure and system variables are not identified yet, and if they are identified, the value of the variable is not yet known. Simulation provides the ability to analyze different scenarios, include a broad array of variables, and test different values per variable in order to improve the current situation.
- No interpretation skill requirements. Since the aim of the simulation technique in the method fragment is a high educational value of the simulations to be developed, the results of each simulation run can be interpreted by every stakeholder in the hospital. Therefore the opinion and ideas of each individual stakeholder can be grouped into a scenario and analyzed to test the effects on the overall efficiency. With the use of simulation in the development process to improve the efficiency of the outpatient clinics, the development team is able to create a firm basis for discussion about the solution. The two-dimensional and three-dimensional visualizations results in low interpretation skill requirements; every participant can join in on the discussion. This induces a collaborative, co-constructive setting.
- Ability to analyze complex, multidisciplinary, knowledge intensive processes. The connectedness of the organizational departments creates a complex solution space. As discussed

when describing the hospital complexity, departments are interdependent in terms of staff, patients, shared resources, financial budgets, and space capacity. A proposed change in one department can influence many other areas, similar to non-linear feedback loops. Simple statistical analysis is not sufficient in these situations.

- Relatively low costs. Analyzing and interpreting the different alternatives demands many resources, and is therefore quite intensive. However, the costs are, compared to the costs of a trial and error approach relatively low. Next to that we state that simulation provides us with the tool to analyze a very broad set of scenarios, leading to a more optimal solution design.

**Configuration**

Both method fragments need to be configured to meet the specific characteristics of the Gelre organization in which the ISD project is conducted. In this section the configuration of the method fragments is discussed.

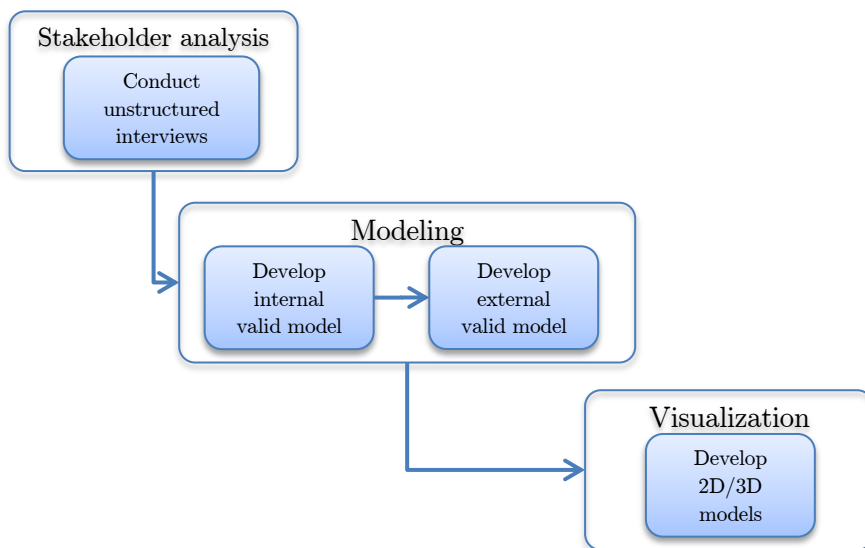
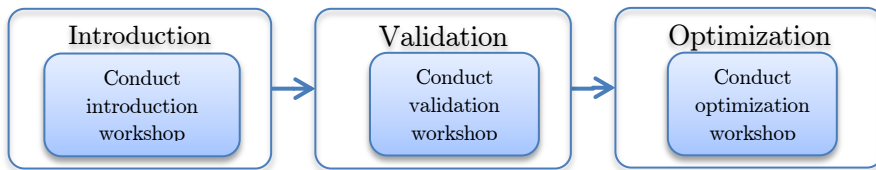


Figure 5-3: Simulation method fragment - general configuration.

Since this BI&E stage is the second BI&E iteration in the overall ADR cycle, the intervention in the organization using the method chunk under-development involves next to ADR team (consisting of researchers and practitioners) the end-users. This intervention penetrates the organization to a further extent. Therefore the collaborative design method fragment is applied to its full advantage, in contrast to the application of this method fragment in chapter 4. The general configuration of the simulation method fragment and the collaborative design method fragment is depicted in Figure 5-3, respectively Figure 5-4.



**Figure 5-4: Collaborative design method fragment - general configuration.**

### **Simulation**

*Stakeholder analysis:* is discussed to its full extent in chapter 4, paragraph 4.3.4.

*Modeling:* Simulation models are often used as a method in problem solving and decision making (Sargent, 2005). In modeling a real world problem there is always a trade-off between types of validity, amongst others the internal and external validity, the educational validity, and representational validity (Tsjernikova, 2009). Educational validity means the level of complexity and to what extent the model still can be understood. Representational validity implies the actual model similarity with the real world. In this research, initially the situation is modeled in a very generic, simple, and well understandable model. This generic model contains a high educational validity. In follow-up versions the level of detail increases and the scope of the model reduces. Hence, focus in this simulation project moves from initially the educational validity, to the representational validity.



*Visualization:* The results from the simulation are used as a platform, either in graphs, figures or movie clips during collaborative workshops to inform the participants about problems, uncertainties, and opportunities. The software simulation tool provides various forms of interactive input processing. The involved participants interact easily during the analysis of different solutions setups, as the model offered to swiftly alter for instance constraints and variables. To support the simulation technique dedicated simulation software tools are selected (Legion Studio and Flexsim Healthcare) to build the simulation model, and therewith model the movements of patients.

There is a very good reason to use physical patient movements, as offered by the Legion Studio and Flexsim Healthcare software suites, as one of the main indicators of the efficiency of the solution. In this method chunk, results in the physical domain are used as one of our key performance indicators. These indicators are affected by the process efficiency, and effectiveness of both the information systems and the organizational structure. In health care these physical indicators are often of major importance. The following indicators are included in the simulation model:

- Occupation rate and utilization of waiting facilities,
- Utilization of expensive equipment,
- Patient throughput,
- Waiting times (both for patients and medical staff)

These measures are incorporated in the simulation models design, analyzed by means of the simulation software, and evaluated in workshop sessions with the process participants. In this case study, the combined effects of a change in the facilities layout, the supporting information systems, and the accompanying adjusted organizational structure, are analyzed using the software simulation tool.

Initially, patients, visitors, and medical staff are represented as multi-agent actors on a two-dimensional layout in Legion Studio.

The agents are modeled according to the believe, desire, intention (BDI) design (Rao and Georgeff, 1995), hence simulating individual's behavior. In the more advanced versions of the simulation specific parts of the simulation are converted to the three-dimensional simulation environment in Flexsim Healthcare, increasing the educational validity, and of course making it more appealing for staff members too. The simulation environment is developed in iterative cycles. The project is initiated by building a very generic model, which is demonstrated to participants of a representative group from the organization. In consecutive cycles we further specify the model in accordance to available design options, results of simulation analyses, and evaluation by the hospitals representatives. Because of the abstraction layer that is required when modeling the full range of outpatient treatments in all departments, we received much feedback considering details. This feedback creates a focus on specific problematic sections of the building's layout and processes. The initial model is aimed at displaying patient flows on the future layout of the entire hospital building, in the next steps we further zoom in on high throughput areas, then on a group of outpatient clinics working together, and finally, only on one outpatient clinic. The level of detail achieved in the last simulation is more than sufficient to prove to the staff the model's internal validity.

*Software:* Legion Studio is primarily developed for simulation of pedestrian flows. This software tool is used to model the movement of patients within the compound facility. Legion Studio is a two-dimensional visualization software tool. It is capable of displaying real-time patient movements on two-dimensional layouts, and generating overviews of Monte Carlo simulations (Rubinstein and Kroese, 2008). Using Legion Studio we generate actual information about the number of patients occupying specific rooms, corridors or waiting areas in the future situation. A simulation model is developed to analyze patient movement in the newly constructed hospital, based on the current CAD drawings, actual patient treatment data, and process flows. The simulation model developed based on Legion Studio also provides detailed insights into the time

patients (individually or on average) are in transit, are waiting for treatment, and are in the facility in total. Even in depth variables like the comfort patients experience based on the proximity of other patients can be modeled. Flexsim Healthcare is a developed to analyze the appropriateness of compound facilities' physical layout. We use this simulation software product to analyze the occupancy of equipment and shared resources. But the best asset of Flexsim Healthcare is the three-dimensional visualization capability. The use of three-dimensional visualizations result in models that are more interesting for workshop participants to review.

### **Collaborative design**

The collaborative design method fragment consists of a series of workshops. The number of workshops is determined based on the size of the organization and the number of stakeholders involved. The ADR team identified the number 30 as an optimal number to involve as many stakeholders in one session, while maintaining a collaborative, proactive attitude. More participants in one workshop would diminish the productivity of the session, less participant would diminish the collaborative / multi-disciplinary aspect of the workshops. During the workshops the results from the simulation modeling fragment are used as input for developing the organization (i.e. the processes and IS). There are three distinct workshop types; introduction workshops, validation workshops and optimization workshops. In the ISD project these three types are used as a basis for six workshops that are organized over a period of nine months:

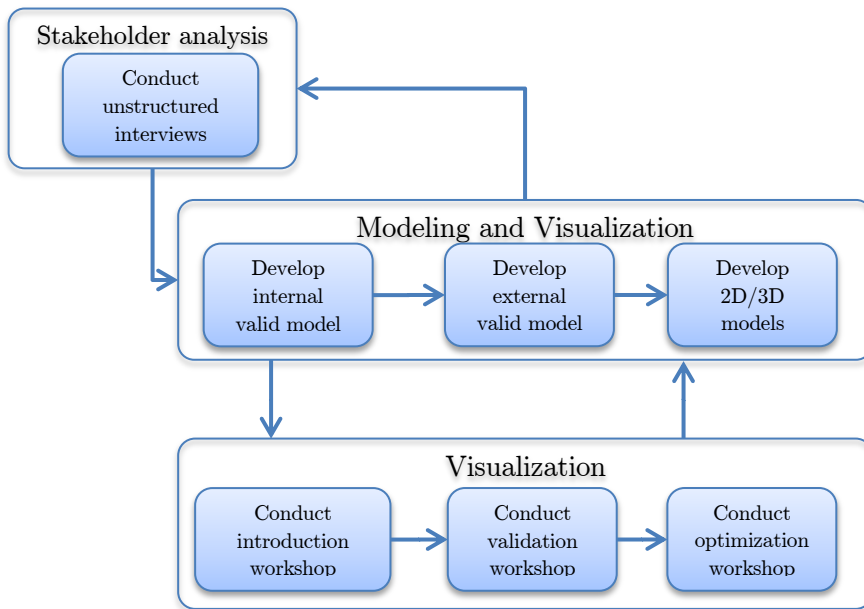
- Workshop 1: The goal is to initialize the development process and present an initial design; we present a basis simulation model to set a foundation for the following sessions. The audience in this session consists of physicians and administrative management.
- Workshop 2: In this workshop we aim to validate the model we create based on interviews, process descriptions and real-time patient treatment data. We use the results from this session to further enhance the simulation model. Furthermore

we discuss with the audience, consisting of physicians and administrative management the means and ends in primary care processes.

- Workshop 3: The goal of this workshop is to design new processes and redefine tasks and roles, based on the outcomes of the simulation model. Our target audience in this session consists of administrative management.
- Workshop 4: Dissemination of results: we provide the audience with a detailed analysis of processes, now and in the future situation. We use the workshop to get feedback on the model and conclusion. The results are used as input for further detailed design of process: the identification of decision making. The audience consists of physicians, nurses, and administrative support staff. We organize this session twice.
- Workshop 5: Designing future organization: future planning process, systems and organizational structure are designed by using the results of the detailed simulation and from workshop 4 as input. Audience in this session consists of administrative management.
- Workshop 6: Information dissemination and further design: future process and layouts are determined, in this session the processes, systems, and organization are developed in a collaborative session. By means of collaborative design, we aim to increase the acceptance rate of the new design. Audience in these sessions (organized four times) is physicians, nurses, administrative staff, and administrative support.

### *Assembly*

The simulation method fragment and the collaborative design method fragment are assembled into one method chunk in this section. In this section the process and products of the newly developed method chunk are discussed. The chunk's process is depicted in Figure 5-5.

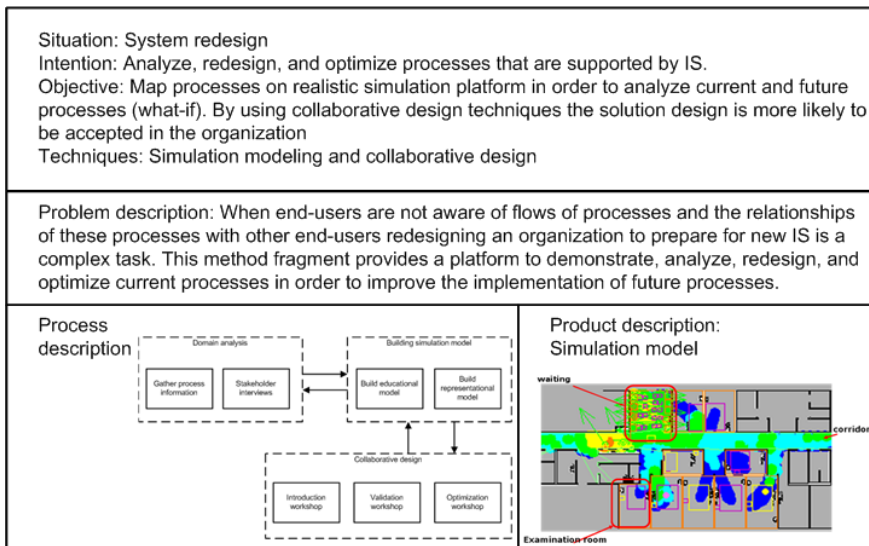


**Figure 5-5: Simulation and collaborative design process flow.**

In Figure 5-5 three types of workshops are mentioned. In the ISD project the ADR team is involved six workshops are conducted with regards to the focus of this process; the IS support of the new patient logistics in the outpatient clinics. Some workshops are repeated more than once to ensure every stakeholder is given the opportunity to participate in a collaborative design session (see the remarks about the number of participants at the introduction of the previous section).

In our approach we aim to combine multiple methods' perspectives; we include spatial analysis, detailed historical patient treatment data, collaborative design, visual simulation, and end-user involvement to analyze, design and transform processes in the hospital organization under-review. The main analysis instruments and design artifacts are simulation models that were created as input for a thorough analysis of the processes in place. This method chunk offers a means to address resistance in organizations that is caused by unawareness of current process flows. This resistance prevents

further analysis of improved designs for processes and the IS aligned with the processes. A simulation model is used as a platform to analyze, redesign, optimize and implement current (redesigned) processes. Because of the use of the collaborative design method fragments, it is more likely that the improved design is accepted in the PSO. The simulation modeling is supported by software tools like Legion Studio and Flexsim.



**Figure 5-6: Method fragment description: simulation and collaborative design.**

*Process:* The process of the simulation / collaborative design method chunk is not sequentially structured, see Figure 5-5. There are three stages in the process; (1) domain analysis, which is done by an analysis of the processes and stakeholder interviews; (2) the building of a simulation model that consists of building an educational, representational model, and visualizations; and (3) the collaborative design that consists of at least three different approaches to introduce, validate, and optimize the design under-review.

*Products:* The outcomes of this method chunk are a (1) simulation model in various stages / comprehension levels; and (2) an optimized

design of current processes, departmental layouts, and IS support for the processes. Figure 5-6 depicts the method chunk's summary.

## 5.4 Intervention

This paragraph describes intervention with the developed method chunk in the organization under-review. The approach to develop the method chunk is based on literature and insights from the first BI&E iteration in chapter 4. In this intervention the method chunk supports an organizational-driven reengineering process of the IS support for a complete set of outpatient department of mid-sized hospital. The application of the method fragment shows positive results on the organizational learning process and the design outcomes. The results of the method chunk used in the ISD project are twofold; the outcomes of the simulation model by which the process efficiency and quality are improved, and the outcomes of the use of the method to improve the hospital's processes and organization. Both outcomes are discussed in this paragraph.

*Hospital efficiency.* As is stated before, the design of this simulation is done in iterative cycles. The first model and analysis is built with a broad scope on a high level of abstraction, following models are more focused on problematic areas. The final model, in which results are conveyed to the complete hospital staff, is focused on one specific outpatient clinic; ophthalmology. The results from the analysis of this department are considered to be exemplary for every outpatient clinic in this hospital. Because of the high level of detail and the small scope included in this final simulation model, the results are easier to understand for a broad audience. Furthermore the high level of detail provides a high level of trustworthiness for the medical staff. The overview of the simulation results in this section only discusses the outcomes of the ophthalmology outpatient clinic. The results of the simulation consist of:

- Occupation rate of the waiting area (with a limitation on available seats) near to the clinic

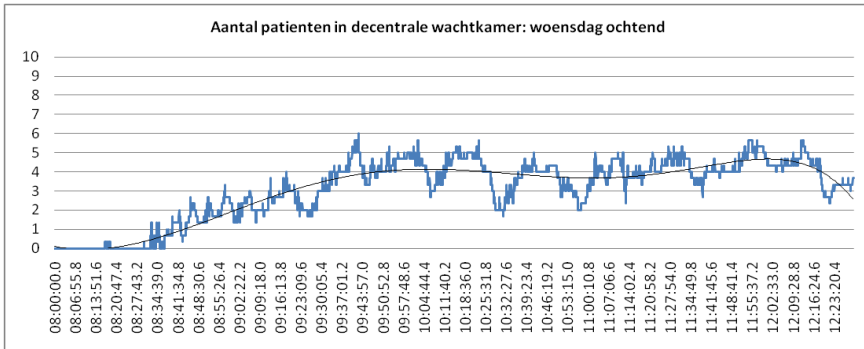
- Occupation rate of this clinics examination room
- Heat map of the corridor adjacent to the clinic
- Movie clips displaying a glimpse of future situation, both in 2D and 3D

A detailed description of the process that is modeled in this simulation is presented at: Appendix B: Process description outpatient clinics: Gelre case. The four outcomes of the simulation are the prime interests of the stakeholders in this organization. As is discussed in the case description in paragraph 5.3.1, the basic interest of many stakeholders is on the feasibility of the new processes and IS that is being developed. The four selected outputs from the simulation environment present some key indicators that the future configuration of processes and information systems is feasible.

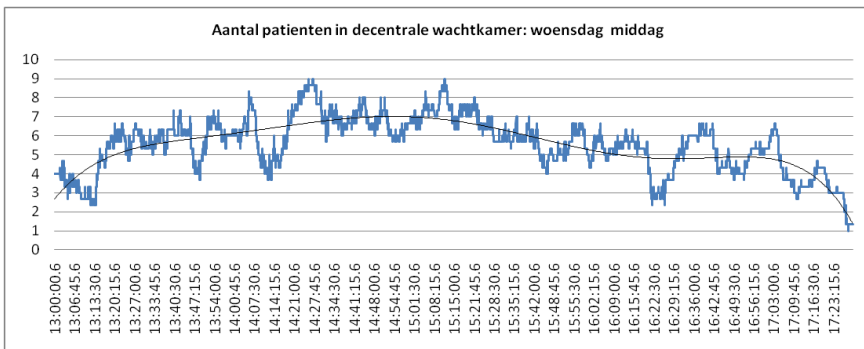
The following graphs and overviews are exemplary of the data set the simulation runs produce. The results from the simulations are shared in the collaborative workshop sessions with the specific healthcare professionals that are affected by the specific results.

*Waiting occupation:* Because of the limitation of available space in the new building and consequent maximal waiting capacity, the occupation rate of the waiting area is of high importance to the medical staff. Each outpatient clinic has a maximum availability of 18 seats per waiting area. In the simulation model only patients are modeled, accompanying visitors are left out of the treatment process. To make up for these persons, the hospital management determined a *1 simulation agent = 1.75 real life person* allocation, based on prior experience. Figure 5-7 and Figure 5-8 show the occupation rate of the ophthalmology waiting. Both figures show the amount of simulation patients populating the waiting. Hence to come to real life persons, the numbers should be multiplied by 1.75. As shown in Figure 5-8, the most occupied period is between 14:15 and 15:15 hours. The maximum amount of patients in that time frame does not exceed 9, which implies nearly 16 persons.





**Figure 5-7: Simulation method fragment: Occupation rate waiting; morning only.**



**Figure 5-8: Simulation method fragment: Occupation rate waiting, afternoon only.**

*Examination rooms:* A second concern is the examination room occupation by patients. This concern is also expressed by the medical staff. When the patient logistics system does not function to the full extent, and the patient throughput from the central waiting area (near the central entrance) to the examination rooms falters, medical specialists are waiting for patients. The medical staff considers this situation to be highly undesirable in this hospital. When a process is designed such that the medical specialists don't have to wait for patients, their requirements are satisfied. The results in Figure 5-9 and Figure 5-10 display primarily a repetitive scheme, which makes a constant flow of patients very likely. The number of patients treated by the medical staff was certainly acquired during the simulations. Earlier investigations about the occupation rate of examination

rooms in this hospital made it clear that a level over 35% was not encountered in any in outpatient clinic. The results from these figures therefore seem to match reality. The processes are designed such that at least the patient logistics system’s performance does seem to suffice.

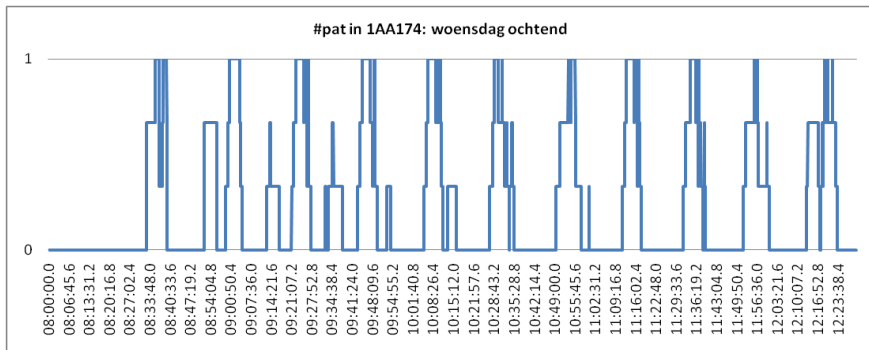


Figure 5-9: Simulation method fragment: Occupation examination room (1AA174) morning only.

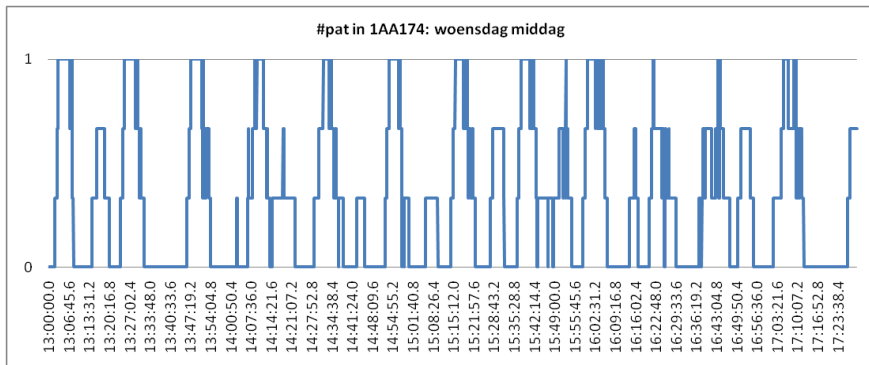


Figure 5-10: Simulation method fragment: Occupation examination room (1AA174), afternoon only.

*Heat map:* The heat map displays the maximum patient activity in the ophthalmology outpatient clinic on the future building layout. The more the color diverges from green, to blue, to yellow, to red, the more activity is forecasted on a specific area. Figure 5-11 displays this one outpatient clinic, inclusive of waiting, administrative desk,

and examination rooms. These kinds of images are used during workshops to convey an outlook for the future situation.

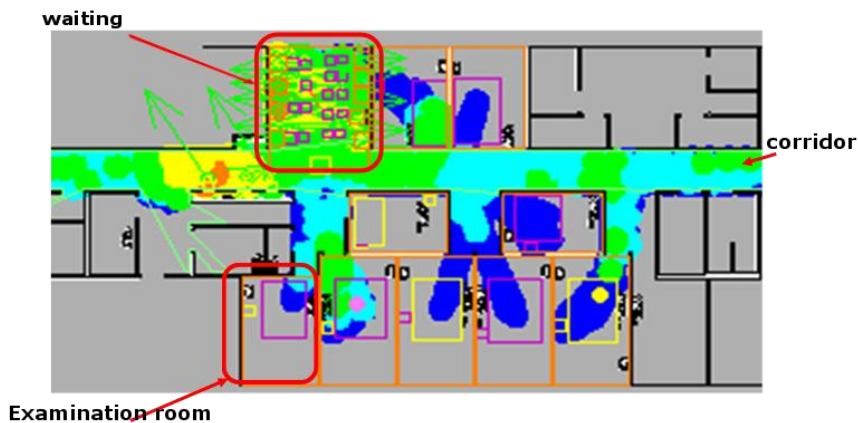


Figure 5-11: Simulation method fragment: Heat map ophthalmology outpatient clinic.

