



Stockholm  
University

DSV Report Series No. 16-014

# Aspect-Oriented Business Process Management

Amin Jalali





# Aspect-Oriented Business Process Management

**Amin Jalali**

Academic dissertation for the Degree of Doctor of Philosophy in Computer and Systems Sciences at Stockholm University to be publicly defended on Monday 19 December 2016 at 13.00 in L50, NOD-huset, Borgarfjordsgatan 12.

## **Abstract**

Separation of concerns has long been considered an effective and efficient strategy to deal with complexity in information systems. One sort of concern, like security and privacy, crosses over other concerns in a system. Such concerns are called cross-cutting concerns. As a result, the realization of these concerns is scattered through the whole system, which makes their management difficult.

Aspect Orientation is a paradigm in information systems which aims to modularize cross-cutting concerns. This paradigm is well researched in the programming area, where many aspect-oriented programming languages have been developed, e.g., AspectJ. It has also been investigated in other areas, such as requirement engineering and service composition. In the Business Process Management (BPM) area, Aspect Oriented Business Process Modeling aims to specify how this modularization technique can support encapsulating cross-cutting concerns in process models. However, it is not clear how these models should be supported in the whole BPM lifecycle. In addition, the support for designing these models has only been limited to imperative process models that support rigid business processes. Neither has it been investigated how this modularization technique can be supported through declarative or hybrid models to support the separation of cross-cutting concerns for flexible business processes.

Therefore, this thesis investigates how aspect orientation can be supported over the whole BPM lifecycle using imperative aspect-oriented business process models. It also investigates how declarative and hybrid aspect-oriented business process models can support the separation of cross-cutting concerns in the BPM area. This thesis has been carried out following the design science framework, and the result is presented as a set of artifacts (in the form of constructs, models, methods, and instantiations) and empirical findings.

The artifacts support modeling, analysis, implementation/configuration, enactment, monitoring, adjustment, and mining cross-cutting concerns while supporting business processes using Business Process Management Systems. Thus, it covers the support for the management of these concerns over the whole BPM lifecycle. The use of these artifacts and their application shows that they can reduce the complexity of process models by separating different concerns.

Stockholm 2016

<http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-135317>

ISBN 978-91-7649-587-2

ISBN 978-91-7649-588-9

ISSN 1101-8526

**Department of Computer and Systems Sciences**



Stockholm University, 164 07 Kista





# Aspect-Oriented Business Process Management

Amin Jalali

©Amin Jalali, Stockholm University 2016

ISBN (printed version): 978-91-7649-587-2

ISBN (digital version): 978-91-7649-588-9

DSV Report Series No. 16-014

ISSN 1101-8526

Printed by US-AB, Stockholm 2016

Distributor: Department of Computer and Systems Sciences, Stockholm University

# Abstract

Separation of concerns has long been considered an effective and efficient strategy to deal with complexity in information systems. One sort of concern, like security and privacy, crosses over other concerns in a system. Such concerns are called cross-cutting concerns. As a result, the realization of these concerns is scattered through the whole system, which makes their management difficult.

Aspect Orientation is a paradigm in information systems which aims to modularize cross-cutting concerns. This paradigm is well researched in the programming area, where many aspect-oriented programming languages have been developed, e.g., AspectJ. It has also been investigated in other areas, such as requirement engineering and service composition. In the Business Process Management (BPM) area, Aspect Oriented Business Process Modeling aims to specify how this modularization technique can support encapsulating cross-cutting concerns in process models. However, it is not clear how these models should be supported in the whole BPM lifecycle. In addition, the support for designing these models has only been limited to imperative process models that support rigid business processes. Neither has it been investigated how this modularization technique can be supported through declarative or hybrid models to support the separation of cross-cutting concerns for flexible business processes.

Therefore, this thesis investigates how aspect orientation can be supported over the whole BPM lifecycle using imperative aspect-oriented business process models. It also investigates how declarative and hybrid aspect-oriented business process models can support the separation of cross-cutting concerns in the BPM area. This thesis has been carried out following the design science framework, and the result is presented as a set of artifacts (in the form of constructs, models, methods, and instantiations) and empirical findings.

The artifacts support modeling, analysis, implementation/configuration, enactment, monitoring, adjustment, and mining cross-cutting concerns while supporting business processes using Business Process Management Systems. Thus, it covers the support for the management of these concerns over the whole BPM lifecycle. The use of these artifacts and their application shows that they can reduce the complexity of process models by separating different concerns.



# Sammanfattning

Att separera angelägenheter har länge ansetts som en effektiv och ändamålsenlig strategi för att hantera komplexitet i informationssystem. Sådana angelägenheter, till exempel säkerhet och enskildhet, kan skära tvärs över andra angelägenheter i ett system, och de kallas därför övergripande angelägenheter. Hanteringen av dessa kan vara utspridda genom hela systemet, vilket ökar komplexiteten.

Aspektorientering är ett paradig som syftar till att modularisera övergripande angelägenheter. Detta paradig är väl utforskat i programvaruområdet, där många aspektorienterade programmeringsspråk utvecklats. Paradigmet har också undersökts i andra områden som kravhantering och tjänstesammansättning. I ärendehanteringsområdet (BPM) syftar aspektorienterad processmodellering till att inkapsla övergripande angelägenheter i processmodeller. Det är dock inte klart hur dessa modeller bör stödjas i hela ärendehanteringslivscykeln. Dessutom är existerande stöd för att utforma dessa modeller begränsat till tvingande processmodeller som enbart stödjer rigida affärsprocesser. Det har inte heller undersökts hur denna modulariseringsteknik kan stödjas genom deklarativa eller hybridmodeller för att hantera separation av övergripande angelägenheter för flexibla affärsprocesser.