*Movie clips (2D and 3D):* The overall simulation results are presented both in hard data and graphs, but foremost as movie clips showing a possible outlook on the future facility. Initially we were quite reluctant to demonstrate this view to the medical and administrative staff, because they are only a snapshot; it is hard to draw conclusions based on snapshots. Initially the research team assumes that the use of movie clips that have a duration of about 5 minutes do not add any value in demonstrating the future process flow; it would take hours to get a proper idea of the future situation. However, we notice remarkable enthusiasm after a brief try out. Although lacking educational validity, the representational validity was enough to make these movie clips useful. Figure 5-12 displays shots from the 3D environment that is applied.



**Figure 5-12: Simulation method fragment: 3D results from outpatient simulation (screenshots).**

## 5.5 Evaluation

This paragraph describes the application of the method chunk that is developed in this chapter. In this ADR iteration a method chunk is developed for the second time in an organizational setting. The results of the method chunk show this approach is applicable for both the problem and its context. The problem in this case is complex and multidisciplinary. The context, a PSO, is characterized by a high autonomy of the professionals and also by organizational structures that impede organizational learning and an efficient knowledge exchange. The method chunk especially focuses on both phenomena and together with the involved actors realizes a unique combination of simulation in collaborative design with a high amount of participation.

In a series of six iterative workshops, some of which done in multiple iterations, we invited a heterogeneous group of healthcare management, specialists, doctor assistants and nurses to collaboratively inspect the simulation, put it to the test, and assess its implications. The findings are based on observations and interviews at the start, during and after the ADR team worked on the development in this organization. At every session, workshop and interview observational notes were carefully made, and processed afterwards. Based on these observations six major contributions of the application of the developed method chunk in this organization are identified:

1. Enhancement and validity of the model. From the beginning the model is built around our perception of the hospital's reality. After interviewing and collaborative reflection our model is adjusted and in some case substantially due to the differences between how practitioners deploy their tasks and also because of differences between specialties or departments within the hospital.
2. Increase in acceptance, insights and awareness of the medical staff. The medical staff in the beginning took a rather critical perspective versus the simulation method itself, but also against the implementation and change project plan of the hospital's board itself. Gradually this critical attitude changed. Also due to the exemplary behavior and attendance of top management during the workshops, but especially because of the well founded arguments given by the simulation model and the results. At the end the participants reported to have learned substantially on operational processes, planning and use of resources in the new situation.
3. New social interactions and collaborations are initiated. The reorganization in the hospital requires a lot of medical staff to start new collaborations. Doctor assistants for example in the new situation will be organized over multiple specialties. This has a great impact on their social and working environment. An important side effect that was not originally our intention is the initiation of new social interactions through the collaborative workshops. The simulation model and its outcomes gave the participants a common theme for discussion and a start to initiate their first collaborations. A majority of the medical staff reports this to be of a major contributor the success of the project.
4. Rephrasing the original *problem* into a constructive setting. The original problem; how to cope with limited available space, the consequential changes in processes and workflow, is rephrased in a setting on how to make most effective and efficient use of shared resources and improve interdepartmental alignment. The need for interdepartmental

collaboration is one of the shared visions on the future situation.

5. Increase knowledge about the patient treatment process chain. As in many healthcare organizations, the hospital staff in our research had very little knowledge about the overall process chain their patients are involved in. During the intensive workshop discussions, insights in the process chain were naturally created. By providing more knowledge about the patient treatment process, the healthcare professionals are better equipped to make last-minute decisions about changes in the patient scheduling.
6. Rethinking organically grown processes. As discussed in the introduction, many processes are not necessarily designed but are created in a natural manner. Just because a certain procedure seems to be working and performing sufficient, there was never felt any reason to redesign a process. When the major changes in the organization were announced, many healthcare professionals reacted very hesitant. They never realized the full potential of the processes they were working on, but were definitely not motivated to improve on any. An often recorded quote during the research project is: *this is the way we work here*. The introduction of the simulation model that was built collaboratively with the medical and administrative staff created a constructive setting. With this introduction, processes that were always assumed to be invariable suddenly were topic of discussion. Many processes have been redesigned using the simulation as a reference model to analyze the feasibility.

The participants report that especially the combination of simulation and collaborative design two methods is fruitful. The simulation offers the participants an accessible way to discuss and try out design alternatives during which an efficient knowledge exchange process develops between these professionals. Therefore it can also be concluded that the application of the IT-artifact in the context of a PSO is a viable approach, because during the process the simulation model incrementally increases in detail and accuracy and

also its acceptance. The combination of the two techniques in the method fragment finally leads to viable redesigns of operational processes and a new physical hospital layout. This redesign is a proper foundation for the next stage of the ISD process.

### ***Evaluation of the IT artifact***

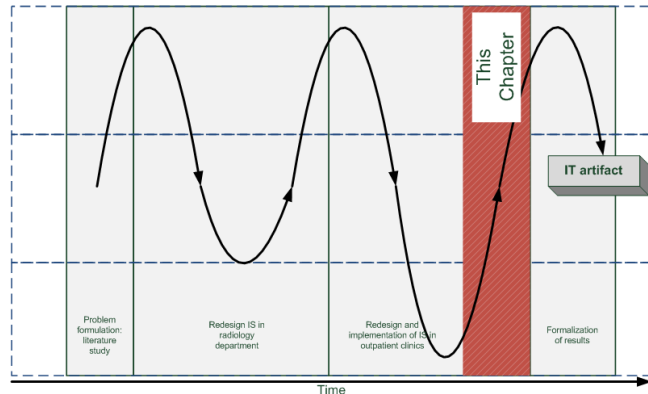
Since this is the second and last iteration in the development of the IT-artifact (a situational method to develop method chunks for information system development in professional organizations) the evaluation of the IT-artifact itself is conducted in the next chapter; chapter 6. In this chapter the reflections on the development of the initial design of the artifact are discussed, which results in a final design of the method under-development.





## 6 Reflection and learning

This chapter reflects on the design of the development method for IS for professional service organizations. In chapter 1 the problem statement of



this research is formulated as the aim for the situational development of information systems for professional service organizations. As is discussed in chapter 1, IS in service organizations does not produce the desired outcomes because of misfit in the development and implementation approach taken for this particular organizational domain. The mainstream approaches to develop IS are briefly discussed, upon which can be concluded that there is no *silver bullet* that addresses the particular requirements for information systems development for PSOs. In chapter 3 an extensive literature review that discusses the various paradigms, approaches, and methodologies for information systems development and the key characteristics for professional service organizations reveals an appropriate solution direction for this class of problems: situational method engineering. After a literature review on SME, an initial design to develop IS for PSO *the IT artifact* is produced. Since overarching methodologies that address every aspect of the information systems development process are found to be inappropriate to be applied in professional service organizations, the initial design aims to develop method chunks that address very specific areas of the ISD process. The initial design of the IT artifact consists of four sequential stages that cover respectively the identification of contingency variables, the definition of the method

chunks scope, the selection and assembly of the method chunk itself, and the implementation of the method chunk. This design, alongside a set of design principles is applied and further enhanced in the two BI&E iterations in chapter 4 and 5. This chapter reflects upon the development process in order to identify what can be learned from it on a conceptual level. Action design research has two ends; (1) the development of a solution for an organizational specific problem; and (2) the development of a solution for a class of problems the organizational specific problem is an instance of. In this research the problem instances are the problems in the ISD process in two distinct professional service organizations, covered in chapter 4 and 5. The class of problems is *situational development of information systems for professional service*. The IS development iterations in chapter 4 and 5 reveal insights that aid in the understanding of this research's problem statement as a class of problems. In this chapter the individual BI&E stages are reflected upon to put them into the perspective of the general research objective.

The method applied in this stage is depicted in Table 6-1.

Method	Goal
Observational evaluation: workshop sessions	(a) Get feedback on the applicability and effects of IT artifact in the problem domain. (b) Improve the IT artifact under-development
Technology rules	A means to utilize scientific research in design science research areas and are to be used by professionals in the field

**Table 6-1: Applied methods in the reflection and learning stage.**

As is discussed in chapter 2, in action design research, reflection and learning occurs in parallel with problem formulation and building, intervention and evaluation. The synergy of the combination of the three activities is what separates ADR from other design research studies, and what is most valuable to ADR. During this stage the research process is adjusted to resemble improved understanding of the domain specific problem and the IT artifact that provides the solution. Furthermore, the contributions to the body of knowledge are identified. This stage draws on the principle of *guided emergence*.

## **Principle 6: Guided emergence**

Emergence is typically the result of a random act of creativity; it is hard to plan, structure, and enforce (Gero, 1996). This principle implies there is a delicate balance between the initial structured design, and the ongoing shaping by the organizational use (Sein et al., 2010). In this stage the constant interaction between the researchers, the practitioners and end-users is an important factor, not only in addressing the problem instances, but also defining the solution to the class of problems the problem instances are part of. To gain insights in the fit between IT artifact and the organizational domain we use workshops as an observation evaluation method.

### ***6.1.1 Method: Observational evaluation: workshop sessions***

The situational development of IS for professional service organization; the research problem in this research, is addressed in two BI&E iterations, starting with the analysis of the problem domain, to eventual implementation and testing. At the ending of each cycle we aim to discuss the developed artifact with the involved practitioners and end-users. In this research workshop sessions are organized in which the use of the artifact will be discussed. Using the open setting of workshop sessions, we motivate the organizational participants to provide exhaustive feedback on the development process.

### ***6.1.2 Method: Technology rules***

Sein et al recommend for ADR research to move towards generalization by either re-connecting to source theories, articulating design principles, or generating new, prescriptive theories (Sein et al., 2010). Since it is very problematic to induct new theories from a single ADR process (Lee and Baskerville, 2003; Yin, 2009), the technological rules approach is taken for this research (Aken, 2004). Technological rules are a means to utilize scientific research in design science research areas and are to be used by professionals in the field. The technological rules approach answers to the five key users-need of practitioners regarding academic management theory: descriptive

relevance; goal relevance; operational validity; non-obviousness; and timeliness (Thomas and Tymon Jr, 1982).

## 6.2 Factsheet

In line with previous chapters, in this paragraph the input that is used and output that is delivered is discussed. In Figure 6-1 the snapshot of the process-data model is depicted. Below the figure the inputs and outputs are discussed.

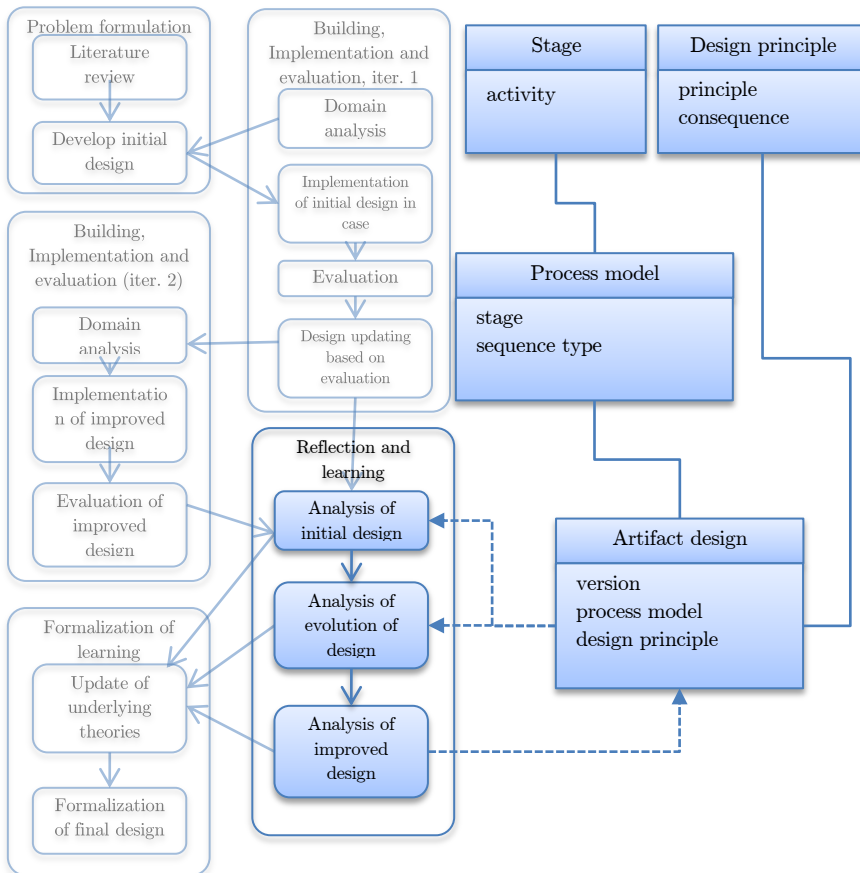


Figure 6-1: Process-data model: Snapshot Reflection and learning stage.

Input: Initial design:

Type	Description
Process model	The process model of the initial design is sequentially structured
Stages	There are four stages in the initial design
Design principles	Based on the literature review a set of design principles is formulated

Input: Beta-version of design:

Type	Description
Process model	The process model is upgraded to a reciprocally structured process; stages are operationalized reciprocally, by which parallel execution of activities in different stages can occur.
Stages	In the beta-version four stages remain the same as in the initial design
Design principles	The design principles are not updated based on the first BI&E stage

Output: final version of IT artifact:

Type	Description
Process model	The process model remains stable after the first BI&E stage
Stages	In the final version four stages remain the same as in the beta-design
Design principles	The design principles are updated after in accordance the evaluation of the previous BI&E stages

### 6.3 Initial design

Based on a literature review, in chapter 3 an initial approach is designed to deal with the formulated research statement. In this paragraph this initial design is analyzed.

The initial design to approach IS development in professional service organizations is based on the situational method engineering paradigm. The initial approach that is applied and further enhanced in the case studies in chapter 4 and 5, is aimed at developing method chunks that fill a void in traditional ISD processes that are often applied in professional service organizations. By developing method chunks that address specific needs in the ISD process, the outcome of this process is intended to be more successful; thereby facilitating the PSO under-review with a better fit IS that supports the business goals.

The initial design consists of a process description, covering four distinct, sequential stages, and three design principles. Both the process and the design principles are based on the ISD / SME approaches as found in literature, tailored to the characteristics of professional service organizations.

The proposed process in chapter 3 consists of the stages: identify contingency variables; define the method chunk's scope; select and assemble the method chunk, and implement the resulting chunk in the organization. On a finer level of detail, the identification of the contingency variables consists of two activities: the characterization of the ISD process as proposed, and the characterization of the organization the ISD process is executed in. The definition of the method chunk's scope consists of the definition of the process, and the definition of the products the chunk is supposed to produce and aid the ISD process. In the selection and assembly of the chunk, method fragments are selected to form a chunk (at least two; one product oriented, one process oriented), the individual method fragments are configured, and finally assembled into one method chunk. The final stage provides guidance on how to use the newly

developed method chunk, and actually implements the chunk in the organization. The process is designed to be executed sequentially.

The set of design principles consists of three principles: involve end-users in the development; define tasks and parts of the IS development process upfront; and integrate different interests into the process.

## **6.4 Evolution of initial design**

Both the initially designed process and the design principles are used as starting point for the organizational cases the ideas are implemented in. During the course of the two organizational studies the initial approach to develop method chunks that address PSO specific needs in the ISD process, the approach is iteratively altered and further enhanced based on insights the applications of the approach granted the researchers. This is in line with the ideas from action design research in which the design of the artifact is developed during the actual development of the artifact itself. Eventually the iterative design process leads to the improved design to develop IS with the means of method chunks, as presented in paragraph 6.5. In this paragraph the evolution of the initial design is reflected upon. The reflection is based on the experiences as part of the design activities in the two professional service organizations. The motivations to alter and improve the approach to develop IS in PSOs is the main topic of this paragraph. The aspects of the *IT-artifact* development process that are discussed in this paragraph are the layer of abstraction, the structuredness of the approach taken, the time perspective on the development process, the matching process between the ISD paradigm and the organizational effectiveness paradigm, and the agility of the approach.

### ***6.4.1 Layer of abstraction***

One finding during the development of the method chunks in the case studies is the relationship between the end-users' capacities to interpret research findings and the effects on their daily practices,

and the abstraction level of the design sessions. In Table 6-2 a two-by-two matrix depicts this relationship.

		Capacities of the end-users	
		Low	High
Abstraction level of the design sessions	Low	Productive session, little innovation, audience, results content maximum	Productive session, little innovation, audience not content, sub-optimal results
	High	Non-productive session, little innovation, audience not content, sub-optimal results	Productive session, much innovation, audience, maximum results

**Table 6-2: Matching abstraction level to audience capabilities.**

The capacities of the end-users should match the level of the design sessions in ensure the progression during these sessions. If the capacities of the end-user groups are high, but the abstraction level of the design sessions is low, the output is lower than expected; if the capacities of the end-user are low, but the sessions' abstraction level is high, than the output is also mediocre.

#### **6.4.2 Structuredness of the method**

The action design research team in both BI&E iterations consists of researchers and professionals, as is discussed in chapter 2. Although the process for constructing a method chunk that is the main focus of in the case studies is rather straightforward, the ADR approach that is the foundation of the development of the IT-artifact, enabled an *engineering* approach to the construction of these method chunks. This engineering approach implies a step-by-step, fuzzy process in which every direction can be explored and wrong choices easily be restored by selecting a different direction. The development of the *goal modeling – collaborative design* method chunk and the



*simulation – collaborative design* chunk follows this engineering oriented process.

Although the process for developing a successful IT development approach, as described in the initial design, is quite structured, the operationalization in the first BI&E cycle is rather fuzzy. By involving practitioners in the development process, unforeseen feedback is received immediately. Issues are raised such as the delay caused by the need for initial analysis of the ISD process and the lacking skills by the professionals to read process diagrams. Because of the ADR approach taken in this research, the researchers in the research team have to react ad-hoc on this feedback. Therefore, in the first BI&E cycle the original structured approach, as developed in the initial design, is relaxed. This results in a harder to manage process, which is not beneficial to the project progress. For the second BI&E cycle, executed in the second organization, the structured approach that is designed for the initial design is updated with a more agile flavor. The researchers recognize that a sequentially structured approach does not suit an ISD process in which goals are ill-defined, and desired outcomes vary over time. The flow of the updated development process can be depicted in four stages in which tasks and actions reciprocally interact:

*Problem domain analysis:* The professional service organization cases in which the method chunks are developed represent characteristic PSO environments; complex processes, professional autonomy, IS used to coordinate processes, and an organizational model that aims for human resource development. The most complex factor in both cases is that solution direction for the ISD process is not defined at the start of the research project. Much effort spent on the analysis of processes, workflows, organization structure, and requirements analysis for the information system.

*Scope setting:* Since the management in both cases has not yet defined the development direction at the start of the research project, many alternatives are available to support the ISD process. One of which is the analysis of the organizational structure, staff allocation, and tasks redesign. However, after an extensive period of

interviewing it becomes clear that processes and IS are not aligned because of the staff's lacking awareness of processes in their own departments let alone adjacent departments. It becomes clear that, before moving to the optimization stage, first awareness has to be created about the current situation. Since the professionals that employ this PSO are not trained to use modeling languages such as UML or BPML, other alternative have to be found. Visualization by means of simulation turns out to be a technique that is understandable for every process-layman. The solution direction is therewith decided: a method chunk that supports the ISD process in the analysis, optimization, and implementation of processes and the IS support.

*Method chunk selection and assembly:* The development of the method chunks passes through several stages before being finalized. Initially general designs are created that can be understood by every stakeholder. The more the IS development process advances the more specific the method chunks become. For example, in the simulation – collaborative design case, the model is too generic and participants in presentations and workshops request more detailed simulations. However, the more detailed a process is simulations, the more complex the overall behavior of the simulation becomes. Therefore it is decided to zoom in on particular problematic areas and time period such as the entrance and registration area, processes that are located in crowded areas, and *rush hour*. After several stages in which the simulation model is narrowed down and further detailed, results a very specific simulation of the processes of one outpatient department at a very busy day. By visualizing current and future processes that are aligned with future IS, participants in workshops and presentations can identify problems and opportunities for further improvement. Results from a detailed simulation of every department are distributed to every stakeholder. Based on these simulations collaborative design sessions are very productive.

*Implementation:* After the method chunks are developed, they are implemented in the organizations. Because of the IT-dominant approach taken to action design research, the first BI&E iteration

(the goal modeling – collaborative design method chunk) is not fully implemented in the organization. The second BI&E iteration, in which the simulation – collaborative design method chunk is developed is fully implemented in the organization. In this BI&E iteration, the plan for the collaborative design sessions is continuously improved. Initially the sessions are oriented toward direct improvement. During the sessions a very structured approach is taken to identify key performance indicators, future developments, interactions between departments. Based on this analysis during the sessions an improved design is constructed. During initial sessions a lack of awareness is recognized. This lacking awareness creates fertile ground for end-user resistance to the ISD project. Because the stakeholders' lack of awareness is recognized during initial sessions, the design of the sessions is adapted towards more informative presentations which are followed by workshop sessions. During the research project end-users regularly visit presentations and workshop sessions. In later sessions the awareness is much more improved and resistance towards the ISD process is changed in a more collaborative atmosphere.

### ***6.4.3 End-user involvement***

In accordance to the ADR approach, during the research in the simulation – collaborative design project the development process of the method chunk is influenced by the end-users to a significant extent. In the end, the method chunk needs to be tailor-fit to the organizational characteristics. Therefore the requirements of the method chunk reflect the end-users in this project are the administrative and medical staff of a hospital. Initially it becomes clear that awareness needs to be created about the upcoming organizational and corresponding IS changes and about the efficiency of the processes in the current situation. During the course of the method chunk development process the focus gradually shifts from configuring and analyzing future situations to guiding and directing the stakeholders towards the future situation. Naturally, just the presence of the ADR team that is analyzing the current situation by conducting interviews, discussing improvements influenced the

stakeholders' opinion towards the development process. In general there are several lessons the ADR team learned by observation during the development of the method chunk in close cooperation with the end-users:

The development process in a PSO is conducted most efficient bottom-up. In the early stages of the process it becomes clear that a top-down approach in which standards are predefined, plenary presentation sessions are used to share ideas, and process analysis is done without consultation of the end-users, is a fertile ground for a lot of resistance. The end-users in this PSO demand to contribute to both the method fragment development process as to the use of the fragment in the organization.

End-users demand a shared goal to work towards, as well as a vision about how to get there. The explication of the goal of the process and the process stages greatly enhance commitment of end-users to constructively participate in the development process. Without this shared vision end-users in this PSO tend to critically address the need of every stage in the process. By depicting a shared goal end-users proactively contribute in the collaborative design process.

The uniqueness (self-proclaimed or proven) of every department's processes, workflows, staff, and IS can be used as means to clarify a development process. In early experiences in this and other PSOs the uniqueness of a department is often used as an argument to not conform to general standards. In the application of the simulation – collaborative design method chunk the uniqueness of one outpatient department is used to explain that general standard can be applied to one unique department, and therefore to every department. The detailed modeling of the department is used by the ADR team to convince staff of the applicability of the new standards to the current processes.

Even in the collaborative design sessions, end-users prefer to have a valid design for their processes, workflows, and IS presented by the ADR team than to construct one themselves from scratch. Any valid

design can be used as a foundation for the collaborative design process. Especially during the course of the development project the demands for a more specific design increase. If end-users are left with some general thought to reflect upon, they experience that a very negative.

Collaborative design is not limited to plenary sessions only. Since the focus of the research project narrows down to the ophthalmology clinic the ADR team contact with this clinic intensifies. More interviews are conducted in the ophthalmology clinic, medical staff is actively involved in the analysis, modeling and optimization activities, and personal relationships improve. This all contributed to the attitude of this clinic being very constructive thinking respectively to other departments. Staff of this department considers the change process as an opportunity instead of a confronting issue.

#### ***6.4.4 Paradigm alignment***

Paradigm alignment refers in this research to the matching of the ISD paradigm as is dominant in an organization under-review and development of the method chunk. The ISD paradigms, described by Hirschheim vary from the functionalistic paradigm (information systems are designed to contribute to specific ends); the social relativist paradigm (system objectives emerge as part of the organizational construction of reality); the radical structuralist paradigm (on a social level the ends of the information system are open to multiple interpretations, but on an economic level there is a clear defined end); and the neo-humanist paradigm (information systems are the means to overcome barriers to rational discourse). The idea is that the development of the method chunk and the method chunk itself should align with the dominant perspective on ISD in an organization.

In this research there are two levels of evaluation in regards to the ISD paradigm matching process; the process to develop the method itself (*the IT-artifact*) and the developed method chunk as an individual unit. Obviously the method chunk itself should be aligned to the ISD paradigm, but also the alignment of the method to

develop a method fragment is a major determinant of the overall success. In this section we discriminate between these types of alignment.

*Process alignment:* During the first BI&E stage described in chapter 4 the process to develop the method chunk is set-up to be well-structured and therefore matching properly with the functionalistic ISD paradigm identified in this organization (although, as the research advances the structuredness is relaxed somewhat, as discussed in section 6.4.2). However, the research team foresees alignment issues when the method to develop method chunks is less suitable in an organization in which a different ISD paradigm is adopted than the functionalistic. After the evaluation of the first BI&E stage, the process to develop method chunks is therefore adapted to fit other ISD paradigms. In the second BI&E stage of this research, the upgraded process is used to develop the simulation – collaborative design method chunk. The updated process is based on agile principles, including reciprocity and iterative stages. Basically this approach to develop method chunks is much like the ADR approach in which the various stages are operationalized in parallel, allowing intermediate results from one stage directly affect activities and outcomes in other stages.

*Method chunk alignment:* In the first BI&E stage the developed method chunk is not in line with the ISD paradigm. The goal modeling – collaborative design assumes different stakeholders to have different perspectives on the added value of the IS in the organization, which does not align with the functionalistic ISD paradigm. But, because of the IT-dominant variant of ADR applied in this research, the first BI&E stage does not involve end-users. Therefore the method chunk is not implemented to a full extent in this organization, and the effectiveness of this method chunk itself cannot be evaluated properly. Apart from the lacking end-user evaluation of the first developed method chunk, the researchers assume in retrospective a low fit of the method chunk with the overall ISD approach taken in this organization. The second developed method chunk, the simulation – collaborative design

chunk, is developed in an organization in which the neo-humanist ISD paradigm is dominant. The method chunk itself can be characterized as being open for various perspectives from the involved stakeholders in the organization. Since this method chunk is developed in the second BI&E stage of the ADR approach, end-users are involved in the evaluation of the developed artifact. During the evaluation of this method chunk in chapter 5, it becomes evident that the alignment of the chunk with the dominant ISD paradigm is beneficial to the implementation of it. This can be concluded based on the increased acceptance and awareness of the stakeholders in the ISD process, the newly created interactions and collaborations and the redefinition of the original issues in the ISD process to a more constructive setting.

#### ***6.4.5 Agility techniques***

In chapter's 3 literature review, the evolution of ISD methodology describes the transformation of structured, sequential, traditional methodologies to ill-structured, ad-hoc, reciprocal (*agile*) methodologies. The *backlash against methodologies* resulting from the over-engineered traditional methodologies paved the way for the less structured, less formal methodologies that cover not necessarily every aspect of IS development. However, the initial design of the approach to develop method chunks resembles more the traditional, sequentially stages methodologies for ISD, than the agile ones. Since this was just the initial design, it is upgraded to a more agile approach during the two building-evaluation-and-implementation stages. The resulting approach includes activities that are executed reciprocally and iteratively. But herewith is a major share of process complexity introduced. During the course of the two case studies in the two selected organizations clear contributions of method chunk under-development are not described. Therefore the outcomes are less restricted to fit particular needs. Because of the lower expectations, process inefficiency caused by the agile approach is not necessarily a major issue. However in retrospection the process efficiency can be dramatically improved by introducing techniques from the agile methodology tool set. Reflecting on the original

research goal – *to develop a situation approach for ISD for professional service organizations* – the approach developed in this research can be much improved by introducing techniques like time-boxing, brainstorm sessions, and regular review sessions.

#### **6.4.6 Regular feedback loops**

In chapter 3 three design principles are presented that are set as guidelines in the development of method chunks. The first principle stresses the need to involve end-users in the development of chunks, because method chunks cannot be developed from a single perspective (like the expert perspective), development of IS is more successful if end-users are co-creators of the methodology to develop the IS, and interaction with the end-users provides the means to check the design against the original and updated requirement. In section 6.4.3 implications and improvements about involving end-users in this research is discussed to a larger extent.

In this section another aspects of end-user involvement is discussed; the need for structural and regular feedback loops. Although this research is set-up to be action design research, thus implying the firm involvement of practitioners and end-users, predominantly the regularity of involving end-users in *developing* the method chunks turned out to be omitted. During the first BI&E stage end-users are not involved because of the IT-dominant perspective taken on ADR in this research. Therefore during the second BI&E stage in this research no mile-stones or other regular feedback loops were set to discuss the design. *Nota bene*; regular sessions were set for discussing the outcomes of the designing process, but the evaluation of the developing activity itself was not regularly scheduled.

Reflecting upon the lacking regularity of involvement of end-users in the developing activities, leads to the conclusion that this has been fertile ground for end-user resistance. During the course of the second BI&E stage, the feedback session were not regularly schedule, although emphasis during feedback sessions was put on the development rather that the results from the developing activities. Based on this experience the researchers adjust the first design



principle to involve end-users on regular intervals during developing activities.

#### ***6.4.7 Adaptive task definition***

The second design principle to develop method chunks that is identified as a result of the literature review is the need to define tasks upfront. Task definition is one of the major issues affecting productivity of professionals that are employed in a PSO. Because this research aims to develop IS for the PSO domain, and in chapter 3 IS is identified to support communication and coordination in the complexity of professional organizations, the definition of the aim and task as a part of the overall ISD process is defined as a design principle.

However, requirements upon which tasks are based change over the course of an ISD project, as became evident during the two BI&E stages that are part of this research. Especially the second BI&E stage in which the simulation – collaborative design method chunk is developed. Since the main contribution for the method chunk under-development in this case study was ill-defined at the initiation of the project, tasks and activities could not be clearly defined upfront. In retrospective, the unclear task definition is one of the major aspects for overall delay in this project. Naturally there is a payoff between the action design research approach, in which the requirements for the artifact under-construction are developed parallel to the artifact itself, and the delay caused by lack of defined tasks that form the core of a strict planning. The more freedom in the ADR approach can provide fertile ground for innovative solutions, but a drawback is the increased risk for underperformance.

During the development of the two method chunks and in parallel the method to develop method chunks, the researchers concluded that adaptive task definition provides the best means to cope with insecurity on the one hand, and enforcing progress by means of a planning on the other hand. Adaptive task definition implies that tasks are defined at every point in time in a project, although the

tasks might not be the right ones in the end. As soon as a better course of action is identified, the tasks are redefined.

#### ***6.4.8 Multi-perspective integration***

Very typical of professional service organizations is the importance of the interests of every stakeholder in a process. The professional autonomy and unique knowledge every stakeholder has in a specific process make a single perspective view on a process (and the modifications of this process) impossible. Therefore during the initial design of the approach to develop method chunks, the integration of interests is defined as a design principle.

The importance of this design principle became evident in both development processes in both organizations. In the first BI&E stage, during the development of the goal modeling – collaborative design method chunk. In this organization the development of an IS for one specific department is influenced not only by the interests of the various stakeholders within the department, like the radiologists, nurses, and administrative staff, but also by other departments adjacent to the radiology department. Because of the *client-server* relation the adjacent departments have with the radiology department, these departments have a major interest in the outline of the IS of the radiology department. For instance the means for internal / external scheduling of the activities the radiology staff commences, or the means to communicate results to other departments. In the second BI&E stage every outpatient clinic is involved in the redesign of the IS supporting their processes. Since next to the IS, the processes and the staffing are part of the redesign of the organization. Therefore every staff member's interest is needed to be included in the redesign process. In this case study the researchers found a broad array of interests, often conflicting. For instance one workshop session in this project dealt with the definition of efficiency and its implication for an outpatient clinic. The multiple stakeholder groups that were present in this workshop all held different definitions of efficiency. But also within a group, for instance the surgeons, several definitions of efficiency were put

forward. Efficiency ranged from throughput, waiting times, patient satisfaction, queue length, operation room occupation, occupation rate in general, cost effectiveness, staff occupation, staff per patient, patients per hour, and many more. This imposes serious effects for the development of the method chunk with which (part of) the IS is developed.

For the development of the method chunk this implies that the integration of these perspectives on the redesign process are major determinants for the overall applicability of the method chunk. In practice this means that in the second BI&E stage simulation is selected as a method fragment to identify (lack of) efficiency, predominantly the efficiency and effectiveness of the waiting areas. Next to this collaborative design is chosen as a method fragment to develop a future design of IS, processes and staffing. Although collaborative design is not the most efficient means to design new IS, in this organization it resulted to be one of the few means to deal with the broad array of individual interests.

## **6.5 Improved design**

The goal set for this research is to develop a situational approach to develop ISD for the PSO domain. In chapter 3 an approach is presented that forms a foundation for a situational approach to develop ISD specific for PSOs. A literature review on information systems development, professional service organizations, and situational method engineering provides input for an initial design to develop information systems in PSOs. This approach's initial design as described in chapter 3 and paragraph 6.3 is iteratively upgraded in the BI&E stages in chapter 4 and 5. The initial approach is based on situational method engineering, with which method chunks are introduced as building components for an overall ISD methodology. Initially the development of method chunks based on the SME approach appears a straightforward process. But the iterative application of the approach in two specific organizations that can be characterized as PSOs, feedback is received on the aspects of the approach that needed to be improved. In paragraph 6.4 the evolution

of the approach is described. In the paragraph we specifically reflect on the layer of abstraction, structuredness of the approach, involvement of end-users, ISD paradigm alignment, agility techniques, regular feedback loops, adaptive task definition, and multi-perspective integration.

Aspects of evolution of the initial design      Effects in the updated design

Layer of abstraction	Adaptive abstraction level during design sessions with practitioners and end-users. The more advanced the participants in the design sessions, the more results are gained with a higher abstraction layer
Structuredness	The extent to which the method to engineer method chunks is structured determines the manageability of the development process. If the process is too structured and rigid, unforeseen feedback from either the stakeholders or influences from the environment can result in a hard to manage process.
End-user involvement	In professional service organizations, end-users are domain experts. They have a high level of professional autonomy. It is therefore productive to involve end-users from the very beginning in the design and development process (and not apply the expert approach as a research)
ISD paradigm alignment	Both the process to develop method chunks as the process of the developed method chunk should align with the identified ISD paradigm in the organization under review

Agility techniques	Agile techniques like time-boxing, brainstorming and regular review sessions are very beneficial to the outcome of the method chunk development process, especially during the definition of the method chunk's scope and development of its process and products
Regular feedback loops	The process of the method to develop method chunks is over the course of two BI&E stages updated to a more agile oriented process, including reciprocal, mutual influencing activities, and regular feedback loops. This aspect is also included as a design principle
Adaptive task definition	Because of the dynamic nature of professional service organizations, tasks and activities are regularly updated. Therefore the definition of tasks should be adaptively. This aspect is also included as a design principle
Multi-perspective integration	Professional service organizations can be characterized by their high level of conflicting goals. Since stakeholders in an IS development process in a PSO can have opposing goals, it is important to include these perspective in the development process. This aspect is also included as a design principle

**Table 6-3: Effects of evolutionary aspects of the initial design on the updated design.**

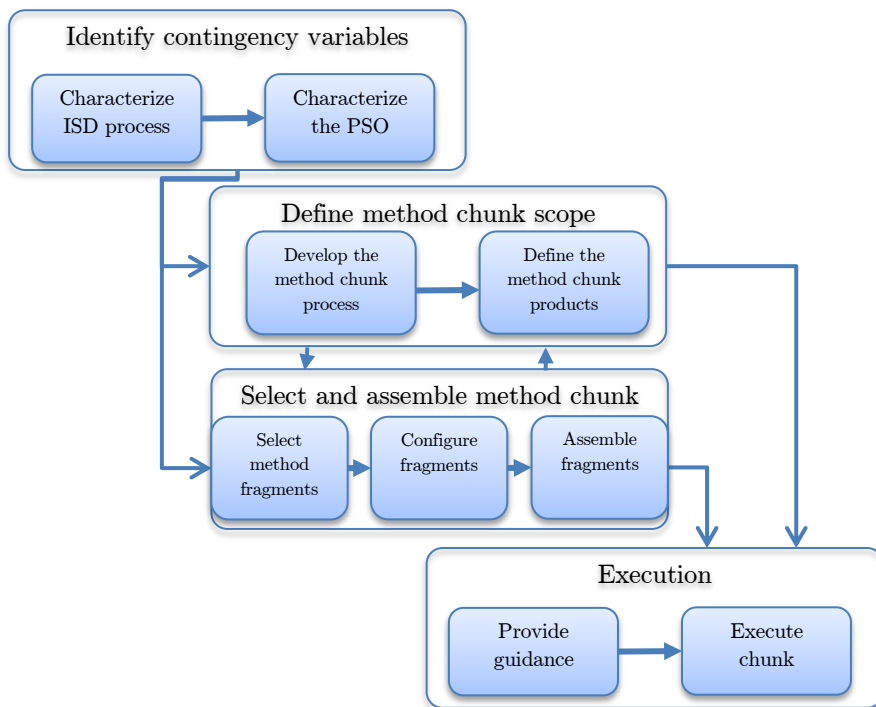
In this paragraph the final design for the situational approach to develop information systems specific for the professional service organizations domain. As is concluded in paragraph 6.4, is the design

still not perfect, and can be improved on several aspects. The approach that is presented in this paragraph consists of a process with specific activities and a set of design principles.

In this paragraph the outcomes of the two BI&E stages are discussed in light of the original problem statement. The outcomes so far consist of a process description to develop method chunks, a description of the products of the approach to develop method chunks, and a set of upgraded design principles.

### ***6.5.1 Method chunk development process***

The development of the first method chunk is done in a structured organizational setting. The second development process is situated in an organization in which the IS development is less structured. The resulting method chunk's process is therefore also unstructured. In this second BI&E stage, the early recognition of environment variables, the lacking elements in the overall ISDM and the contributions of the method chunk under-development to this ISDM are clearly identifiable. Based on this analysis, the outcomes of the to-be-developed method chunk are selectable straightforward too. During the development process in this case, the connectedness of the selection of the fragments supporting the method chunk and the definition of the method chunk's process became evident. The IT artifact's process is therefore far from sequential; every stage of the artifact is connected to other stages in the approach. In ISDM classifications this approach is termed *agile*, as is discussed in chapter 3. The initial design of the approach to develop situational IS for professional service organizations, as presented in chapter 3, is therefore updated in accordance to the insights of the two BI&E iterations. The updated process flow is depicted in Figure 6-2.



**Figure 6-2: Method chunk development process model: update.**

This section describes the process to construct method fragments. The process consists of four stages: (1) identify the contingency variables; (2) define the method chunk’s scope; (3) select and assemble the method fragments / chunk; (4) implementation. As is depicted in Figure 6-2, the stages are operationalized reciprocally, in an *agile*-based approach.

**Identify the contingency variables**

As discussed in the in both BIE iterations in this research, it is very useful to identify and define upfront: (1) the stage and the current ISD approach taken in the organization in which the method chunk is required. Since many ISD processes are not anymore sequentially organized as traditional ISDMs (see chapter 3) the *stage* of the ISD process can refer to either the phase in the process (e.g. requirements analysis phase in the software development lifecycle- SDLC) or the

intention of the stage in the process (e.g. the requirements discipline in the rational unified process -RUP). Based on this characterization, the method chunk's contribution to the ISD process can be defined. A clear distinction can be made between chunks that aid by developing a product (e.g. document) or aid in a process (e.g. change management). This distinction is discussed in literature in chapter 3, see (Ralyté et al., 2003). Furthermore the voids in the current ISD process can be defined. The gaps in the current process are the main focus for the development of a method chunk to be filled. Void can be either identified regarding products or processes. Next to the characterization of the ISD process, it is very useful to characterize the PS organization in accordance to Leavitt's diamond model for organizations. The organization can be characterized on four areas; information technology, tasks, people and structure.

### **Define the method chunk's scope**

In correspondence to the identified contingency variables in the previous step of the process, in this stage of the method chunk's scope is defined. This stage in the process consists of two elements; the development of the chunk's process, and the development of the chunk's products. Since a method chunk can be process oriented or product oriented (see chapter 3) and a method chunk consists at least of a product oriented and a process oriented method fragment. When the products of the IT chunk are determined, the process of the method can be defined. Important variables in the process are the beginning and end-state, the number and roles of the participant, and the extent to which the process is required to be structured. When these variables are determined the process can be defined in a number of stages, activities, and relationships. UML diagrams can be used to draw the process of the method chunk.

### **Select and assembly of the method fragments into a chunk**

During the process interaction with the above described stages in the method chunk development process, method fragments are to be selected and assembled into a method chunk. The environmental



variables, products, and process of the method chunk set the stage for the selection of method fragments that fits the scope. At least a process oriented method fragment and a product oriented method fragment are required to be assembled in a method chunk. A selection can be made from a broad array of fragments available, as long as the fragment is in line with the scope set for the method chunk. After the method fragments are selected, they are configured to fit the organizational specific need in the IS development process. When at least one suitable process oriented method fragment and one product oriented method fragment are selected, the method fragments are assembled into the method chunk under-development. In the development processes in this research, the product oriented method fragment turned out to be supporting the process oriented method fragment, e.g. the simulation model supported the collaborative design sessions.

### **Implementation**

When the chunk's products, process, and techniques are identified, a support document is produced in order to provide guidance on how to use the chunk in the context of the ISD project. The guidance should address every stakeholder in the ISD process. After this stage, the method chunk is ready to be implemented in the overall ISD process.

#### ***6.5.2 Method chunk development products***

The approach to develop method chunks produces several products. This paragraph describes four products of the IT-artifact: (1) the method chunk description; (2) the method chunk product description; (3) the method chunk process description; and (4) the guidance on how to use the method chunk. The products can be assembled in an overview, such as is presented in Figure 6-3, where the simulation – collaborative design method chunk is described.

## Method chunk description

During the construction process of the method chunk several environmental variables are identified. These variables form the main content of the method chunk description: the stage of the ISD process in which the chunk is to be used, what is the contribution of the chunk to the ISD process, and what voids are filled in an ISD process? Next to these variables information about the input and output is part of the description.

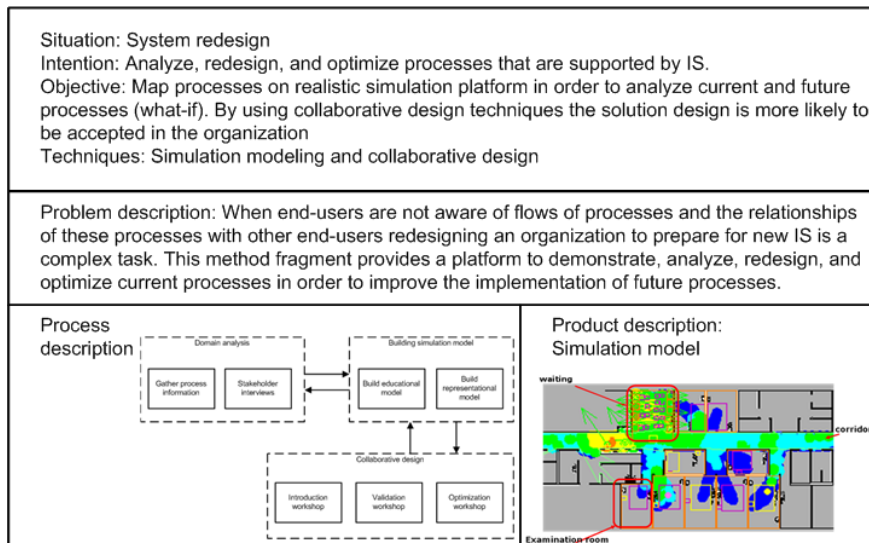


Figure 6-3: Method chunk description. Simulation - collaborative design example.

## Method chunk product description

The products of a method chunk can either be process oriented (e.g. a discussion method fragment) or product oriented (e.g. a stakeholder analysis). In both cases the method chunk delivers some products. The output and form of the products are to be described.

## Method chunk process description

The process of a method chunk can be structured or unstructured. In both cases the stages of the process and intentions of the process are

to be described. The description of the process should be accompanied with a set of rules on how to use and implement the chunk.

## **Guidance**

The guidance document provides an overview on how to use and implement the method chunk and adjective techniques. Basically the guidance provided with the chunk is a summary of the three descriptions of the IT-artifact's products that are listed above.

### ***6.5.3 Design principles***

A third outcome of this ADR research is the updated list of design principles. The principles for constructing method chunks are used as a basis that is expanded with experiences in this research. In chapter 3 three design principles are identified based on literature. The characterization of PSOs and ISD processes in chapter 3 provide useful principles for the initial design of the IT-artifact to anticipate on two different organizations in which method chunks are developed. After two BIE iterations the rules are updated and a fourth principle is added. Since this research is aimed at the development of chunks in a PSO environment, the rules in this research direct specifically to the deployment in the PSO domain.

Table 6-4 depicts the revised design principles for the approach to construct method chunks. The revisions are marked by italic font. The two BIE iterations revealed that end-users need to be involved in the development process by means of regular feedback loops till the chunk is implemented. Since tasks in PSOs are frequently changing due to the complex process flow, the principle to define the goal for the method chunk is revised in *adaptive* task definition. This results in that the aim for the method chunk under development is adapted when the underlying process is altered. The viewpoints of stakeholder groups in a PSO turn out to be divers. The integration of interests from the stakeholder groups therefore need to be done using a multi perspective approach in which interests are first isolated and afterwards integrated for the requirements of the

method chunk. A fourth design principle adds the use of tools and techniques developed in the agile ISD domain as useful support in the method chunk development process.

Design principle	Description
<i>End-user involvement with regular feedback loops during development activities</i>	The development process of method chunks in a PSO should be organized as a bi-directional process. Professionals should be involved in the development of a method chunk to create willingness to cooperate in the application of the method chunk in an ISD process.
<i>Adaptive definition</i> task	For successful developing method chunks with the IT artifact first the aim of the method chunk should be defined. Since processes are ill-defined in PSOs, the definition of tasks can be altered during execution of the task.
<i>Multi-perspective interest integration</i>	Various stakeholders in the ISD process and the business processes that are subject to the ISD process have very distinct interest in the IS and the ISD. It is important to capture this broad array of interests.
<i>Use agile techniques to guide the development process</i>	Tools and techniques from the agile information systems development approaches support the development of method chunks, by providing support for ill-structured processes, undefined tasks, and undefined outcomes.

Table 6-4: Revised design principles for a method to construct method chunks.

## 6.6 Applicability of ADR and SME

Both ADR and SME have not reached a plateau of maturity, as in main stream adoption. In this paragraph the reasons for the lack of adoption are discussed (and whether they are valid) in light of the results from this research.

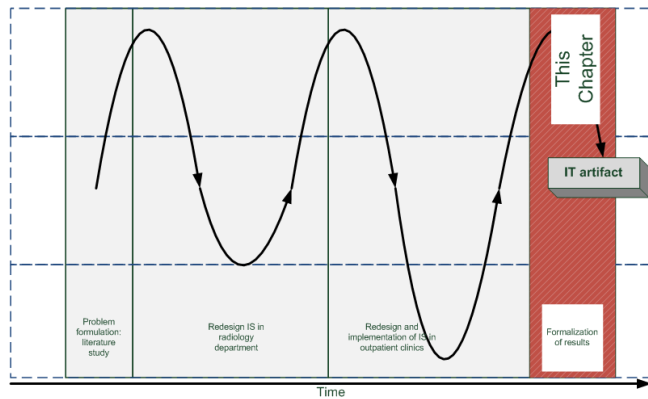
Action design research is, as discussed in paragraph 2.2 in chapter 2 a blend of design science and action research. It was first published in (Cole et al., 2005) and more awareness was created with (Sein et al., 2010). The main difference between design science and ADR is the interwoven nature of the design stages in ADR. In principle the problem formulation stage, building, intervention and evaluation stage, and the reflection stage are conducted reciprocally; the intermediate results in one stage influencing the processes in other stages. A major advantage of this ill-structured approach ADR is well suited to cope with ill-defined problems. However, at first sight, the ill-structuredness provide little grip for a manageable process. It appears the process of an ADR based research project is not very easy to replicate. This causes major concern in light of research rigor (Lee, 1999). Research described in this PhD thesis provides insight on whether these assumptions are valid. The first observation concerning ADR is that it is actually very valuable to approach a problem statement in an iterative approach. The definition of a problem and subsequent tasks is one of the hardest activities in every research. In this research the problem statement was ill defined, therefore the iterative approach aided the proper definition of it. Next to that the reciprocally structured stages provided rapid feedback loops on work in progress. For instance the development of the first method chunk started off with drawing of process models, which were not very well understood by the practitioners. The ADR approach opened up the possibility to change both the problem formulation for this ISD process, and the solution design attached to that. Therefore it was very easy to shift the development of the method chunk to the goal based approach, which is also part of the end result of this first development stage. A downside of applying ADR as the main research methodology is the means to structurally write down the research. The sequential flow of the chapters in this thesis does not corresponds to the temporal development of the research that is described in these chapters. In this thesis ADR itself is selected as a framework to structure the chapters, which turns out to be a valuable aid.

Situation method engineering on the other hand is also not a very well established method by a broad audience. There are several reasons that can be mentioned that cause the lack of adoption. In essence SME is a very rigid approach to ISD methodology development. In light of the granularity of the different variants of SME, as discussed in (Harmsen et al., 1994a), almost with mathematical precision chunks are identified to form the overall methodology. Similar, as for instance with UML, this rigid approach might repel the general audience. But foremost, standard ISD methodologies require initially little configuration. Application of SME requires all the configuration upfront, therefore making the initial investment in time and resources significantly larger than of-the-shelf methodologies. In later stadia this investments is naturally compensated because of a better fit with the organization. In this research SME is applied in a less rigid approach. Ideas from SME are used as foundation for the development of methodology elements that address very specific requirements in specific organizations: situation method chunks.

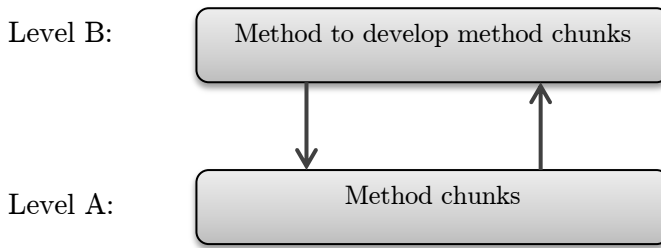
In this research the two approaches, ADR and SME are blend for the new approach to develop situational method chunks specifically aimed at organizations in the PSO domain. The ill-structured, reciprocal approach from ADR is combined with the targeting and filling of specific needs in overall ISD methodologies. In essence method chunks are developed for ill-defined problems for specific organizations that reside in the PSO domain. The combination of ADR and SME proves to provide synergy to attain this goal.

## 7 Formalization of learning

In this chapter the results from the research up to this point is formalized in order to provide a replicable approach to deal with information systems development – situational method chunk development in particular – in the professional service organizations domain. The key contribution of this research lies in the application of SME and method chunk development in this specific domain. Therefore, in this fourth stage of action design research, the solution to the specific organizational problem that is developed is perceived as an instance of a class of solutions that addresses a class of problems (Sein et al., 2010). In the reflection in chapter 6 a generic process for method chunk development in the PSO domain is discussed alongside a set of design principles that aid the development of method chunks. This chapter advances in this direction: based on the reflections on this research in chapter 6 a generic approach is presented to develop method chunks. From the reflections from chapter 6 contributions to underlying theories are distilled in the current stage of the research.



The developed method is a layered approach which consists of a method chunk layer (level A) and a method to develop method chunks layer (level B), see Figure 7-1. These two layers are discussed in this chapter in terms of the general solution to the class of problems that form the research statement of this chapter.



**Figure 7-1: Abstraction layers in this research.**

The problem with action design research is that the final artifact’s definition that will satisfy the initial problem statement is not yet seen before. Therefore, the new artifact will demonstrate a new domain for which the current knowledge base might not provide sufficient background. During the design and development of the artifact that eventually (and hopefully) satisfies the problem statement, new additions have to be made to the knowledge base as a sort of byproduct. New methods and measures that determine the designed artifact’s relative success can be part of these new additions to the knowledge base (Hevner et al., 2004).

As with every chapter in this research, the method(s) applied in this chapter are introduced first. In Table 7-1 we provide an overview of the research method we apply in this stage.

Method	Goal
Interpretivism	Our knowledge of reality is a social construction by human actors

**Table 7-1: Applied methods in the formalization and learning stage.**

**Principle 7: Generalized outcomes**

The goal of our research is to develop a situational approach to develop IS in the professional service organizations domain. The end of action design research is to find general outcomes based on specific problems. Therefore, the aim is to generalize the solutions that are developed for the particular problems in two building, intervention, and evaluation stages, that are discussed in chapter 4 and 5, to



generalized outcomes. In this chapter we reconnect this research to the underlying theories upon which the case studies are based in paragraph 7.3. In paragraph 7.4 the generalized approach is presented to develop situational ISD for the PSO domain.

### ***7.1.1 Method: Interpretivism to generalize***

One of the less developed sides of action design research is the means to deduct contributions to underlying theories based on the results of individual cases; that is: the formalization stage of each action design research. For this research a closer look is taken into interpretivism to gain better understanding for formalization of design science-based research.

Design science and action design research are paradigms to construct a solution for a problem instance, and by developing this solution provide a solution to a class of problems. The particular problem is an instance of this class of problems. The last stage of the ADR approach is to generalize from the solutions to the problem instances that are developed in the previous chapters. However, generalizing is a form of truth finding. In chapter 3 the design science paradigm, in which action design research approach resides, is discussed as an alternative paradigm to the behavioral, truth finding paradigms that are predominantly applied in IS research. As Hevner states; “the behavioral-science paradigm seeks to find what is true. In contrast, the design-science paradigm seeks to create what is effective” (Hevner et al., 2004).

There are several behavioral research paradigms for IS, however considerable controversy surrounds how the various research paradigms should be classified (Myers and Klein, 2011). Traditionally there are two main truth finding paradigms in IS research; positivism and interpretivism (Walsham, 1995), later on a third one is added: critical research (Richardson and Robinson, 2007). Applied in research positivism is by far the dominant approach in this classification scheme (Chen and Hirschheim, 2004; Dubé and Paré, 2003). Positivism refers to the belief that social-science research should emulate how research is done in the natural sciences – *the*

scientific method - in which knowledge is acquired through sense experiences and their logical and mathematical treatment (Lee, 1999). The interpretive paradigm adopts the position that our knowledge of reality is a social construction by human actors. In this view, value free data cannot be obtained, since the enquirer uses his or her preconceptions in order to guide the process of enquiry, and furthermore the researcher interacts with the human subjects of the enquiry, changing the perceptions of both parties (Walsham, 1995). According to Richardson critical research is defined more by being an ideologically oriented inquiry; the aim of critical research is to expose through critique of the illusions and contradictions of social existence with a view to enabling and encouraging social change (Richardson and Robinson, 2007).

Philosophical debates on how to conduct IS research have been the focus of much recent attention (Hevner et al., 2004). Generalizability in research that does not involve statistical, or sampling-based analysis (that is: critical research and interpretivism) is a major concern for the information systems field (Lee and Baskerville, 2003). As such the importance of relevance and rigor has been addressed over and over again (Benbasat and Zmud, 1999, 2003). For instance rigorous evaluation methods are extremely difficult to apply in design-science research (Hevner et al., 2004). For example, the use of a design artifact on a single project may not generalize to different environments (Markus et al., 2002).

In a paper focused on case study research, Yin offers three examples of generalizing from empirical to theoretical statements: generalizing from experimental findings to theory; generalizing from case study findings to theory; and generalizing from population characteristics to theory (Yin, 2009). Lee classifies many different options to generalize from the various research paradigms. According to Lee, there are four types of generalizing: the Cartesian product of: generalizing FROM Theoretical statements OR Empirical statements, TO Empirical statements OR Theoretical statements. ADR research classifies as ET generalizability (from Empirical statements TO Theoretical statements): generalizing from

description and observations to theory (Lee and Baskerville, 2003). This involves generalizing measurement, observation or other description to a theory. This type of generalization also applies to this research.

Although Yin's approach to generalizing, which is the foundation for Lee's generalization classification, is considered to be positivist, ET generalizability is considered to be in line with the interpretivist paradigm (Lee and Baskerville, 2003).

Lee recommends for ET generalizability several approaches; however these approaches are based on traditional case study research. An example is Eisenhardt's approach where the process that is described to generalize from empirical data and observations from case studies is based on the grounded theory approach (Glaser and Strauss, 2007). In this process first cases are selected, instruments are created to measure data, data is extracted, data is analyzed, hypotheses are shaped, and finally the hypotheses are cross checked with literature (Eisenhardt, 1989). As is demonstrated in this research, ADR approaches studies in the field differently. First the process is not sequentially structured and second artifacts are constructed based on insights from literature; theory is used upfront to aid the design process instead of afterwards to verify the findings.

Since generalizability is necessary for ADR to make the outcomes of the research accessible, Sein et al put forward a set of approaches that can contribute in this aim (Sein et al., 2010), by extending the grounded theory direction considered by Baskerville towards generalization by reconnecting to source theories, articulating design principles, or generating new, prescriptive design theories (Gregor, 2006; Lee and Baskerville, 2003; Sein et al., 2010). This direction is adopted in this research; in the next paragraph the findings in the development stages of this research are interpreted in light of the underlying theories. The aim is to formalize the learning in this research.

## 7.2 Factsheet

The final stage of the ADR process is in this chapter, like in previous chapters, accompanied with a process-data model, presented in Figure 7-2. Below the figure the output is discussed.

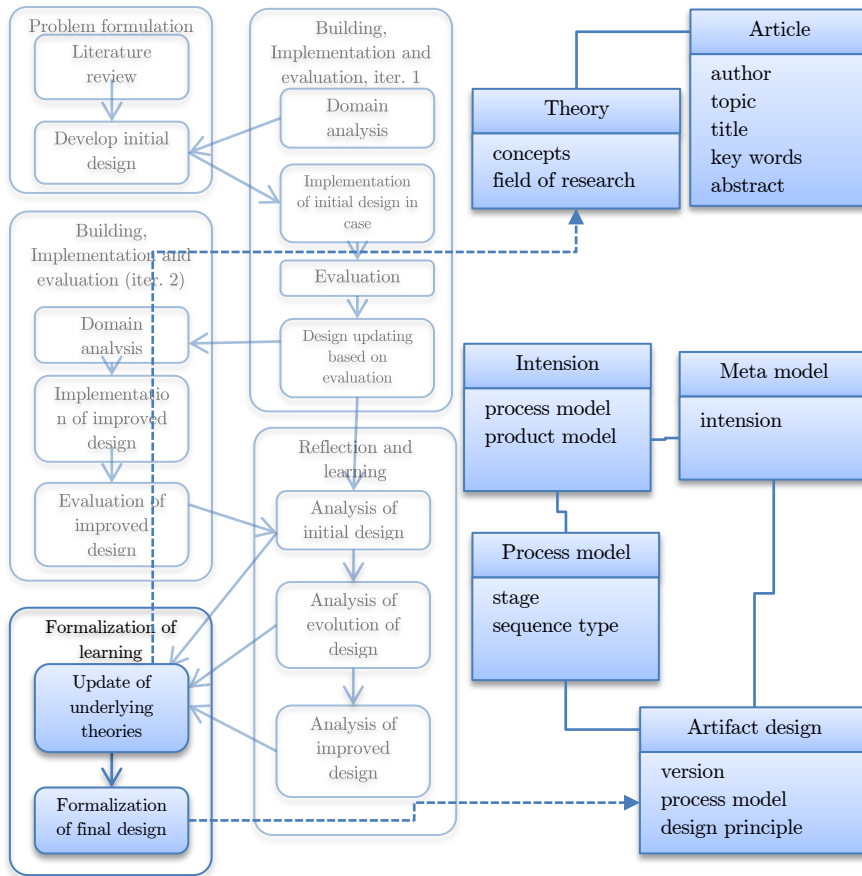


Figure 7-2: Process-data model: Snapshot Formalization of learning.

Output: Reconnect to the underlying theories:

Type	Description
Theory	Several theories are applied as a foundation for this research. In this stage of the ARD process, insights are used to reassess the underlying theories to analyze if the theories need an update.

Output: Final artifact design:

Type	Description
Artifact design	The final artifact design is presented in this chapter
Process model	Part of the final artifact design is the underlying process model; this model is discussed in chapter 6
Meta model	Based on the initial design and building stages a meta-model is constructed that formalizes the IT-artifact
Intension	Main concept in the meta-model is <i>intension</i> which combines the various ISD environmental aspects

### 7.3 Reconnect to the underlying theories

In chapter 3 an extensive overview is presented that covers the underlying research and theories for the aim of this research to develop a situation approach for ISD in professional service organizations. In this paragraph, as part of the formalization process, the findings in the two BI&E stages in chapter 4 and 5, and the reflection on the BI&E stages in chapter 6 are reconnected to the original underlying theories. The aim is to identify any contributions to the body of knowledge based on the action design research conducted in this field. In chapter 3 three major literature themes are covered; information systems development, professional service organizations, and situational method engineering. The contributions to themes are discussed in the sections of this paragraph.

### 7.3.1 ISD methodologies

The literature that discusses aspects of information systems development that are discussed in chapter 3 covers ISD paradigms, ISD approaches, and ISD methodologies. In chapter 3 we assume, based on the literature on professional service organizations, that the neo-humanist ISD paradigm is the best fit with the organizational domain. In this paradigm *information systems are developed to remove distorting influences and other barriers to rational discourse* (Hirschheim and Klein, 1989). Based on the neo-humanist paradigm, we assume that professional organizations develop IS to overcome a perpetual state of conflict in the processes, task assignment, and workflows. The other paradigms for ISD are thought of to be unfit for the professional service organizational domain. Based on this assumption the method chunk development process predefines the paradigm contingency variable in the identification stage of the approach.

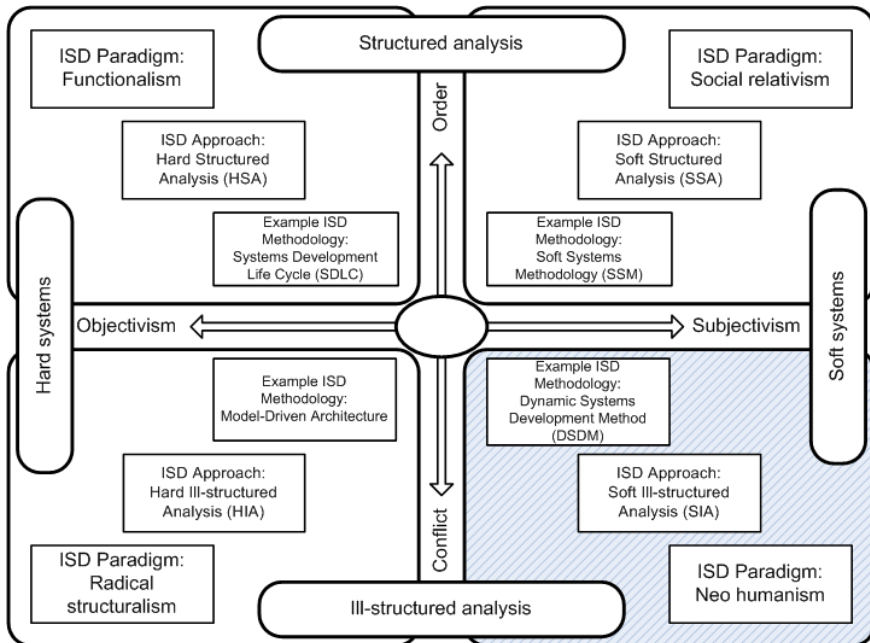


Figure 7-3: Summary of information system development.

However during the development of the method chunks as part of the overall ISD methodology in the case studies, it became evident that the neo-humanist paradigm for ISD is not the only paradigm employed. In the first BI&E stage the functionalistic paradigm is encountered. Based on this finding the approach to develop method chunks is altered, and the identification of the dominant ISD paradigm is selected as part of the method. The conclusion can be drawn that a no one ISD paradigm is necessary dominant in the PSO domain.

In line with the finding in regards to the ISD paradigm, we can draw similar conclusions regarding the ISD approach, and ISD methodology; by default there is no full fit with the PSO domain. The ISD approach sets the goals, guiding principles and fundamental concepts for the ISD process. The ISD methodology is an organized collection of concepts, methods, beliefs and procedures that guide the work of stakeholders involved in the ISD process. The ISD approach that aligns with the neo-humanist ISD paradigm is the soft, ill-structured analysis approach, which aims for functionality over process formality and assumes the multiple perspectives on the aspects of the organization to be modeled. ISD methodologies like the dynamic systems development method (DSDM) align in their place with this ISD approach. However the same conclusion is drawn here; there are other ISD approaches and ISD methodologies that equally well fit the PSO domain.

The only formalization that remains in this research in regards to the information systems development aspect is the overview presented in Figure 7-3, with the notion that the shaded area, based on literature, can be preferred for application in the PSO domain. This implies that based on literature and findings in the application of this research in practice we can confirm that a soft systems approach in which the organization and issues in that organization are ill-defined, combined with an ill-structured analysis (in other words: an agile IS development approach), has in principle the best fit with the PSO domain. In practice this approach is characterized by refuting the necessity for predefined structures in the ISD process flow, the need

for formal notation and documentation. It aims for delivering functionality over process formality. Furthermore, the insights of every stakeholder are required in the development process, because every stakeholder in the ISD process interprets the IS underdevelopment and its functionality from a different perspective.

### 7.3.2 *PSO domain*

In chapter 3 Leavitt's diamond model is adapted to map the characteristics of professional service organizations (Leavitt, 1978). Although extensions are suggested over the years, the original model proved to have the best fit with the task to characterize PSO for developing a situational ISD approach.

<p><b>Information technology:</b></p> <ul style="list-style-type: none"> <li>• The extent to which the IT is used to deal with coordination in complexity</li> <li>• <i>Extent to which IT is used to improve efficiency and effectiveness</i></li> <li>• <i>(De-)centrality of IT systems and organization</i></li> <li>• <i>(Low) IT maturity</i></li> </ul>	<p><b>Tasks:</b></p> <ul style="list-style-type: none"> <li>• Level of process definition (lack of)</li> <li>• (High) level of conflicting goals within a process</li> <li>• Level of process flow predictability</li> <li>• <i>Extent to which processes are operationalized by different departments / stakeholders</i></li> </ul>
<p><b>People:</b></p> <ul style="list-style-type: none"> <li>• Level of self-responsibility</li> <li>• Level of professional autonomy</li> <li>• Emphasis on quality and / or quantity</li> <li>• Availability of specific, unique knowledge</li> </ul>	<p><b>Structure:</b></p> <ul style="list-style-type: none"> <li>• Split governance structure</li> <li>• Effectiveness aim</li> <li>• Value shop organization</li> <li>• Processes complexity</li> </ul>

Table 7-2: Diamond model: PSO characterization relevant for ISD.



The diamond model consists of four areas that cover a specific aspect of an organization: information technology, tasks, people, and structure. In chapter 3 for each of these areas, professional service organizations are characterized.

During the development process in the two BI&E stages, described in chapter 4 and 5, it becomes evident that professional organizations do not score typically the same on every aspect of the diamond model. However the subcategories for the four aspect prove to be valuable for the characterization process of the organization in light of the IS development. In Table 7-2 the four categories from the diamond model are extended with subcategories, based on the literature review in chapter 3, and extended with the input from the BI&E stages in chapter 4 and 5. The extensions from the BI&E stages are highlighted in an italic font. In the sections below Table 7-2 each quadrant is explained in more detail.

### **Information technology**

From the literature study in chapter 3, the main aspect that characterizes PSOs is the extent to which the organization uses IS as a means to coordinate the complex structures and processes. From the BI&E stages in chapter 4 and 5 three more aspects are identified that influence the information system development process. The first is the extent to which IS is used to improve effectiveness and/or efficiency. IS can primarily be a means to attain higher levels of efficiency and effectiveness, but it can also serve other means, such as knowledge sharing. The extent to which an organization employs (or plans to employ) IS primarily for efficiency gains influences the ISD process – for instance the extent to which stakeholders can influence the development direction. As second aspect that is identified during the BI&E stages is the (de-)centrality of the IT systems in the current and future situation. Since activities in PSOs are based on very specialist knowledge, information systems can be organized according to these knowledge areas. The desire to integrate the systems supporting the knowledge areas depends on the connectedness of the expertise areas the IS supports. The third additional aspect identified during the BI&E stages is the maturity

of the IS and the IS organization supporting it. As with other type of organizations, this aspect influences the ISD process; high IS maturity enables a broad range of ISD methods (like many agile approaches) that are not applicable in an organization with low IS maturity.

### **Task**

Three aspects that relate to the task perspective of an organization are identified in the literature review in chapter 3, a fourth aspect is identified during the BI&E stages. The first aspect is the level of process definition. When processes are clearly defined and formalized, there is no need to spend time on this during the ISD process. Undefined processes are a major concern if a IS is to be developed that supports them. The second aspect is the level of conflicting goals in the organization. A high level of conflicting goals demands extra attention during the ISD process. The third aspect is the predictability of the process flow. PSOs are characterized by unpredictable processes (that is, when a process is started, the next activities are not yet know). This aspect closely relates to the level of process definition. In the two BI&E stages the researchers encountered many processes of which no process diagram could be drawn. If this is the case, the development of IS that supports these processes is organized differently than when processes are very predictable. An aspect that is identified during the BI&E stages is the extent to which processes are operationalized by different departments. If multiple departments execute parts of a single process, this implies the integration of the IS supporting this process. Combined with the (de-)centrality aspect of IS, the split responsibility of processes by departments requires extra attention during the ISD process.

### **People**

Four aspects in the people quadrant are identified in the literature review that are of importance to map for the ISD process. The level of self-responsibility determines the influence every stakeholder is required to have in the ISD process. Professional autonomy is closely

related to this aspect, and implies the extent to which staff is granted the right to make process decisions based on their professional expertise. If professionals have a high level of autonomy, the IS supporting their activities should enable them in their decision process. During the literature review the importance of quality over quantity in the performance measures of a PSO became evident. However during the BI&E stages, the importance of quantity in a PSO also has drawn attention. Therefore the identification of the importance of both factors is advised to be mapped for the development of IS in a PSO. This can be operationalized by the identification of performance measures like key performance indicators (KPI). The last aspect that is identified to be of importance to the development of IS in a PSO is the availability of specific / unique knowledge. As can be assumed, knowledge is one of the key assets in a PSO, maintaining, facilitating and storing this knowledge is therefore important. If knowledge is identified to be a key asset in the organization, the development of the IS supporting the processes in which the knowledge is applied is to be facilitated.

### **Structure**

Four aspects are identified to be of influence on the ISD process in a PSO. The split governance structure which is often found in a PSO implies at least two distinct stakeholders with possible contradicting stakes; the administrative management, and the specialist staff. If a PSO has this division in the governance structure is identified, the ISD process is influenced by the distinct stakeholder opinions on the IS development. Effectiveness can be defined for a PSO as various models. In this literature review in chapter 3, spatial model for effectiveness is adapted to identify the quest for effectiveness in an organization (Quinn and Rohrbaugh, 1983). This model is proven to be valuable in literature and also in the BI&E stages in this research. In chapter 3 a broad overview of this model and the four major alternatives in this model is presented. The value shop organization, which is typical of a PSO, determines the IS that supports it. In a value shop expertise knowledge is used to solve problems, make decisions in the process, execute the decision, and evaluate the

outcomes. This is a distinct process layout, which requires special attention when developing the IS. The last aspect that is to be identified when developing IS for a PSO is the complexity of the process. The more complex the processes are, the more effort needs to be put in with regards to analysis, optimization, and redesign during the IS development process in order to get the best IS alignment.

### ***7.3.3 Situational method engineering***

In chapter 3 situational method engineering is identified to be a major foundation for the approach developed in this research; a situational approach to develop method chunks for professional service organizations. This research contributes to the body of knowledge about SME on two areas; (1) an approach to create method fragments, and (2) the development of a SME-based methodology using the action design research approach. Literature clearly identifies the creation of method fragments, particularly based on meta-models, as one of the main challenges of SME field (Henderson-Sellers and Ralyté, 2010). Concerning the second area where this research contributes; ISD methodologies are criticized in literature for (a) not addressing the particular requirements of a specific project, and (b) not being able to adapt to changing requirements during the course of a project (Avison and Fitzgerald, 2006). The first concern is addressed by SME, though the second concern only is marginally addressed by SME in literature (Rossi et al., 2004). The SME workflow is by default rather traditional; first requirements are identified, then the method is assembled based on project characteristics and available method fragments, and finally the ISD project is implemented. By combining SME with the ADR approach, this research will deliver a means to use SME in project for which the requirements are not clearly identified upfront. A main reason not to define requirements upfront is a challenging business environment with many conflicting goals that cause changing requirement – like the PSO domain. In the following sections contributions to the field of SME are discussed in more detail; first

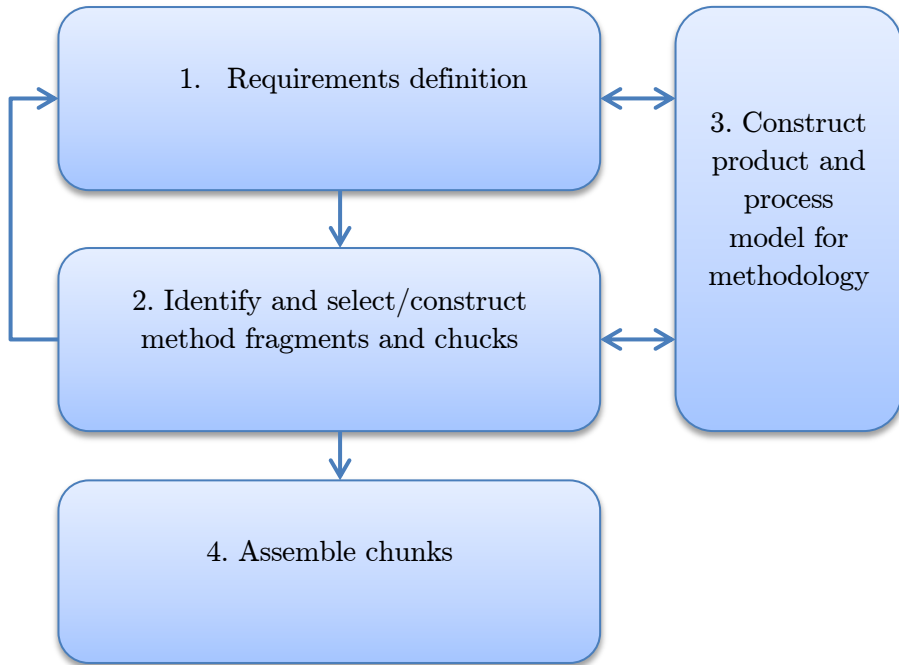
the adaptation of SME to the action design research based approach, and second a domain specific variation of SME.

### **SME-based approach adapted to ADR**

To construct method chunks that support organizational performance by enabling an effective ISD process (the IT-artifact), in this research the SME direction is selected. Although the construction of method fragments and chunks is only a subset of the overall SME process, for this research the main workflow of the SME approach is adapted and further enhanced with the ADR approach, see Figure 7-4. This is because of the end-goal of any method fragment construction process is to develop an ISDM. As is discussed that the paradigm-based approach to creating methodologies from fragments has the best fit with the PSO domain. Therefore the method-under-construction in this research is based on the paradigm-based approach. In essence, the paradigm-based SME approach consists of four stages: (1) define requirements; (2) design the methodology to-be based on a process model and a product model; (3) identification of method fragments, and (4) assemble the method fragments and chunks into a methodology – if the method base does not cover some parts of either the product or process model, fragments have to be constructed. Since the focus of this research is on the final statement; to construct fragments, the previous steps cannot be omitted without prior analysis.

Next the ADR perspective is to be inserted into the SME-based approach. Since one of the criticisms on traditional methodologies is that the static workflow does not align with the continuously changing organizational domain, the *traditional* view of SME has to be addressed for this research. ADR teaches us that the best way to design an artifact is to do it in close cooperation with the audience that will be using the artifact in future instances, see chapter 2. This implies that the current, paradigm based workflow is adapted to the ADR viewpoint. To do so, the process stages (define requirements, design methodology's process and product model, identify

fragments/chunks, assemble chunks into a method) are left the same, only the workflow is updated.



**Figure 7-4: SME process adapted to ADR. High-level view.**

Figure 7-4 depicts a high level view on the SME workflow when it is adapted to the ADR approach. This is no longer a sequential process; in this adapted SME workflow the definition of requirements is worked out in close cooperation with the design and construction of the product and process model. Simultaneously the method base is scanned for available fragments, and if these are lacking method fragments are constructed. However, due to the continuously updated requirements and corresponding product and process model, the identification / construction of method fragments is fully dependable on whether the fragments still fill a void in the overall methodology.

The focus of this research is, as stated at this chapter's introduction, on stage 2 in the figure: the construction of method fragments.

## Domain specific

As is discussed in the literature overview, the PSO domain is a very specific area where traditional methodologies for ISD have not been successful. Main reason that is discussed in literature is the incompatibility of standard methodologies with the key characteristics of PSOs, as is discussed in section 7.3.2 (i.e. information technology, tasks, people, and organization structure). Although the adaptation of methodology instances derived with SME is discussed in literature (Ralyté et al., 2003), the issue how to define whether a method fragments covers a specific domain is still a topic of discussion (Han et al., 2008). This research aims to develop a domain specific method to construct method fragments. Therefore the traditional SME approach is slightly adapted to insert the domain-specific element. For the construction of the method to develop method fragments, the domain characteristics are taken *a priori* into account.

## 7.4 Generalized approach

In this paragraph a generalized approach is described, with which is implied: based on the initial design from the literature review from chapter 3, that is further developed in two iterative building cycles in chapter 4 and 5, upon lessons-learned are draw during the reflection stage in chapter 6, an general approach is presented that to the best insights of the researchers resembles the most suited solution to the research statement set in chapter 2: a situational approach to develop information systems for an organization in the professional services domain. The provided approach in this research consists of a process and a set of design principles, which are both described in chapter 6. The addition in this paragraph is a proposed narrowed down description of the matching process of the available method chunks with the PSO domain. Both the elements in the proposed approach, the PSO domain and method chunk development, are discussed in the previous paragraph, in which the results from this action design research are reconnected to the original theories which underlie the design developed in this research.

### 7.4.1 *Method to develop method chunks*

The method to develop method chunks for PSOs consists of four main activities, as is discussed in chapter 6. In this section the situational approach to develop the method chunks is presented. This approach is developed based on the research presented in the previous chapters.

#### **Process**

The process consists of four stages, (1) the identification of contingency variables; (2) the definition of the scope of the method chunk to-be; (3) the selection and assembly of the method chunk; and (4) the implementation of the chunk. The process is designed based on the *agile methods* paradigm, which implies the iteratively deployment of the four stages. Especially stage 2 and 3 are proposed to be executed simultaneously, see Figure 6-2 on page 199.

*Identification of the contingency variables.* Two elements of the organization under-review are to be identified: the ISD process, and the organizational characteristics. The ISD process is analyzed using the elements that are presented in the Figure 7-3. Using the elements from this figure provides insight what type of ISD paradigm, approach, and methodology is adapted in the organization under-review. The second element is the analysis of the PSO characteristics. Although a PSO cannot be typically characterized on the diamond model for organizational characteristics, as is assumed in the initial design, the set of characteristics that are of importance for the development of IS are determined during the course of this research. Four main characteristic categories are identified, based on the diamond model (Leavitt, 1978). The categories are the information technology, task, people, and structure. For each category several aspects are identified that characterize a PSO in regards to the development of IS. These aspects are listed in paragraph 7.3.2. The analysis from the two sets of contingency variables provides a firm foundation for the following stages in the method chunk development process.



*Definition of method chunk's scope:* After the contingency variables are defined for the organization under-review, the scope of the method chunk that is to be developed can be defined. Naturally the method chunk's scope is based on the contingency variables that are defined in the previous stage. The scope of the method chunk defines the process and the products of the method chunk. The process consists of the activities that are to be part of the ISD process, for instance workshop activities, (re)design activities, and analysis activities. The products are the (intermediate) deliverables of the method chunk like design documents, agreements, and analysis documents / models.

*Fragment selection and chunk assembly:* As the method chunk's process and products are defined, a selection can be made from available method fragments that fulfill the requirements defined in the previous described stages. As is demonstrated in this research, a method chunk consists of at least a product oriented fragment (a fragment that delivers a product) and a process oriented fragment (a fragment that predominantly consists of a series of activities). There are several means to select a fragment that is suitable for the requirements that are defined, as is described in chapter 3. The most common means is to select a fragment from a repository, but a fragment can also be reverse engineered from existing methodologies, or build from scratch. For this research we have selected existing, established method fragments. When at least two fragments are selected, they have to be configured to fit the specifics of the organization and its ISD process. Each method fragment can be adapted in terms of the contingency variables that are identified in the first stage. When the selected fragments are adapted, they can be assembled into a method chunk that can be implemented.

*Implementation:* When at least two method fragments are assembled into a method chunk that meets the requirements based on the contingency variables of the organization, the method chunk can be implemented. The implementation consists of the provision of guidance on how to use the chunk and the actual usage itself.

### 7.4.2 Method chunks for professional service organizations ISD

In this section the method chunk itself the main topic. The method chunk to-be-developed is the result of the approach that is described in the previous section. In this paragraph a meta-model for method chunks for the PSO domain is presented as well the application area.

#### Meta-model

A meta-model for the approach to construct method chunks can be deduced. Figure 7-5 depicts this model. It is based on the depiction of the process to construct method chunks as is described in paragraph 7.4.1.

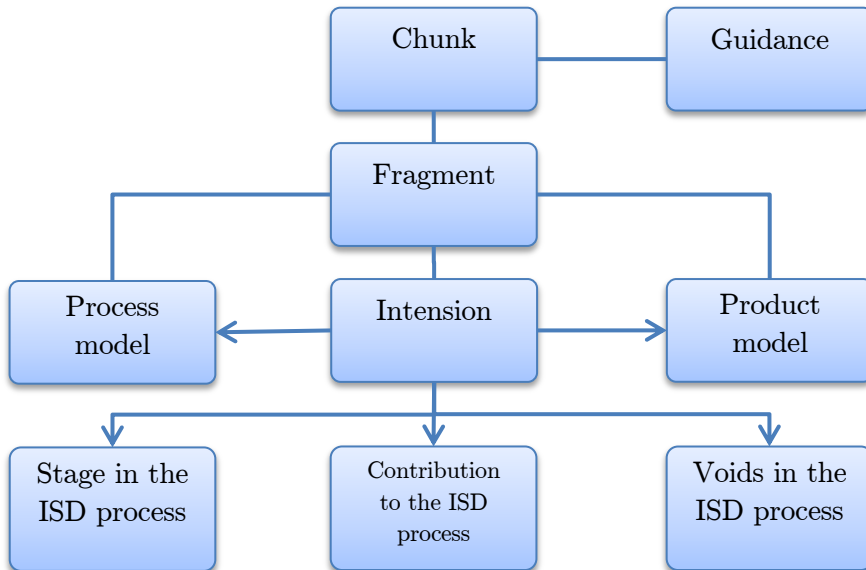


Figure 7-5: Meta-model: method to construct method chunks.

In this model environmental variables indicate the *intension* for the method chunk. The *intension* concept resembles the results from the analysis of contingency variables in the IT-artifact's process. The *intension* concept is reused from the method engineering meta-model, as is presented in Figure 3-15 on page 85. Based on the intension, a *product model* and a *process model* are developed, in close

interaction with the selection of *method fragments*. After their selection the method fragments are assembled in a *method chunk*. Finally the *guidance* is provided on how to use the chunk.

## **Application**

In this section the application area of the method to develop method chunks is set. Because the level of abstraction in the previous sections is rather high, in this section some real-world (possible) applications are listed that fulfill the general requirements for PSOs as described in section 7.3.2.

*Serious gaming*: is developed to solve problems in organizations. Serious gaming can be a process oriented method fragment in the context of this research. A game can be played about the developed direction of IS, that is the process to refine the requirements of the IS under-development. A serious game is intended to serve training, simulation and education purposed in organizational development (Alvarez and Rampnoux, 2007). A serious game is not a simulation alone; it can be a simulation combined with elements from gameplay, specifically the chance to win. The intension of the application of serious gaming in an ISD process is to motivate, educate, and train the players.

*Business modeling*: is a description of how an organization captures, delivers, and creates value (Osterwalder and Pigneur, 2010). Although business model construction is originally part of the business strategy, it can be as well used for the formal and informal description of infrastructure, goals, organizational structures, and operational processes. The aim of business modeling is to create a high level view on the aspect of the organization that is being modeled. The goal of business modeling is to draw and explore possibilities for future development (Osterwalder and Pigneur, 2002). In the context of IS development, business modeling can be an important method fragment to capture, design, innovate, and transform the processes that are supported by the IS that is developed (Chen and Jung, 2009). This method fragment fits the scope of the approach described in this research (the methods to

develop IS for the PSO domain), because business modeling is a multi-disciplinary activity, involving multiple stakeholders, that don't require specific modeling knowledge. It offers the practitioners in the modeling endeavor insight in their own (complex) processes, and processes that are operationalized by other stakeholders in the organization.

*(Paper) prototyping*: a user-centered design method chunk that aids in the IS development process to create an IS that meets the end-user's expectations, especially for the design of interfaces. The idea is to develop roughly drawn prototypes to present the design under-development to the end-users, with which they get better insight, and are better involved in the development of the IS during the requirements analysis and building stage (Snyder, 2003). Paper prototypes are advised to be applied when: (1) tools the designer wants to use in creating a prototype are not available, (2) the designer wants to make a sincere effort to allow all members of a team, including those with limited software skills, to take part in the design process, and (3) the tests of the design lead to a great deal of drawings (Sefelin et al., 2003) This is a very suitable method fragment for development of IS in the PSO domain since various stakeholders are directly involved in the development of the IS that is to be supporting the processes the stakeholders are participating in. The various perspectives are clearly identifiable, and future design can be conducted collaboratively.

## 8 Conclusions

This research proposes that the various issues with development of IS in professional service organizations can be addressed with the situational development of method chunks; dedicated parts of methodologies for IS development, tailored to organizational specific requirements. This chapter revisits the problem statement as formulated in the first chapter and reflects on the key findings achieved in this research. Furthermore, in light of the key findings, in this chapter the main contributions, limitations and future research agenda are discussed.

### 8.1 Review of the problem statement

In chapter one the problem statement is formulated as:

*The development of a situational method to develop information systems for professional service organizations.*

To scope and specify this research's problem statement, the following research objectives are formulated:

- Characterize the professional service organizations domain
- Identify the information systems development approach that aligns with the organizational characteristics of organizations in the PSO domain
- Provide an approach to develop information systems development methodologies specific for professional organizations.

This research has used action design research approach (ADR) to provide a solution for the problem statement. Because of the ADR based research approach, the solution direction in this research is formulated iteratively; step by step the solution for the problematic development of IS for organizations in the PSO domain is aligned with the situational method engineering paradigm. The ADR process flow in this research consists of four stages; the problem formulation

stage; the building, intervention and evaluation stage, the reflection stage, and the formalization stage. In this research the IT-dominant version of the ADR approach is applied, by which the building, intervention and evaluation stage is iterated twice. There are two cases selected for this research, both in organizations in the PSO domain. The first building, intervention and evaluation iteration is conducted in an organization in which solutions to the problem statement are designed by only involving professionals from the organization. In the second building, intervention and evaluation stage a second version of the solution to the problem statement is designed by involving both professionals and end-users from the organization. Both the designed solutions are instances of a class of solutions to a class of problems. In the reflection stage the class of solutions is defined as the development of method chunks that fill a void in an IS development methodology. In the formalization stage the class of solutions is formalized using the concept of technological rules to provide design rules for future similar problems in other PSOs.

## 8.2 Review on the research process

In this paragraph the development process of the IT artifact is reflected upon. Table 8-1 summarizes the ADR process that leads to the current state of the IT artifact.

STAGES		ARTIFACT
Stage one: Problem formulation		
Principle 1: Praxis inspired research	Research is driven by the lacking methodological approach to develop IS for organizations in the PSO domain. The PSO characteristics and current IS methodologies are identified as the root causes of this problem.	Recognition: Methodologies do not have to cover every aspects of the ISD process. Dedicated approaches custom fit to a specific organizational setting are described in literature as more effective. The aim of this research is focused on the development of domain specific method chunks in the PSO setting. As
Principle 2: Theory ingrained	ISD Methodologies have considerably evolved over the last decades. The focus has	

artifact	shifted from traditional structured, sequential approaches to less structured modular approaches. Situational method engineering is selected as supporting paradigm for this research.	a result of an extensive literature review an initial approach for developing method chunks is designed.
<b>Stage two: BIE; Alpha version</b>		
Principle 3: Reciprocal shaping	In two BIE iterations method chunks are developed based on the insights from the initially develop approach to construct method chunks. Feedback from practitioners and end-users provides directions to adapt the development process instantaneously.	The initial design of the approach to construct method chunks in PSOs is enhanced; the process sequence is adapted and some elements from the process are eliminated, other elements are added. Using this approach in this alpha stage a method chunk is developed.
Principle 4: Mutually influential roles	The ADR team formed includes researchers and practitioners. Both parties actively participated in the development of the method chunk <i>goal modeling</i> .	
Principle 5: Authentic and concurrent evaluation	The goal modeling method chunk is evaluated within the ADR team. Based on the evaluation of the method chunk the design of the IT artifact is further improved.	
<b>Stage two: BIE; Beta version</b>		
Principle 3: Reciprocal shaping	Initial case analysis is conducted to apply the IT artifact in this case. Ideas, problems and opportunities are discussed in ADR team sessions. In this BIE stage the method chunk development focuses on the lack of awareness for current organizational processes.	The method chunk under development and the approach to develop a method chunk are further improved using iterative feedback sessions. The approach to construct the method chunks is adapted to cope with less structured organizational environments.
Principle 4: Mutually influential roles	Since the second iteration of the BIE stage is implemented in a different organization, a new ADR team is formed.	The updated IT artifact's process is now initiated by the identification of

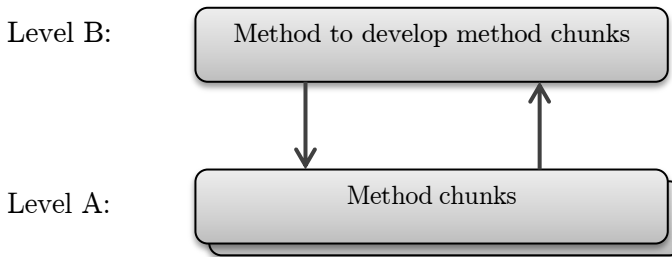
Principle 5: Authentic and concurrent evaluation	<p>Again researchers and practitioners from the ADR team.</p> <p>The approach to develop the method chunk in this BIE iteration is evaluated in iterative cycles with both practitioners and end-users.</p>	<p>environmental variables, next products and suitable method fragments are selected simultaneously with the definition of the process flow for the method chunk. When the method chunk is assembled guidance is provided.</p>
<b>Stage three: Reflection</b>		
Principle 6: Guided emergence	<p>The two BIE stages provided insight related to the PSO domain and ISD in the PSO domain. The most important finding is that an agile, domain specific approach to the development of method chunks can be distilled from the two BIE iterations</p>	<p>Emerging vision: the evolution of the initial approach to construct method chunks is discussed. Based on the reflection in the two BIE stages an improved design is proposed to develop method chunks. This approach consists of a suggested process, products to be delivered and a set of design principles.</p>
<b>Stage four: Formalization</b>		
Principle 7: Generalized outcomes	<p>In the formalization and learning stage the results of this research are reconnected to the underlying theories. Based on the research outcomes contributions can be made to these original theories.</p>	<p>An improved perspective on ISD methodologies for organizations is presented alongside an updated view on the PSO domain itself. Next to that extensions to situational method engineering are suggested. Based on the literature review, the case studies in two distinct organizations in the PSO domain and reflection on this research a generalized approach to develop ISD methodologies for organizations in the PSO domain.</p>

**Table 8-1: Summary of the ADR process to develop a method to construct method chunks.**



### 8.3 Key contributions

In this research five key contributions can be identified: (1) the development of information systems in organizations in the PSO domain; (2) the development of two method chunks; (3) an approach to develop method chunks in organizations in the PSO domain; (4) improved insights in the situational method engineering approach; and (5) a concrete application of action design research. In this paragraph these contributions are discussed. To amplify the relation between contribution 2 and 3, Figure 8-1 is depicted below. In this research there are two artifact types developed: a method to develop method chunks (level B), and two method chunks (level A). The development of level B and level A are mutual influencing. On the other hand there is no strong relationship between the two developed method chunks in the two distinct organizations. The chunks are developed in two separate organizations, with separate ISD processes, with specific characteristics.



**Figure 8-1: Relationship between development method and developed method chunks.**

#### *8.3.1 ISD in the PSO domain*

In this research the problematic development and configuration of information systems for organizations in the professional service organization domain is analyzed. Based on the ADR approach, a firm literature review provided the basis for the application of initial findings in two organizational settings. The ADR approach was aimed at the mutual analysis and development of the problems and their solutions in this field. As a result the major results of in this

research include a characterization of typical professional service organizations and an approach to develop IS in this kind of organization. In chapter 1, in section 7.3.1 and 7.3.2 the contributions to both bodies of knowledge are identified. In this section the development of IS specifically in organizations in the PSO domain is discussed.

As is discussed in section 7.3.1 ISD can be mapped on the schema developed by (Hirschheim and Klein, 1989). Two axis are identified in this schema; the level of structuredness, and the assumption to which extent problems are be perceived as hard or soft problems. The two axis form four quadrants, see Figure 7-3 on page 214. For each quadrant an ISD approach and ISD paradigm are defined in chapter 1.

Next to ISD, literature on professional service organization is reviewed in chapter 1. Using the framework from (Leavitt, 1978), four characterizing areas of PSOs are identified; the use of information technology, tasks, people and structure. Based on literature, for each of the areas characteristics for PSOs are identified, see Table 3-2 on page 71. This table is updated during the course of this research. The final insights are presented in Table 7-2 on page 216.

During the literature review the soft-systems – ill-structured quadrant of the schema is selected to be the best fit with professional service organizations, however during the course of this research it is concluded that other type of ISD paradigms are also applied in this organizational domain. Therefore, the soft systems – ill-structured ISD approach is not identified as the default approach in the PSO domain, but in developing method chunks it is preferred to be the fundamental starting point for the method chunks to be developed.

Development of IS in an organization in the PSO domain is based on the situation method engineering approach, this can be concluded as a finding in this research. In the SME approach, methodologies for IS development are constructed by combining smaller method fragments and chunks to combine to a custom methodology. In this

research process consisting of four stages is configured to develop tailored method chunks. This process is describe in paragraph 6.5 on page 195 and in paragraph 7.4.1 on page 224. Because of the dynamic nature of PSOs the iterative process is operationalized in an reciprocal approach, in which stages influence activities in other stages.

### ***8.3.2 Development of two method chunks***

The practical application of the ADR in this thesis implies that artifacts are developed during the testing of the research hypothesis itself. In this research a hypothesis is formulated in terms of an initial design of an approach to develop method chunks, accompanied by an outline of a method chunk itself. The next step in the ADR approach was the development of artifacts based on the initial design. In two iterative building stages, two method chunks are developed in two distinct organizations. As stated in the introduction of paragraph 8.3 there is no strong relationship between the two method chunks, since both chunks are developed in distinct organizations with specific characteristics.

In the first organizational setting, an alpha version of a method chunk is developed; the goal modeling – collaborative design method chunk. This method chunk addresses the need to agree on requirements before a system is being implemented. Many conflicting goals are identified in the organization in the process to develop the method chunk, and as such the requirements for the system to be implemented could not be defined. Professional employees in this organization have their own professional autonomy and responsibility, and as such control other performance output measures such as output and quality. Because of the nature of the organization at hand (and of organizations in the PSO domain in general), the decision to elicit a sub-set of goals as the requirements for the system to be developed was not easy to make. The method chunk contributed in the ISD process by first identifying the various (conflicting) goals in the organization in regards to output, quality, autonomy, and responsibility. After the goals were identified, the

goals were modeled which resulted in a goal model. This model was used as input for the collaborative design sessions, in which was decided which the goals form the requirements for the future system. The development of this method chunk and its implementation in the organization is presented in chapter 4.

In the second BI&E stage, another method chunk is developed. In another organization a beta version of a method chunk that addressed the lack of definition and awareness of current processes that needed to be supported by a new to be developed IS. Because of the organic evolution of the organization, the processes were suboptimal. Just further supporting those processes with a new IS would not improve the situation. Therefore a method chunk was developed that addressed this sub-optimal configuration and, at the same time, identified the requirements for the IS to be developed. As with the first chunk, this method chunk also consisted of two method fragments combined. The first fragment analyzed the sub-optimal configuration of the processes and the organization by simulating current and (possible) future layouts. The second method fragment that formed the method chunk applied collaboration as a means to design future configurations. Hence the simulation – collaborative design method chunk was developed. By applying this method chunk first a simulation model was created in which all kinds of process, IS, and organizational configurations could be simulated. This simulation model was then used as input for the collaborative design sessions in which every stakeholder in the organization was invited to put forward ideas, issues, and objections towards a future configuration of the organizational processes and structure. Output from the stakeholders was then used as input for the simulation model, with which possible configuration could be tested. The development and implementation of this method fragment is presented in chapter 5.

### ***8.3.3 Method to develop method chunks***

Based on the literature review an initial design was created to develop method chunks. In the two BI&E stages two method chunks

are developed, as described in paragraph 8.3.2. By combining the initial design and the evaluation and reflection on the approach to develop method chunks an overall approach could be deducted to develop method chunks. In paragraph 6.5 and paragraph 7.4 this approach is presented.

In essence this approach consists of a description of a process, the definition of the products, and a set of design principles.

In Figure 6-2 on page 199, the general outline of the process to develop method chunks is depicted. The process consists of four stages which are executed iteratively and reciprocally. This implies that specific activities in each stage on some occasions are conducted in parallel with activities in other stages. In terms of the ISD literature that is discussed in paragraph 3.4 this approach maps with the ill-structured approaches, also termed agile approaches. The four stages that form the method to construct method chunks are: (1) the identification of contingency variables; (2) the definition of the method chunk's scope; (3) the selection and assembly of the method chunk itself; and (4) the implementation of the method chunk. Especially activities in the second stage influence activities in the third stage and vice versa. In the first stage (a) the ISD process is characterized as is being executed in the organization and (b) the organization itself is characterized according to the aspects that are relevant for IS development, as is discussed in section 8.3.1. In the second stage (a) the process for the method chunk under-development is defined, and (b) the products the method chunk under-development should produce are identified. In the third stage (a) suitable method fragments that fill in the method chunk's scope, as defined in stage two, are selected, (b) the fragments are configured to fit the specific needs of the organization, and (c) the fragments are assembled into a method chunk. In the fourth stage (a) guidance is provided to use the method, and (b) the method is implemented in the overall ISD methodology.

Four products are to be described that result from the process to develop method chunks; (1) the method chunk's description, in order

to reuse and store the method chunk; (2) the description of the method chunk's products; (3) the description of the method chunk's process; and (4) the guidance on how to use the method chunk.

The final element of the method to construct method chunks is a set of design principles. The set of design principles that is identified for method chunk development is depicted in Table 6-4 on page 204. For the development of method chunks four design principles are defined: (1) the involvement of end-users, with regular feedback loops during the development activities; (2) upfront and adaptive definition of tasks; (3) integration of perspectives from various stakeholders; and (4) develop the method chunks by applying agile techniques to guide the process.

#### ***8.3.4 Situational method engineering based on ADR***

In paragraph 7.2.3 we discuss that this research contributes to the SME body of knowledge by (1) developing an approach to construct method chunks, and (2) by mapping SME on the action design research paradigm. In section 8.3.3 the first contribution to SME is discussed. In this section the second contribution to the SME field is presented.

The situational method engineering approach as is developed in (Brinkkemper, 1996) and, amongst others, discussed in an extensive literature review in (Henderson-Sellers and Ralyté, 2010), is in essence a very structured, sequential approach. The mapping of SME on the ISD characterization scheme that is presented in chapter 1, is depicted in Figure 8-2. This figure is a result from the literature review that is described in chapter 3.1, specifically paragraph 3.4.

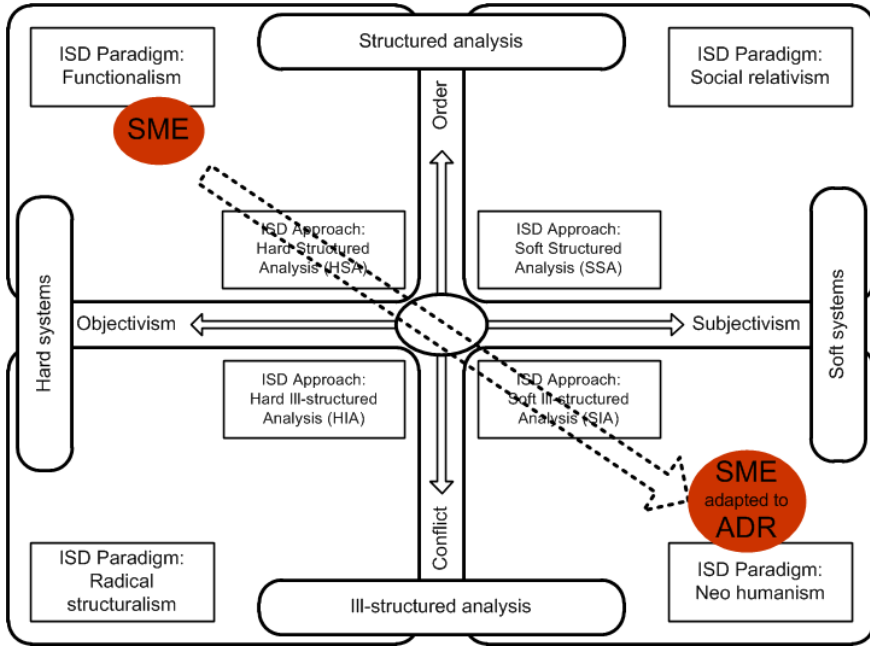


Figure 8-2: SME and SME-ADR positioning.

As presented in the traditional literature on the topic, the SME approach is well-structured and sequential. The organizational domain is perceived from the hard systems perspective (problems are well-defined and there is a straightforward means to solve them) and the process to address ISD is well-structured. Therefore *traditional* SME resides in Figure 8-2 in the traditional *functionalism* ISD paradigm. In this research, we adopted SME to the ADR approach, making the method chunk under-development and the organizational domain in which it is developed mutual influencing. Because of this adaptation of SME to ADR it corresponds to the *neo humanism* ISD paradigm. We conclude as key finding of this research that this adaptation of SME provides an increased applicability of situational method engineering in the professional service organization domain.

### 8.3.5 Application of the ADR paradigm

The final main contribution in this research is a large scale application of action design research as a research method. ADR was

first introduced in (Cole et al., 2005), but reached a plateau of acceptance with (Sein et al., 2010). In this research this relatively new methodology is adapted both in execution and as a framework for structuring this thesis. As is discussed in chapter 2, ADR consists of four stages which are operationalized iteratively and sometimes parallel. The main advantage of ADR is that it most closely resembles the *engineering process*, the ill-structured, but very effective approach to develop solutions to problems that are not encountered before. The downside of the ADR approach is also the ill-structuredness, which makes it hard to document and repeat in other situations. Since the novelty of this research methodology, extra attention is being paid to the detailed description of the application of the method in this research. In section 2.2.1 the four stages of the ADR approach are presented; the problem formulation, the building, intervention and evaluation stage, the reflection stage and the formalization stage. As is discussed in section 2.3.3, out of the two variants available for ADR, this research adopts the IT-dominant flavor. This implies that the BI&E stage is iterated twice, the first time to develop an alpha-version of the artifact under-development, the second iteration to develop a more elaborate version of the artifact.

The difficulty ARD poses by using it both as a framework for executing research and for structuring the written results, is the distinction between the development progress in terms of the temporal axis and the progress results in terms of a structured reproduction. Because of the iteratively, reciprocal workflow ARD proposes, the development of the artifact under-development and the actual results the development activities produce are not always in sync. In this thesis the *out-of-sync* issues are addressed by using the ADR approach as an framework to structure the results.

Because of the ability to deal with ill-defined problems, ADR has proven to be a major contribution to this research. Without the explicit aim to co-develop hypotheses about the artifact to be developed and the development of the artifact itself, the current contributions would not be possible.



## 8.4 Limitations: No silver bullet

In chapter 1 the problematic development of information systems for the professional service organizations is discussed. In the same chapter the mainstream current solutions are discussed too, only to conclude, as Brooks states, that there is no silver bullet for this problem (Brooks, 1987). In this research ideas from situational method development are applied in the professional service organization domain. Although SME puts the development of IS for this particular organizational domain in a new perspective, we cannot conclude that SME is the silver bullet either.

As with any research, the outcomes of this research are bound by a set of limitations. In this paragraph three limitations are discussed; the limited predictability of outcomes that are outside of the dataset, the experimental nature of the research method applied, and partial implementation of the research in the PSO domain.

### **Hume's problem**

In essence, Hume's problem is the problem of induction. The question is to what extent we can reason outside of the scope of the data set that is the result of this research. Although Hume himself does not use the word induction, Hume's problem is usually divided in two; the descriptive problem: how do stakeholders form opinions about unobserved facts; and the normative problem: are beliefs formed in this way justified (Millican, 1995). The problem with induction is that there is no assurance that results we have found in this research can be assumed to hold for the whole PSO domain. Initially we can only conclude the results are valid for the organizational cases we have applied them in. And even in these cases, we, as researchers, have been the actors that implemented the changes in the environment, we, as researchers have also been the observers of the effects of the implemented changes. Therefore, as Hume states, every hypothesis that appears to be the ultimate explanation should be refuted (Hume, 1978).

### **Experimental research method**

Action design research is applied in this research as the main approach to analyze the problem domain and configure a solution to the identified problems. This research method is a relative novelty. The first main publication on ADR dates from (Cole et al., 2005), but ADR became only mainstream with (Sein et al., 2010). As discussed in chapter 2, paragraph 2.2, ADR is a blend of design science and action research. ADR differs from the comparable design science in the sense that the problem analysis, solution design, implementation and evaluation are conducted synchronous. Whereas in design science first the solution to the problem is developed, then implemented, and eventually evaluated, in ADR these stages are operationalized at the same time. This is the influence from action research where the researcher is both observer and the person the implements a change in the organizational domain. A potential problem here is the researcher bias, as is discussed in the previous section.

Action design research is therefore a relatively new approach to the information systems field. The number of research cases in which ADR has been applied is limited. As Ostrowski formulates: “ *thus, though generally highly regarded and widely cited, DS methodological guidance from the precursors Hevner and Walls that existing guidelines and methods are insufficiently clear, or inadequately operationalized - still too high a level of abstraction [...] stating lack of a commonly accepted reference process model for DS researchj guide researchers across the DS lifecycle*” (Ostrowski and Helfert, 2011).

### **Partial implementation**

The implementation of the products of this ADR research are not implemented in the case organization to the full extend. Since this research followed the IT-dominant version of ADR, the first iteration of the building, implementation and evaluation stage is implemented to a very scoped area of the organization, namely the end-users were excluded from this first iteration. In the second BI&E stage, the

developed method chunk is implemented in a broader scope of the organization; end-users are involved. However the organizational scope of the second iteration was limited to the outpatient clinics of the hospital.

## **8.5 Future research agenda**

The application of situational method engineering in the professional service organizations domain, used to develop method chunks that aid specific needs in the information systems development process, as is topic of this research, is not a main stream approach yet. Lack of awareness, assumed complexity, and the need for upfront development in case of application of the method, are reasons that prevent SME to be adopted as a main stream approach. In this paragraph the main ideas to overcome these barriers to adoption are discussed.

### **More method chunks and method base**

In this research two method chunks are developed and configured that fit the requirements of two distinct organizations. If configured to other organizations' requirements, these method chunks are applicable in other PSOs as well. It is desirable to develop more method chunks that fit the general profile of the PSO domain. In chapter 1 some suitable method chunks are listed that can be subject of future research. By developing more method chunks, and storing them in a method base, specifically aimed for this organizational domain, the barrier to apply the approach that is developed in this research is lowered. Method chunks for this organizational domain can be developed based on a broad array of inputs, as is discussed in the literature review on SME in chapter 1, but in this research demonstrates that parts of the agile ISDM's are likely to have the best fit with the domain. In paragraph 6.4 a set of design improvements are listed that aid in the development of method chunks for professional service organizations.

**Case studies with practitioners**

The key to adoption of the approach that is developed in this research is the application of it by practitioners in the information systems development field. Herewith the approach can be validated in a broader variety of organizations in the PSO domain. In this research the organizations in the PSO domain are limited to healthcare organizations. Since healthcare organizations are assumed to be characteristic for the PSO domain, the results of this research are expected to be transferable to other PSO organizations, like judicial organizations, educational organizations, and consulting firms. Naturally this assumption has to be evaluated. Based on the experiences in this research project, the exact need for a method chunk is most unclear upfront. Therefore the agile approach is advised to develop method chunks. When more evidence of other case studies is provided, the awareness about the usefulness and effectiveness of this approach is raised.

## 9 Appendices

### 9.1 Appendix A: Process description medical diagnostics, MST case

The case description will be provided by means of a process model of both the neurosurgery treatment process of HNP, and the process model of the medical diagnostics as provided by the radiology department.

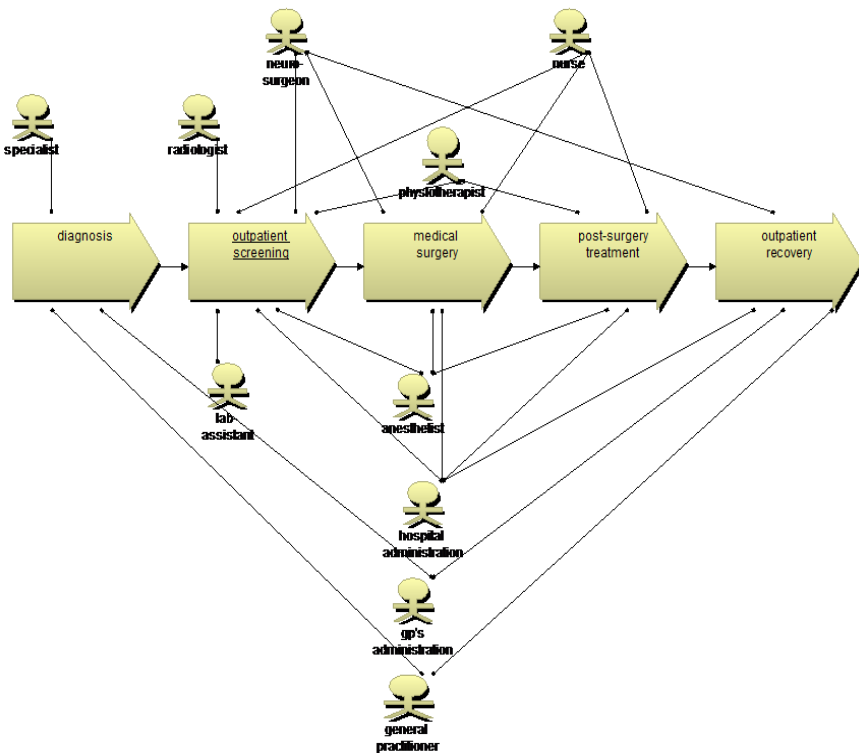


Figure 9-1: Overview process model HNP treatment.

The process overview of the clinical trial can be modeled as in Figure 9-1. In this overview five phases are involved, ranging from the diagnosis to the final recovery. Only the medical surgery and the post-surgery treatment are fully conducted within the hospital itself

(clinical). The actors involved in each process part are linked to each step.

When zooming in on the outpatient screening (second module in the process overview), we get the overview depicted in Figure 9-2. In this part of the treatment 10 major process steps can be identified, ranging from amnesias (collecting patient information) to the final preparation and check on the patient file.

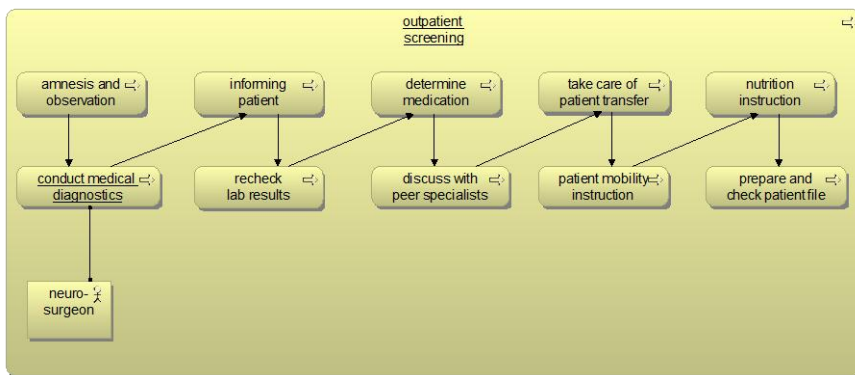


Figure 9-2. Overview process model outpatient screening, HNP treatment

Finally, when zooming in on the medical diagnostics process module (second module), as part of the outpatient screening, the process model can be displayed as in Figure 9-3. In this process, two diagnostic operations are performed; the lab (e.g. body fluids) testing, and the radiology (body imaging) testing.

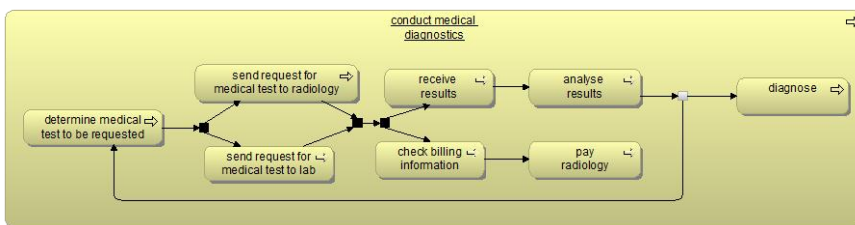
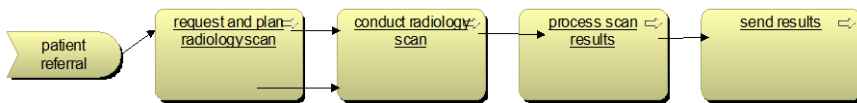


Figure 9-3. Medical diagnostics process, part of outpatient screening, part of HNP treatment.

In both the lab and radiology testing, basically, the requests for testing are accepted (or denied), the test itself is scheduled, the diagnostics are performed, and the test results are processed. After this step, the results are sent to the requesting department (in this case the neurosurgery).

The second viewpoint in this clinical trial, we zoom in on the radiology department, as part of the HNP treatment process. In this report we will focus on one location of the hospital (out of 3 radiology departments in the overall hospital group). In total, this hospital group conducts yearly over 200,000 radiology procedures, the selected radiology department provides over 140,000 of these. In the department 30 lab assistants, 8 doctors assistants, 13 radiologists, and 4 secretaries are working. In this department 6 rooms for radiography, one room for Magnetic Resonance Imaging (MRI), and 2 rooms for Computed Tomography (CT), are available.

The process as viewed from the radiology department can be modeled as in Figure 9-4. This figure displays the process overview, from the moment the patient is referred to by another department (in our case study: the neurosurgery department). Next the patient is tested by means of a body scan. Hereafter, the results from the tests are processed and sent to the requesting department. Finally the diagnosis process is finished.



**Figure 9-4. Radiology department process**

This whole overview of the radiology process model compromises the *outpatient screening* part of the neurosurgery process model. When zooming in on each individual process model, the following seven sublevels of the process model can be identified.

**A. Request and plan radiology scan.** Actors / roles involved are the secretary and the patient.

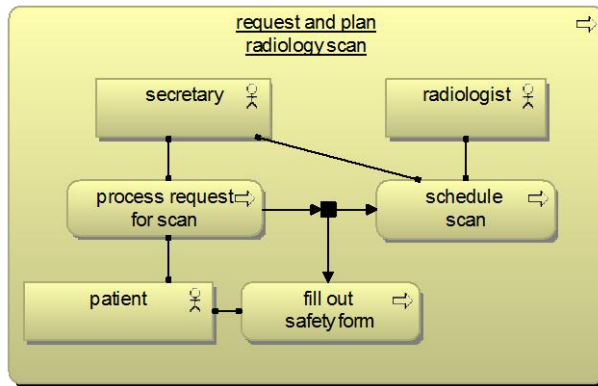


Figure 9-5: Actors involved in the process.

**B. Conduct radiology scan.** Actors involved are the secretary, the patient, the lab assistant, and the patient.

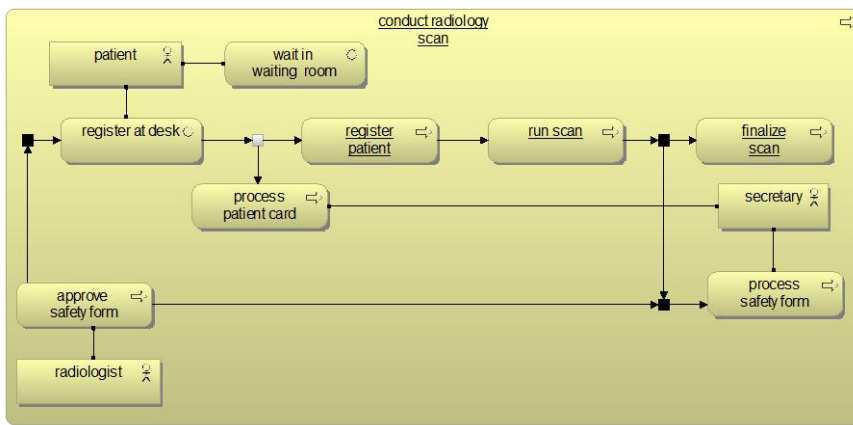


Figure 9-6: Conduct radiology scan process.

**B1. Register patient.** The actor involved is the lab-assistant.



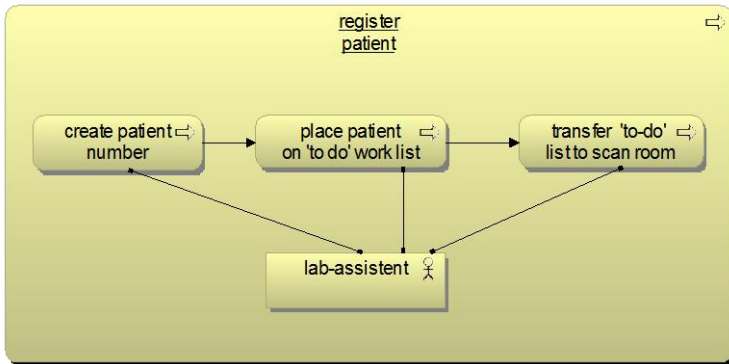


Figure 9-7: Register patient process.

**B2 Run scan.** Actors involved are the radiologist, the lab assistant, and the patient.

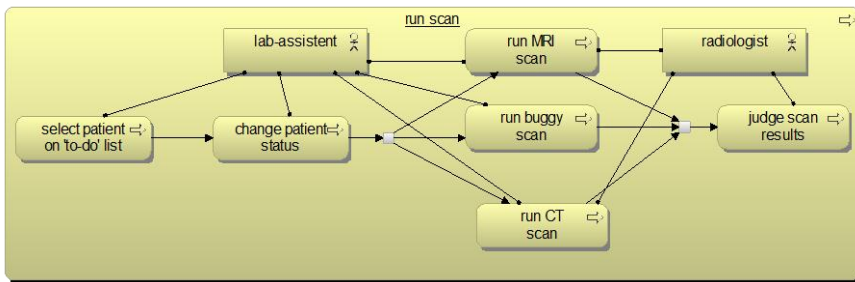


Figure 9-8: Run scan process.

**B3. Finalize scan.** The actor involved is the lab-assistant.

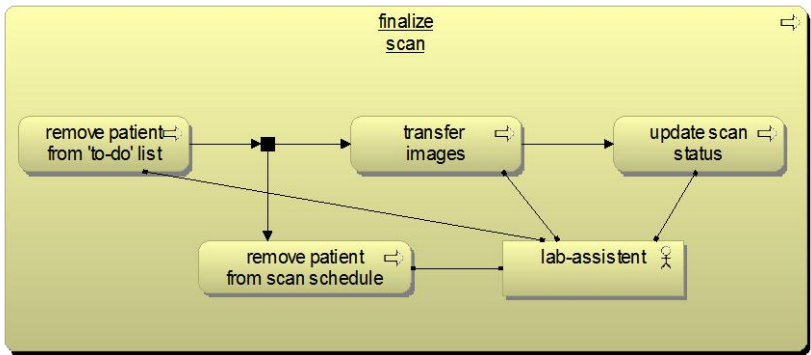


Figure 9-9: Finalize scan process.

C. **Process scan results.** Actors involved are the radiologist, and the secretary.

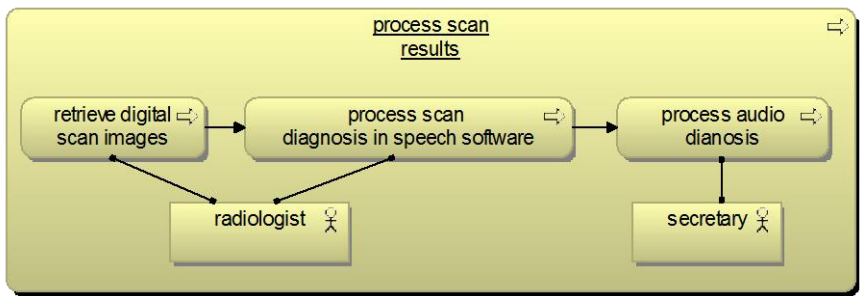
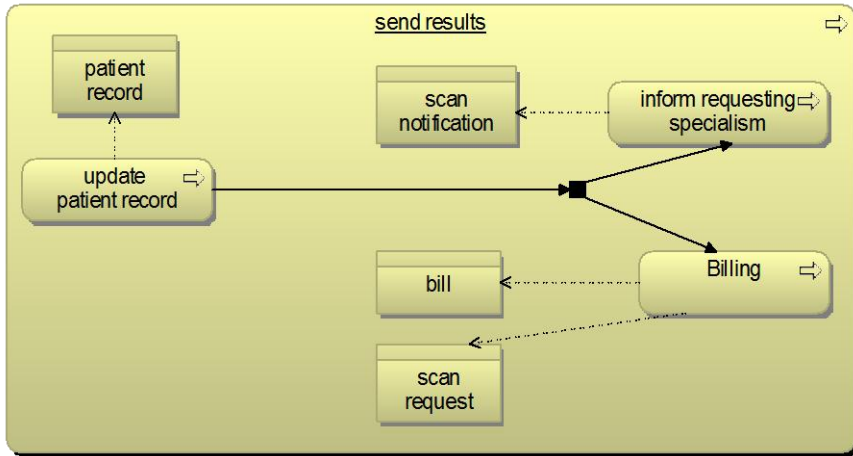


Figure 9-10: Process scan results process.

D. **Send the results.** The actor involved is the secretary.



**Figure 9-11: Send results process.**

This process model depicts the general setting of the treatment of a patient, when send from the neurosurgery to the radiology department.

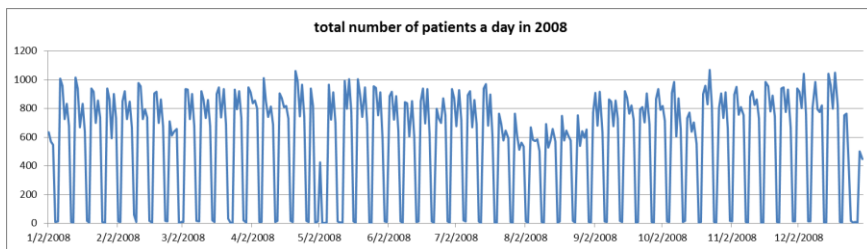
## 9.2 Appendix B: Process description outpatient clinics: Gelre case

Eighteen outpatient clinics are involved in the Gelre case, see Table 9-1. These clinics treat in numbers the majority of the hospital's individual patients yearly.

Anesthesia	Internal medicine	Neurology
Cardiology	Oral surgery	Ophthalmology
Surgery	Otolaryngology	Orthopedic surgery
Dermatology	Pediatrics	Plastic surgery
Geriatrics	Speech and language pathology	Rheumatology
Gynecology	Pulmonology	Physical medicine and rehabilitation

**Table 9-1: Outpatient clinics involved in ISD project.**

Since the future facility is limiting the medical operations in terms of available physical space, the number of patients provides some insights. The total amount of patients (for the outpatient clinics) on a daily basis ranges between 600 and 1000, see Figure 9-12. Figure 9-13 and Figure 9-14 depict respectively the total number of patients on a daily basis for four weeks, and the total number of patients present at any given time on an average day.



**Figure 9-12: Total number of patients for the outpatient clinics.**

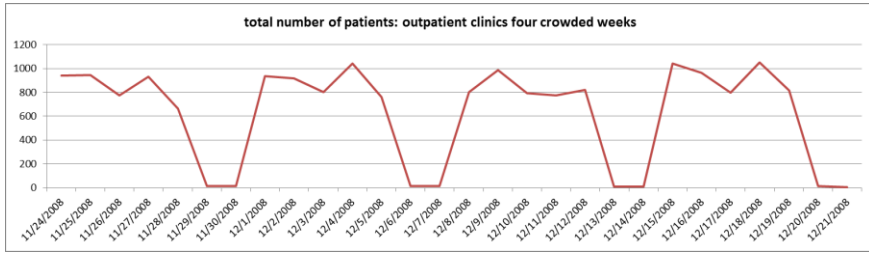


Figure 9-13: Number of patients for the outpatient clinics; 4 weeks.

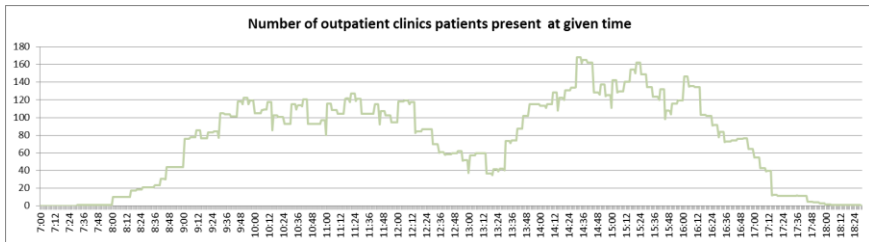


Figure 9-14: Number of patients present in the hospital at any given time.

To calculate the number of patients that are present at any given day on an average day, the total time per patient spend in the hospital is required. This information is acquired by conducting 379 patient interviews, separating per outpatient clinic, whether the treatment starts on time, too early, or too late. Furthermore time of entrance and time of exit are noted. Based on this analysis the ADR team concludes that an average patient spends about 55 minutes within the hospital. From this survey other information can be deducted one of which is not necessary relevant for this research, although very interesting to know: the main cause for treatments being delayed was the physician being late.

During the course of the research many interview are conducted to produce accurate process models of every outpatient clinic. The detailed process models show that every specialism follows its own slightly modified process, adapted to the particular treatment. To provide some idea of a process in the outpatient clinics, the process of otolaryngology is presented since it is very characteristic for outpatient clinics. Furthermore ophthalmology is considered to be

one of the main departments, since it treats per day between 80 and 100 patients. Figure 9-15 presents this process from this clinic's point of view. The various (preferred) roles within the department that process each task are listed in the rounded rectangles. In the MST case, described in chapter 4, the researchers experienced the (lack of) practitioners and end-users process modeling skills. Because advanced modeling is not perceived with much enthusiasm the processes are drawn in this case in a low abstraction level, easily comprehensible fashion.

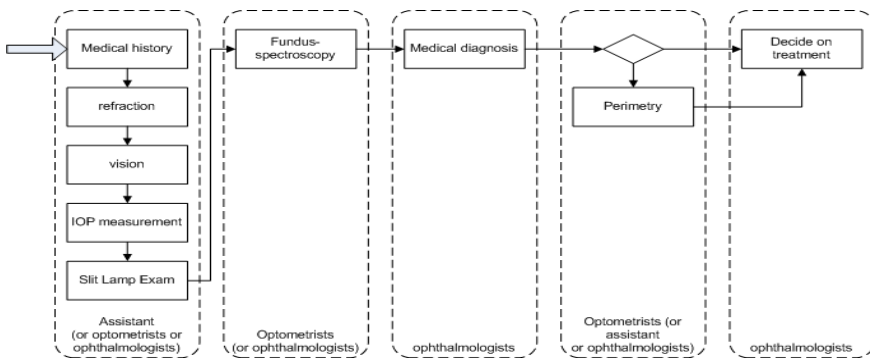


Figure 9-15: Overview process flow ophthalmology.

Figure 9-16 depicts a patient perspective on the ophthalmology process. The main issue for a patient is the many check-ins and waiting area he/she has to visit.

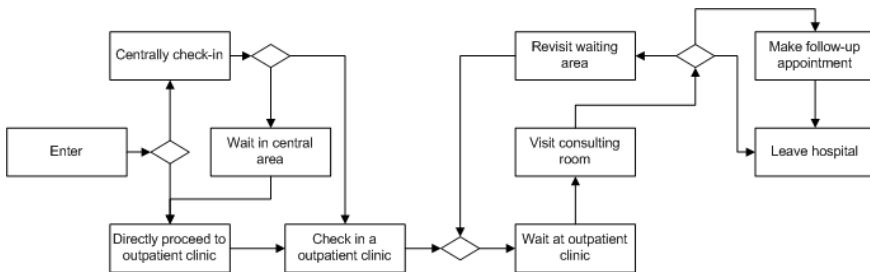


Figure 9-16: Patient perspective on ophthalmology process.

Processes for other outpatient clinics are drawn too, which are used in later ISD development.

## 10 References

- Abrahamsson, P., J. Warsta, M. T. Siponen, and J. Ronkainen, 2003, New directions on agile methods: a comparative analysis: International Conference on Software Engineering, p. 244-254.
- Adams, A., and S. Bond, 2000, Hospital nurses' job satisfaction, individual and organizational characteristics: *Journal of Advanced Nursing*, v. 32, p. 536-543.
- Agarwal, R., and A. P. Sinha, 2003, Object-oriented modeling with UML: a study of developers' perceptions: *Communications of the ACM*, v. 46, p. 256.
- Aken, J. E., 2004, Management research based on the paradigm of the design sciences: The quest for field tested and grounded technological rules: *Journal of management studies*, v. 41, p. 219-246.
- Alvarez, J., and O. Rampnoux, 2007, Serious Game: Just a question of posture?: *Artificial & Ambient Intelligence (AISB'07)*, p. 420-423.
- Anderson, R. A., and R. R. McDaniel Jr, 2000, Managing Health Care Organizations: Where Professionalism Meets Complexity Science: *Health Care Management Review*, v. 25, p. 83.
- Anton, A. I., 1996, Goal-based requirements analysis: Requirements Engineering, 1996., *Proceedings of the Second International Conference on*, p. 136-144.
- Antón, A. I., W. M. McCracken, and C. Potts, 1994, Goal decomposition and scenario analysis in business process reengineering: *Proceedings of the 6th international conference on Advanced information systems engineering table of contents*, p. 94-104.
- Avison, D., and G. Fitzgerald, 2006, Methodologies for Developing Information Systems: An Historical Perspective: *IFIP WCC TC8*, v. 18, p. 08-25.
- Avison, D. E., and G. Fitzgerald, 2003, Where now for development methodologies?: *Communications of the ACM*, v. 46, p. 82.
- Barreteau, O., and G. Abrami, 2007, Variable time scales, agent-based models, and role-playing games: The PIEPLUE river basin management game: *Simulation & Gaming*, v. 38, p. 364.
- Baskerville, R., J. Travis, and D. P. Truex, 1992, Systems without method: The impact of new technologies on information systems development projects: *Proceedings of the IFIP WG8. Working Conference on The Impact of Computer Supported Technologies in Information Systems Development*, p. 241-269.
- Baskerville, R., and A. T. Wood-Harper, 1998, Diversity in information systems action research methods: *European Journal of Information Systems*, v. 7, p. 90-107.
- Benbasat, I., and R. W. Zmud, 1999, Empirical research in information systems: the practice of relevance: *MIS Quarterly*, p. 3-16.
- Benbasat, I., and R. W. Zmud, 2003, The Identity Crisis Within the IS Discipline: Defining and Communicating the Discipline's Core Properties: *MANAGEMENT INFORMATION SYSTEMS QUARTERLY*, v. 27, p. 183-194.
- Berggren, C., 1993, Alternatives to lean production: Work organization in the Swedish auto industry, Cornell University Press.

- Beyer, M., K. A. Kuhn, C. Meiler, S. Jablonski, and R. Lenz, 2004, Towards a flexible, process-oriented IT architecture for an integrated healthcare network: SAC '04 Proceedings of the 2004 ACM symposium on Applied computing, p. 264-271.
- Blair, J. D., M. D. Fottler, G. T. Savage, and C. J. Whitehead, 1990, Challenges in Health Care Management: Strategic Perspectives for Managing Key Stakeholders, Jossey-Bass Publishers.
- Blumberg, B., D. R. Cooper, and P. S. Schindler, 2005, Business research methods, McGraw-Hill.
- Boehm, B., 2002, Get ready for agile methods, with care: COMPUTER, p. 64-69.
- Boehm, B., 2006, A view of 20th and 21st century software engineering: International Conference on Software Engineering, p. 12-29.
- Boehm, B. W., 1988, A spiral model of software development and enhancement: Computer, v. 21, p. 61-72.
- Booch, G., J. Rumbaugh, and I. Jacobson, 2005, Unified Modeling Language User Guide, The (Addison-Wesley Object Technology Series), Addison-Wesley Professional.
- Brancheau, J. C., and C. V. Brown, 1993, The management of end-user computing: status and directions: ACM Computing Surveys (CSUR), v. 25, p. 437-482.
- Braun, C., F. Wortmann, M. Hafner, and R. Winter, 2005, Method construction-a core approach to organizational engineering: ACM symposium on Applied computing, p. 1295-1299.
- Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, 2004, Tropos: An Agent-Oriented Software Development Methodology: Autonomous Agents and Multi-Agent Systems, v. 8, p. 203-236.
- Brinkkemper, S., 1996, Method engineering: engineering of information systems development methods and tools: Information and Software Technology, v. 38, p. 275-280.
- Brinkkemper, S., M. Saeki, and F. Harmsen, 1998, Assembly techniques for method engineering: 10th International Conference, CAiSE'98, p. 381-400.
- Brooks, F. P., 1987, No silver bullet: Essence and accidents of software engineering: IEEE computer, v. 20, p. 10-19.
- Brugha, R., and Z. Varvasovszky, 2000, Stakeholder analysis: a review: Health Policy and Planning, v. 15, p. 239-246.
- Brynjolfsson, E., 1993, The productivity paradox of information technology: Communications of the ACM, v. 36, p. 77.
- Bucher, R., and J. Stelling, 1969, Characteristics of professional organizations: Journal of Health and Social Behavior, v. 10, p. 3-15.
- Bucher, T., M. Klesse, S. Kurpjuweit, and R. Winter, 2007, Situational Method Engineering: Situational Method Engineering: Fundamentals and Experiences, p. 33-48.
- Burns, L. R., and D. R. Wholey, 1993, Adoption and abandonment of matrix management programs: Effects of organizational characteristics and interorganizational networks: The Academy of Management Journal, v. 36, p. 106-138.
- Carr, N. G., 2003, IT doesn't matter: Harvard Business Review, v. 81, p. 41-49.



- Checkland, P., and J. Poulter, 2006, Learning for action: a short definitive account of soft systems methodology and its use, for practitioners, teachers and students.
- Checkland, P. B., 1978, The origins and nature of "hard" systems thinking: *Journal of Applied Systems Analysis*, v. 5, p. 99-110.
- Chen, T. F., and C. Jung, 2009, BUILDING A PLATFORM OF BUSINESS MODEL 2.0 TO CREATING REAL BUSINESS VALUE WITH WEB 2.0 FOR WEB INFORMATION SERVICES INDUSTRY: *International Journal of Electronic Business*, v. 7, p. 168-180.
- Chen, W. S., and R. Hirschheim, 2004, A paradigmatic and methodological examination of information systems research from 1991 to 2001: *Information Systems Journal*, v. 14, p. 197-235.
- Cockburn, A., and J. Highsmith, 2002, Agile software development, the people factor: *Computer*, v. 34, p. 131-133.
- Cohen, D., M. Lindvall, and P. Costa, 2004, An introduction to agile methods: *Advances in Computers*, v. 62, p. 1-66.
- Cole, R., S. Purao, M. Rossi, and M. K. Sein, 2005, Being proactive: where action research meets design research: *The Twenty-Sixth International Conference on Information Systems*, p. 325-336.
- Conboy, K., and B. Fitzgerald, 2004, Toward a conceptual framework of agile methods: *Extreme Programming and Agile Methods-XP/Agile Universe 2004*, p. 19-32.
- Cossentino, M., S. Gaglio, B. Henderson-Sellers, and V. Seidita, 2006, A metamodelling-based approach for method fragment comparison: *International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 06)*, p. 57-70.
- Crawford, L., J. Pollack, and D. England, 2006, Uncovering the trends in project management: *Journal emphases over the last 10 years: International Journal of Project Management*, v. 24, p. 175-184.
- Cron Marion, G., and L. William, 1983, The relationship between computerization and performance: a strategy for maximizing the economic benefits of computerization: *Information & Management*, v. 6, p. 171-181.
- Crosby, B., 1992, *Stakeholder Analysis: A Vital Tool for Strategic Managers*, USAID's Implementing Policy Change Project.
- Curtis, B., M. I. Kellner, and J. Over, 1992, Process modeling: *Communications of the ACM*, v. 35, p. 75-90.
- Cutler, D. M., 2002, Equality, Efficiency, and Market Fundamentals: The Dynamics of International Medical-Care Reform: *Journal of Economic Literature*, v. 40, p. 881-906.
- Czarnecki, K., and U. W. Eisenecker, 2000, *Generative Programming – Methods, Tools, and Applications*, v. 26, Boston: Addison Wesley, 832 p.
- Da Silveira, G., D. Borenstein, and F. S. Fogliatto, 2001, Mass customization: Literature review and research directions: *International journal of production economics*, v. 72, p. 1-13.
- Dardenne, A., V. L. A., and S. Fickas, 1993a, Goal-directed Requirements Acquisition: *Science of Computer Programming*, v. 20, p. 3-50.
- Dardenne, A., S. Fickas, and A. van Lamsweerde, 1991, Goal-directed concept acquisition in requirements elicitation: *International Workshop on Software Specifications & Design*, p. 14-21.

- Dardenne, A., A. van Lamsweerde, and S. Fickas, 1993b, Goal-directed requirements acquisition: Selected Papers of the Sixth International Workshop on Software Specification and Design table of contents, p. 3-50.
- Davenport, T. H., R. J. Thomas, and S. Cantrell, 2002, Knowledge-Worker Performance: MIT Sloan Management Review.
- Davison, R., M. G. Martinsons, and N. Kock, 2004, Principles of canonical action research: *Information Systems Journal*, v. 14, p. 65-86.
- de Sitter, L. U., J. F. den Hertog, and B. Dankbaar, 1997, From complex organizations with simple jobs to simple organizations with complex jobs: *Human Relations*, v. 50, p. 497.
- Deneckere, R., A. Iacovelli, E. Kornysheva, and C. Souveyet, 2008, From method fragments to method services: 13th International Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'08), p. 80-96.
- Denning, P. J., 1997, A new social contract for research: *Communications of the ACM*, v. 40, p. 134.
- Dobing, B., and J. Parsons, 2006, How UML is used: *Communications of the ACM*, v. 49, p. 109-113.
- Donaldson, L., 2001, *The contingency theory of organizations*, Sage Publications, Inc.
- Dooley, K. J., 1997, A Complex Adaptive Systems Model of Organization Change: *Nonlinear Dynamics, Psychology, and Life Sciences*, v. 1, p. 69-97.
- Drucker, P. F., 1999, Knowledge-worker productivity: the biggest challenge: *California Management Review*, v. 41, p. 79-94.
- Drucker, P. F., 2000, Implementing the Effective Management of Knowledge Worker Productivity: The Biggest Challenge: *The knowledge management yearbook 2000-2001*, p. 267.
- Dubé, L., and G. Paré, 2003, Rigor in information systems positivist case research: Current practices, trends, and recommendations: *Mis Quarterly*, p. 597-636.
- Eisenhardt, K. M., 1989, Building theories from case study research: *Academy of management review*, p. 532-550.
- Enthoven, S. A. C., 1980, *Health Plan: The Only Practical Solution to Soaring Health Care Costs*, Reading, Mass.: Addison-Wesley.
- Feiler, P. H., and W. S. Humphrey, 1992, Software process development and enactment: concepts and definitions, p. 28-40.
- Feinstein, A. H., S. Mann, and D. L. Corsun, 2002, Charting the experiential territory: Clarifying definitions and uses of computer simulation, games, and role play: *Journal of Management Development*, v. 21, p. 732-744.
- Feldstein, M. S., 1974, *ECONOMETRIC STUDIES OF HEALTH ECONOMICS: Frontiers of Quantitative Economics: papers invited for presentation at the Econometric Society Winter Meetings, Toronto, 1972*, p. 377.
- Fischer, G., E. Giaccardi, Y. Ye, A. G. Sutcliffe, and N. Mehandjiev, 2004, Meta-design: a manifesto for end-user development: *Communications of the ACM*, v. 47, p. 33-37.
- Fitzgerald, B., 2000, Systems development methodologies: the problem of tenses: *Information Technology & People*, v. 13, p. 174-185.
- Freeman, R. E., 1984, *Strategic Management: A Stakeholder Approach*.

- Fujimoto, M., K. Miyazaki, and N. Von Tunzelmann, 2000, Complex systems in technology and policy: telemedicine and telecare in Japan: *Journal of telemedicine and telecare*, v. 6, p. 187.
- Fuxman, A., L. Liu, J. Mylopoulos, M. Pistore, M. Roveri, and P. Traverso, 2004, Specifying and analyzing early requirements in Tropos: *Requirements Engineering*, v. 9, p. 132-150.
- Gell-Mann, M., and S. Lloyd, 1996, Information measures, effective complexity, and total information: *Complexity*, v. 2, p. 44-52.
- Gero, J. S., 1996, Creativity, emergence and evolution in design: *Knowledge-Based Systems*, v. 9, p. 435-448.
- Gifford, B. D., R. F. Zammuto, and E. A. Goodman, 2002, The relationship between hospital unit culture and nurses' quality of work life: *Journal of Healthcare Management/American College of Healthcare Executives*, v. 47, p. 13.
- Ginzberg, M. J., 1980, An organizational contingencies view of accounting and information systems implementation\* 1: *Accounting, Organizations and Society*, v. 5, p. 369-382.
- Giorgini, P., S. Rizzi, and M. Garzetti, 2008, GRAnD: A goal-oriented approach to requirement analysis in data warehouses: *Decision Support Systems*, v. 45, p. 4-21.
- Giunchiglia, F., J. Mylopoulos, and A. Perini, The Tropos Software Development Methodology: Processes: Models and Diagrams. Technical Report.
- Giunchiglia, F., J. Mylopoulos, and A. Perini, 2002, The Tropos Software Development Methodology: Processes: Models and Diagrams. Technical Report.
- Glaser, B. G., and A. L. Strauss, 2007, *The discovery of grounded theory: Strategies for qualitative research*, Aldine Transaction.
- Glass, R. L., 2003, *Facts and fallacies of software engineering*, Addison-Wesley Professional.
- Glouberman, S., and H. Mintzberg, 2001a, Managing the care of health and the cure of disease-Part I: Differentiation: *Health Care Management Review*, v. 26, p. 56.
- Glouberman, S., and H. Mintzberg, 2001b, Managing the care of health and the cure of disease-Part II: Integration: *Health care management review*, v. 26, p. 70.
- Grandi, A., and R. Grimaldi, 2005, Academics' organizational characteristics and the generation of successful business ideas: *Journal of Business Venturing*, v. 20, p. 821-845.
- Greenbaum, J. M., and M. Kyng, 1991, *Design at work: Cooperative design of computer systems*, L. Erlbaum Associates Inc. Hillsdale, NJ, USA.
- Greenwood, R., S. X. Li, R. Prakash, and D. L. Deephouse, 2005, Reputation, diversification, and organizational explanations of performance in professional service firms: *Organization Science*, v. 16, p. 661-673.
- Gregor, S., 2006, The nature of theory in information systems: *Management Information Systems Quarterly*, v. 30, p. 611.
- Hafferty, F. W., and B. Castellani, 2010, The Increasing Complexities of Professionalism: *Academic Medicine*, v. 85, p. 288.
- Hammer, M., 1990, Reengineering work: don't automate, obliterate: *Harvard business review*, v. 68, p. 104-112.

- Han, T., S. Puro, and V. C. Storey, 2008, Generating large-scale repositories of reusable artifacts for conceptual design of information systems: *Decision Support Systems*, v. 45, p. 665-680.
- Harmsen, A. F., 1997, *Situational method engineering*, University of Twente, Enschede.
- Harmsen, A. F., J. N. Brinkkemper, and J. L. H. Oei, 1994a, *Situational method engineering for information system project approaches*, University of Twente, Department of Computer Science.
- Harmsen, F., S. Brinkkemper, and H. Oei, 1994b, *Situational method engineering for informational system project approaches: Methods and Associated Tools for the Information Systems Life Cycle*, p. 169-194.
- Harper, P. R., and A. K. Shahani, 2002, Modelling for the planning and management of bed capacities in hospitals: *Journal of the Operational Research Society*, v. 53, p. 11-18.
- Harris, J. E., 1977, The internal organization of hospitals: some economic implications: *The Bell Journal of Economics*, v. 8, p. 467-482.
- Henderson-Sellers, B., 2005, Creating a comprehensive agent-oriented methodology-using method engineering and the OPEN metamodel, *in* P. G. B. Henderson-Sellers, Eds. *Idea Group, ed., Agent-Oriented Methodologies* Hershey, PA, Idea Group, p. 368-397
- Henderson-Sellers, B., D. Lowe, and B. Haire, 2002, OPEN process support for web development: *Annals of software Engineering*, v. 13, p. 163-201.
- Henderson-Sellers, B., and J. Ralyté, 2010, *Situational Method Engineering: State-of-the-Art Review*: *Journal of Universal Computer Science*, v. 16, p. 424-478.
- Hevner, A. R., S. T. March, J. Park, and S. Ram, 2004, *Design Science in Information Systems Research: MANAGEMENT INFORMATION SYSTEMS QUARTERLY*, v. 28, p. 75-106.
- Highsmith, J., and A. Cockburn, 2002, *Agile software development: The business of innovation*: *Computer*, v. 34, p. 120-127.
- Hirschheim, H. K., 1996, Exploring the intellectual structures of information systems development: a social action theoretic analysis: *Accounting, Management and Information Technologies*, v. 6, p. 1-64.
- Hirschheim, R., and H. K. Klein, 1989, Four paradigms of information systems development: *Communications of the ACM*, v. 32, p. 1199-1216.
- Hirschheim, R. A., 1983, Assessing participative systems design: Some conclusions from an exploratory study\* 1: *Information & Management*, v. 6, p. 317-327.
- Hofferberth, R. D., 2002, *Professional services automation 2002: PSA research report summary findings*, Boston, MA., Aberdeen Group.
- Holland, J. H., 1996, *Hidden Order: How Adaptation Builds Complexity*, Addison Wesley Publishing Company.
- Hume, D., 1978, *A Treatise of Human Nature*, ed. PH Nidditch, Oxford: Clarendon Press.
- Håkanson, L., and R. Nobel, 2001, Organizational characteristics and reverse technology transfer: *MIR: Management International Review*, v. 41, p. 395-420.

- Iacob, M. E., D. Rothengatter, and J. Hillegersberg, 2009, A Health-care Application of Goal-driven Software Design: Applied Medical Informatics, v. 24, p. 12-33.
- Iivari, J., 2007, Nothing is as Clear as Unclear: Iivari's response: SCANDINAVIAN JOURNAL OF INFORMATION SYSTEMS, v. 19, p. 111.
- Iivari, J., R. Hirschheim, and H. K. Klein, 1998, A paradigmatic analysis contrasting information systems development approaches and methodologies: Information Systems Research, v. 9, p. 164-193.
- Juarrero, A., 2002, Dynamics in action: Intentional behavior as a complex system, The MIT Press.
- Katsma, C. P., 2008, An organizational change approach for enterprise system implementations: Enschede, The Netherlands, University of Twente.
- Keller, S. E., L. G. Kahn, and R. B. Panara, 1990, Specifying Software Quality Requirements with Metrics: System and Software Requirements Engineering, p. 145-163.
- Ketchum, L. D., and E. L. Trist, 1992, All teams are not created equal: How employee empowerment really works, Sage Publications, Inc.
- Kiely, G., and B. Fitzgerald, 2005, An investigation of the use of methods within information systems development projects: The Electronic Journal of Information Systems in Developing Countries, v. 22.
- Klaus, H., M. Rosemann, and G. G. Gable, 2000, What is ERP?: Information Systems Frontiers, v. 2, p. 141-162.
- Kobayashi, M., S. R. Fussell, Y. Xiao, and F. J. Seagull, 2005, Work coordination, workflow, and workarounds in a medical context: Conference on Human Factors in Computing Systems, p. 1561-1564.
- Kobryn, C., 1999, UML 2001: a standardization odyssey: Communications of the ACM, v. 42, p. 29-37.
- Kornysheva, E., R. Deneckere, and C. Salinesi, 2007, Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach: Proceedings of the IFIP WG 8.1 Working Conference: Situational Method Engineering: Fundamentals and Experiences, p. 64-78.
- Kroenke, D., 2011, Using mis, Prentice Hall.
- Krueger, C. W., 2005, Software mass customization: BigLever Software, Inc.
- Krueger, C. W., 2006, New methods in software product line practice: Communications of the ACM, v. 49, p. 37-40.
- Kueng, P., and P. Kawalek, 1997, Goal-based business process models: creation and evaluation: Journal, Vol, v. 3, p. 17-38.
- Kumar, K., and J. Van Hillegersberg, 2000, ERP experiences and evolution: Communications of the ACM, v. 43, p. 23-26.
- Lansing, J. S., 2003, Complex Adaptive Systems: Annual Reviews in Anthropology, v. 32, p. 183-204.
- Lapouchnian, A., 2005, Goal-oriented requirements engineering: An overview of the current research, Toronto, University of Toronto.
- Leavitt, H. J., 1978, Managerial psychology: an introduction to individuals, pairs, and groups in organizations, University of Chicago Press.
- Lee, A. S., 1999, Rigor and relevance in MIS research: beyond the approach of positivism alone: MIS Quarterly, p. 29-33.

- Lee, A. S., and R. L. Baskerville, 2003, Generalizing generalizability in information systems research: *Information systems research*, v. 14, p. 221-243.
- Lindley, C. A., 2005, Game space design foundations for trans-reality games, p. 397-404.
- Lyytinen, K., 1987, A taxonomic perspective of information systems development: theoretical constructs and recommendations, *Critical issues in information systems research*, John Wiley & Sons, Inc., p. 3-41.
- Malone, T. W., and S. A. Smith, 1988, Modeling the performance of organizational structures: *Operations Research*, v. 36, p. 421-436.
- March, S. T., and G. F. Smith, 1995, Design and natural science research on information technology: *Decision Support Systems*, v. 15, p. 251-266.
- Markus, M. L., A. Majchrzak, and G. Les, 2002, A design theory for systems that support emergent knowledge processes: *Mis Quarterly*, p. 179-212.
- McConnell, S., 1993, *Code complete: a practical handbook of software construction*, v. 1, Microsoft Press, Books24x, Inc.
- Meadows, D. L., 2001, Tools for understanding the limits to growth: Comparing a simulation and a game: *Simulation & Gaming*, v. 32, p. 522.
- Mellor, S., K. Scott, A. Uhl, and D. Weise, 2002, Model-driven architecture: *Advances in Object-Oriented Information Systems*, p. 233-239.
- Millican, P. J. R., 1995, *Hume's Argument Concerning Induction: Structure and Interpretation: David Hume: Critical Assessments*, London and New York: Routledge, v. 2, p. 99-144.
- Mintzberg, H., 1998, Covert leadership: Notes on managing professionals: *Harvard Business Review*, v. 76, p. 140-148.
- Mirbel, I., and J. Ralyté, 2006, Situational method engineering: combining assembly-based and roadmap-driven approaches: *Requirements Engineering*, v. 11, p. 58-78.
- Moody, D., and J. Van Hillegersberg, 2009, Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams: *Software Language Engineering*, p. 16-34.
- Morandini, M., L. Penserini, and A. Perini, 2008, Automated Mapping from Goal Models to Self-Adaptive Systems, p. 485-486.
- Morgan, G., 1998, *Images of organization*: San Fransisco, Sage Publications.
- Mostow, J., 1985, Toward Better Models of the Design Process: *AI Magazine*, v. 6, p. 44-57.
- Myers, M. D., and H. K. Klein, 2011, A set of principles for conducting critical research in information systems: *Management Information Systems Quarterly*, v. 35, p. 17-39.
- Mylopoulos, J., L. Chung, and E. Yu, 1999, From object-oriented to goal-oriented requirements analysis: *Communications of the Acm*, v. 42, p. 31-37.
- Newhouse, J. P., 1993, An Iconoclastic View Of Health Cost Containment, *Health Affairs*, p. 152-171.
- Nielsen, J., 2002, The usability engineering life cycle: *Computer*, v. 25, p. 12-22.
- Osterwalder, A., and Y. Pigneur, 2002, An e-business model ontology for modeling e-business: *Bled*, p. 17-19.

- Osterwalder, A., and Y. Pigneur, 2010, *Business model generation: A handbook for visionaries, game changers, and challengers*, Wiley.
- Ostrowski, L., and M. Helfert, 2011, Commonality in various design science methodologies: *Computer Science and Information Systems (FedCSIS)*, 2011 Federated Conference on, p. 317-320.
- Panko, R. R., 1998, What we know about spreadsheet errors: *Journal of End User Computing*, v. 10, p. 15-21.
- Penserini, L., A. Perini, A. Susi, and J. Mylopoulos, 2007, High variability design for software agents: *Extending Tropos*.
- Perini, A., and A. Susi, 2004, Developing Tools for Agent-Oriented Visual Modeling: *LECTURE NOTES IN COMPUTER SCIENCE*, p. 169-182.
- Peters, T. J., and R. H. Waterman, 2004, *In search of excellence: Lessons from America's best-run companies*, Harper Paperbacks.
- Pham, D. N., and A. Klinkert, 2008, Surgical case scheduling as a generalized job shop scheduling problem: *European Journal of Operational Research*, v. 185, p. 1011-1025.
- Potts, C., 1997, Fitness for use: the system quality that matters most: *Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ*, p. 15-28.
- Propper, C., M. Sutton, C. Whitnall, and F. Windmeijer, 2008, Did 'Targets and Terror' Reduce Waiting Times in England for Hospital Care?: *HEALTH CARE*, v. 8, p. 5.
- Punter, T., and K. Lemmen, 1996, The MEMA-model: towards a new approach for Method Engineering: *Information and Software Technology*, v. 38, p. 295-305.
- Quinn, R., and G. Spreitzer, 1991, The psychometric of the competing values culture instrument and an analysis of the impact of organizational culture on quality of life: *Research in organizational change and development*, 5 (??), 115-142.
- Quinn, R. E., and J. Rohrbaugh, 1983, A spatial model of effectiveness criteria: Towards a competing values approach to organizational analysis: *Management science*, v. 29, p. 363-377.
- Rakitin, S., 2001, Manifesto elicits cynicism: *IEEE Computer*, v. 34, p. 4.
- Ralyté, J., 2004, Towards situational methods for information systems development: engineering reusable method chunks: *International Conference on Information Systems Development (ISD'04): Advances in Theory, Practice and Education* p. 271-282.
- Ralyté, J., R. Deneckère, and C. Rolland, 2003, Towards a generic model for situational method engineering: *15th International Conference, CAiSE* p. 1029-1029.
- Ralyté, J., and C. Rolland, 2001a, An approach for method reengineering: *International Conference on Conceptual Modelling (ER2001)*, p. 471-484.
- Ralyté, J., and C. Rolland, 2001b, An assembly process model for method engineering, *in* A. G. K.R. Dittrich, M.C. Norrie, ed., *Advanced Information Systems Engineering: Berlin, Springer*, p. 267-283.
- Rao, A. S., and M. P. Georgeff, 1995, BDI agents: From theory to practice, p. 312-319.
- Raymond, L., 1985, Organizational characteristics and MIS success in the context of small business: *Mis Quarterly*, p. 37-52.

- Ren, Y., S. Kiesler, and S. R. Fussell, 2008, Multiple Group Coordination in Complex and Dynamic Task Environments: Interruptions, Coping Mechanisms, and Technology Recommendations: *Journal of Management Information Systems*, v. 25, p. 105 - 130.
- Rettig, C., 2007, The trouble with enterprise software: *MIT Sloan Management Review*, v. 49, p. 21.
- Richardson, H., and B. Robinson, 2007, The mysterious case of the missing paradigm: a review of critical information systems research 1991–2001: *Information Systems Journal*, v. 17, p. 251-270.
- Rolland, C., and N. Prakash, 1996, A proposal for context-specific method engineering: *IFIP TC8, WG8*, p. 191-208.
- Rolland, C., N. Prakash, and A. Benjamen, 1999, A multi-model view of process modelling: *Requirements Engineering*, v. 4, p. 169-187.
- Rolland, C., C. Souveyet, and A. Ben, 1998, Guiding Goal Modelling using Scenarios: *IEEE Transactions on Software Engineering*, Special Issue on Scenario Management, v. 24, p. 1055-1071.
- Rossi, M., B. Ramesh, K. Lyytinen, and J. P. Tolvanen, 2004, Managing evolutionary method engineering by method rationale: *Journal of the Association for Information Systems*, v. 5, p. 12.
- Roth, A. V., and R. V. Dierdonck, 1995, Hospital Resource Planning: Concepts, Feasibility, and Framework: *Production and Operations Management*, v. 4, p. 2-29.
- Rothengatter, D., C. Katsma, and J. Hillegersberg, 2010, Simulation As a Method To Support Complex Organizational Transformations in Healthcare, Sixteenth Americas Conference on Information Systems, Lima, Peru, AIS.
- Royce, W. W., 1987, Managing the development of large software systems: concepts and techniques: *International Conference on Software Engineering*, p. 328-338.
- Rubinstein, R. Y., and D. P. Kroese, 2008, *Simulation and the Monte Carlo method*, Wiley-interscience.
- Sargent, R. G., 2005, Verification and validation of simulation models, p. 143.
- Schendel, D., and C. W. Hofer, 1979, *Strategic management: A new view of business policy and planning*: Boston, Little, Brown.
- Scherer, E., and M. Zölch, 1995, Design of activities in shopfloor management: a holistic approach to organisation at operational business levels in BPR projects: *Re-engineering the Enterprise*, Proceedings of the IFIP TC5/WG5, v. 7, p. 20-21.
- Schoonhoven, C. B., 1981, Problems with contingency theory: testing assumptions hidden within the language of contingency" theory": *Administrative Science Quarterly*, v. 26, p. 349-377.
- Schwartz, R. B., and M. C. Russo, 2004, How to quickly find articles in the top IS journals: *Communications of the ACM*, v. 47, p. 98-101.
- Scott, W. R., 2002, *Organizations: Rational, natural, and open systems*, Prentice hall.
- Seagull, F. J., P. Cheryl, Y. Xiao, and C. F. Mackenzie, 2003, Collaborative Management of Complex Coordination Systems: Operating Room Schedule Coordination.
- Sefelin, R., M. Tscheligi, and V. Giller, 2003, Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping: *CHI '03 extended abstracts on Human factors in computing systems*, p. 778-779.



- Sein, M., O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, 2010, Action Design Research: *MIS Quarterly*.
- Shapiro, D., 2005, Participatory design: the will to succeed, p. 29-38.
- Shenhar, A. J., 2001, One size does not fit all projects: Exploring classical contingency domains: *Management Science*, v. 47, p. 394-414.
- Shenhar, A. J., and D. Dvir, 1996, Toward a typological theory of project management: *Research Policy*, v. 25, p. 607-632.
- Shortell, S. M., J. L. O'Brien, J. M. Carman, R. W. Foster, E. F. Hughes, H. Boerstler, and E. J. O'Connor, 1995, Assessing the impact of continuous quality improvement/total quality management: concept versus implementation: *Health services research*, v. 30, p. 377.
- Simon, H. A., 1964, On the concept of organizational goal: *Administrative Science Quarterly*, v. 9, p. 1-22.
- Simonsen, J., and M. Hertzum, 2008, Participative design and the challenges of large-scale systems: Extending the iterative PD approach: *Conference on Participatory Design*, p. 1-10.
- Smith, K. G., E. A. Locke, and D. Barry, 1990, Goal setting, planning, and organizational performance: An experimental simulation\* 1: *Organizational Behavior and Human Decision Processes*, v. 46, p. 118-134.
- Snyder, C., 2003, Paper prototyping: The fast and easy way to design and refine user interfaces, Morgan Kaufmann Pub.
- Soffer, P., and Y. Wand, 2004, Goal-driven analysis of process model validity, *Advanced Information Systems Engineering, Proceedings: Biomedical Sciences Instrumentation*, v. 3084, p. 521-535.
- Soffer, P., and Y. Wand, 2005, On the notion of soft-goals in business process modeling.
- Soh, C., S. S. Kien, and J. Tay-Yap, 2000, Enterprise resource planning: cultural fits and misfits: is ERP a universal solution?: *Communications of the ACM*, v. 43, p. 47-51.
- Stabell, C. B., and O. D. Fjeldstad, 1998, Configuring value for competitive advantage: On chains, shops, and networks: *Strategic Management Journal*, v. 19, p. 413-437.
- Staples, M., and M. Niazi, 2008, Systematic review of organizational motivations for adopting CMM-based SPI: *Information and Software Technology*, v. 50, p. 605-620.
- Staples, M., M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, 2007, An exploratory study of why organizations do not adopt CMMI: *Journal of Systems and Software*, v. 80, p. 883-895.
- Stiglic, G., and P. Kokol, 2005a, Patient and staff scheduling multi-agent system: *IEEE 3rd International Conference on Computational Cybernetics*, 2005. *ICCC* p. 25-28.
- Stiglic, G., and P. Kokol, 2005b, Patient and staff scheduling multi-agent system, p. 25-28.
- Strogatz, S., 2007, The end of insight: Brockman, John, What is your dangerous idea.
- Susman, G. I., and R. D. Evered, 1978, An assessment of the scientific merits of action research: *Administrative science quarterly*, p. 582-603.
- Sutcliffe, A., and N. Mehandjiev, 2004, End-user development: *Communications of the ACM*, v. 47, p. 31-32.
- Szilagyi, A. D., M. J. Wallace, and J. M. Ivancevich, 1983, Organizational behavior and performance, Scott, Foresman.

- Szmankiewicz, J., K. Turner, and J. McDonald, 1988, Solving business problems by simulation, McGraw-Hill, Inc. New York, NY, USA.
- Team, C. P., 2006, CMMI for Development, version 1.2: Preface, Software Engineering Institute, Carnegie Mellon University, August.
- Thomas, K. W., and W. G. Tymon Jr, 1982, Necessary properties of relevant research: Lessons from recent criticisms of the organizational sciences: *Academy of Management Review*, p. 345-352.
- Tolvanen, J. P., 1998, Incremental method engineering with modeling tools: theoretical principles and empirical evidence, University of Jyväskylä.
- Trist, E. L., and K. W. Bamforth, 1951, Some social and psychological consequences of the Longwall method of coal-getting: *Human relations*.
- Tsjernikova, II, 2009, Developing and investigating validity of a knowledge management game simulation model, University of Twente.
- Umble, E. J., R. R. Haft, and M. M. Umble, 2003, Enterprise resource planning: Implementation procedures and critical success factors: *European Journal of Operational Research*, v. 146, p. 241-257.
- van de Weerd, I., S. Brinkkemper, and J. Versendaal, 2010, Incremental method evolution in global software product management: A retrospective case study: *Information and Software Technology*, v. 52, p. 720-732.
- van de Weerd, I., J. Souer, J. Versendaal, and S. Brinkkemper, 2005, Situational Requirements Engineering of Web Content Management Implementations: SREP2005, p. 13-30.
- Van Der Aalst, W. M. P., H. A. Reijers, A. Weijters, B. F. Van Dongen, A. K. Alves de Medeiros, M. Song, and H. M. W. Verbeek, 2007, Business process mining: An industrial application: *Information Systems*, v. 32, p. 713-732.
- van der Aalst, W. M. P., and A. Weijters, 2004, Process mining: a research agenda: *Computers in Industry*, v. 53, p. 231-244.
- Van der Zwaan, A. H., and J. De Vries, 2000, A critical assessment of the modern sociotechnical approach within production and operations management: *International Journal of Production Research*, v. 38, p. 1755-1767.
- van Lamsweerde, A., 2000, Requirements engineering in the year 00: a research perspective: 22nd international conference on Software engineering, p. 5-19.
- van Lamsweerde, A., 2004, Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice: 12th IEEE Joint International Requirements Engineering Conference, p. 4-8.
- van Lamsweerde, A., 2005, Goal-Oriented Requirements Engineering: A Guided Tour: 5th IEEE International Symposium on Requirements Engineering, p. 249.
- van Lamsweerde, A., and E. Letier, 2004, From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering: LECTURE NOTES IN COMPUTER SCIENCE, p. 325-340.
- van Lamsweerde, A., and L. Willemet, 1998, Inferring Declarative Requirements Specifications from Operational Scenarios: *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, p. 1089-1114.

- van Merode, G. G., S. Groothuis, and A. Hasman, 2004, Enterprise resource planning for hospitals: *International Journal of Medical Informatics*, v. 73, p. 493-501.
- van Slooten, K., and B. Hodes, 1996, Characterizing IS development projects: IFIP TC8, WG8. 1/8.2 Working Conference on Method Engineering: Method engineering: principles of method construction and tool support, p. 29.
- Vandaele, N., and L. De Boeck, 2003, Advanced resource planning: *Robotics and Computer-Integrated Manufacturing*, v. 19, p. 211-218.
- Varvasovszky, Z., and R. Brugha, 2000, How to do (or not to do)... A stakeholder analysis: *Health Policy and Planning*, v. 15, p. 338-345.
- Vos, L., S. Groothuis, and G. G. van Merode, 2007, Evaluating hospital design from an operations management perspective: *Health Care Management Science*, v. 10, p. 357-364.
- Waldrop, M. M., 1993, *Complexity: The Emerging Science at the Edge of Order and Chaos*, Simon & Schuster.
- Walsham, G., 1995, The emergence of interpretivism in IS research: *Information systems research*, v. 6, p. 376-394.
- Wand, Y., 1996, Ontology as a foundation for meta-modelling and method engineering: *Information and Software Technology*, v. 38, p. 281-287.
- Wang, P., and E. B. Swanson, 2007, Launching professional services automation: Institutional entrepreneurship for information technology innovations: *Information and Organization*, v. 17, p. 59-88.
- Wastell, D. G., 1996, The fetish of technique: methodology as a social defence: *Information Systems Journal*, v. 6, p. 25-40.
- Webster, J., and R. T. Watson, 2002, Analyzing the past to prepare for the future: Writing a literature review, *The Society for Information Management and The Management Information Systems Research Center of the University of Minnesota*, an, p. 13-23.
- Weill, P., and M. H. Olson, 1989, An assessment of the contingency theory of management information systems: *Journal of Management Information Systems*, p. 59-85.
- Welke, R. J., 1981, IS/DSS: DBMS support for information systems development: *Data base management--theory and applications: proceedings of the NATO Advanced Study Institute*, p. 195.
- Wenzler, I., and D. Chartier, 1999, Why do we bother with games and simulations: An organizational learning perspective: *Simulation & Gaming*, v. 30, p. 375.
- Wieringa, R., 2009, *Design Science as Nested Problem Solving*, Malvern, PA, USA, ACM.
- Wieringa, R., and A. Morali, 2012, Technical Action Research as a Validation Method in Information Systems Design Science, *in* K. Peffers, R. M., and B. Kuechler, eds., *DÉSRIST 2012*, v. 7286, Springer, Heidelberg, p. 220-238.
- Wilson, B., 1990, *Systems: concepts, methodologies, and applications*, John Wiley & Sons, Inc. New York, NY, USA.
- Yang, D., and S. S. Zhang, 2003, Goal-oriented analysis and agent-based design of agile supply chain, *Web Technologies and Applications: Lecture Notes in Computer Science*, v. 2642, p. 347-356.
- Yap, C. S., 1990, Distinguishing characteristics of organizations using computers: *Information & Management*, v. 18, p. 97-107.

- Yin, R. K., 2009, *Case study research: Design and methods*, v. 5, Sage Publications, Inc.
- Yu, E. S. K., 2004, What does it mean to say that a specification is complete?, IWSSD-4, Fourth International Workshop on Software Specification and Design.
- Yu, E. S. K., 1995, *Modelling strategic relationships for process reengineering*, University of Toronto Toronto, Ont., Canada, Canada.
- Yu, E. S. K., and J. Mylopoulos, 1994a, Understanding "Why" in Software Process, Modelling, Analysis, and Design: Proc. of the 16th International Conference on Software Engineering ' ICSE, p. 16-21.
- Yu, E. S. K., and J. Mylopoulos, 1994b, Using goals, rules, and methods to support reasoning in businessprocess reengineering: System Sciences, 1994. Vol. IV: Information Systems: Collaboration Technology Organizational Systems and Technology, Proceedings of the Twenty-Seventh Hawaii International Conference on, v. 4.
- Zammuto, R. F., and J. Y. Krakower, 1991, Quantitative and qualitative studies of organizational culture: Research in organizational change and development, 5 (??), 83-114.
- Ågerfalk, P., S. Brinkkemper, C. Gonzalez-Perez, B. Henderson-Sellers, F. Karlsson, S. Kelly, and J. Ralyté, 2007, Modularization Constructs in Method Engineering: Towards Common Ground?: Situational Method Engineering: Fundamentals and Experiences, p. 359-368.

## 11 About the Author

Diederik Rothengatter was born on May 1, 1981, in Apeldoorn, the Netherlands. In 2007 he obtained his Master's degree in Business Information Technology from the University of Twente. On his final Master's project he researched corporate performance management (CPM) and the IT systems enabling CPM. During this time he became especially interested in intersection of business and IT.

Shortly after graduation, Diederik Rothengatter shifted his focus to the methodological aspects of information system development. He conducted research in multiple hospitals where he developed and implemented new methods for custom tailored systems. This doctoral dissertation is one of the products of these studies, which was executed under supervision of prof. dr. J. van Hillegersberg. During his PhD research Diederik participated in research projects in the Centre for Telematics and Information Technology of the University of Twente.

Currently, Diederik Rothengatter is employed as a consultant on IT strategy at Deloitte. Diederik specializes in IT and organizational development, business IT alignment, IT governance, and process reengineering. In the future he hopes to make valuable contributions to research, development, and implementation of digital transformations, worldwide.