Därför undersöker denna avhandling hur aspektorientering kan stödjas i hela ärendehanteringslivscykeln med hjälp av tvingande aspektorienterade affärsprocessmodeller. Den undersöker också hur deklarativa och hybridaspektorienterade affärsprocessmodeller kan stödja separation av övergripande angelägenheter i BPM-området. Avhandlingens resultat bygger på designvetenskaplig forskning, och de presenteras som en uppsättning av artefakter (i form av konstruktioner, modeller, metoder och instansieringar) och som empiriska iakttagelser.

De framtagna artefakterna stödjer modellering, analys, genomförande, konfiguration, övervakning och modifiering av övergripande angelägenheter i affärsprocesser. Artefakterna erbjuder stöd för hantering av dessa angelägenheter för hela ärendehanteringslivscykeln. Användningen av dessa artefakter och deras tillämpningar visar att de kan minska komplexiteten i processmodeller genom att separera övergripande angelägenheter.



*This thesis is dedicated to my parents  
Heidar Ali Jalali & Tahereh Aeen  
for their endless love and support.*



# List of Publications

The following papers are included in this thesis.

- [1] **Aspect Oriented Business Process Modelling with Precedence**  
Jalali A., Wohed P., Ouyang C.;  
*In the 4th International Workshop of Business Process Model and Notation (BPMN)*, (2012), Springer;  
DOI: 10.1007978-3-642-33155-8\_3  
*Amin Jalali contributed to all sections. His contribution to the paper corresponds to around 50%.*
  
- [2] **Multi-Perspective Business Process Monitoring**  
Jalali A., Johannesson P.;  
*In the 14th International Conference on Business Process Modeling, Development, and Support (BPMDS)*, (2013), Springer;  
DOI: 10.1007978-3-642-38484-4\_15  
*Amin Jalali contributed to all sections of this paper. His contribution to the paper corresponds to around 90%.*
  
- [3] **Aspect Mining in Business Process Management**  
Jalali A.;  
*In 13th International Conference on Perspectives in Business Informatics Research (BIR)*, (2014), Springer;  
DOI: 10.1007/978-3-319-11370-8\_18
  
- [4] **Assessing Aspect Oriented Approaches in Business Process Management**  
Jalali A.;  
*In 13th International Conference on Perspectives in Business Informatics Research (BIR)*, (2014), Springer;  
DOI: 10.1007/978-3-319-11370-8\_17
  
- [5] **Towards Aspect Oriented Adaptive Case Management**  
Jalali A., Bider I.;  
*In 18th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOC Workshops)*, (2014), Springer;

DOI: 10.1109/EDOCW.2014.30

*Amin Jalali contributed to all sections except Section IV of this paper. His contribution to the paper corresponds to around 70%.*

[6] **Agile Business Process Development: Why, How and When—Applying Nonaka's theory of knowledge transformation to business process development;**

Bider I., Jalali A.;

In *Information Systems and e-Business Management Journal (ISeB)*, (2014), Springer;

DOI: 10.1007/s10257-014-0256-1

*Amin Jalali contributed to this paper in different sections including the literature review, one reported case, and how aspect-oriented BPM can support agile business process development by applying the proposed framework. His contribution to the paper corresponds to around 33%.*

[7] **Static Weaving in Aspect Oriented Business Process Management;**

Jalali A.;

In *34th International Conference on Conceptual Modeling (ER)*, (2015), Springer;

DOI: 10.1007/978-3-319-25264-3\_41

[8] **Enhancing Aspect-Oriented Business Process Modeling with Declarative Rules;**

Jalali A., Maggi F.M., Reijers H.A.;

In *34th International Conference on Conceptual Modeling (ER)*, (2015), Springer;

DOI: 10.1007/978-3-319-25264-3\_8

*Amin Jalali contributed to all sections of this paper. His contribution to the paper corresponds to around 70%.*

[9] **Supporting Aspect Orientation in Business Process Management;**

Jalali A., Ouyang C., Wohed P., Johannesson P.;

In *Software & Systems Modeling Journal (SoSyM)*, (2015), Springer;

DOI: 10.1007/s10270-015-0496-7

*Amin Jalali contributed to all sections of this paper. His contribution to the paper corresponds to around 70%.*

---

There are also other publications to which I contributed, which are not included in this thesis. They are listed below.

- [10] **Foundation of Aspect Oriented Business Process Management**  
Jalali A.;  
*Master thesis*, (2011), Stockholm University.
- [11] **Operational Semantics of Aspects in Business Process Management**  
Jalali A., Wohed P., Ouyang C.;  
*In On the Move to Meaningful Internet Systems: OTM Workshops*, (2012), Springer.  
DOI: 10.1007/978-3-642-33618-8\_85
- [12] **Service Oriented Modularization using Coloured Petri Nets;**  
Jalali A.;  
*In Algorithmen und Werkzeuge für Petrinetze (AWPN)*, (2012);
- [13] **Dynamic Weaving in Aspect Oriented Business Process Management;**  
Jalali A., Wohed P., Ouyang C., Johannesson P.;  
*In On the Move to Meaningful Internet Systems: OTM 2013 (CoopIS)*, (2013), Springer;  
DOI: 10.1007/978-3-642-41030-7\_2
- [14] **Adaptive Case Management as a Process of Construction of and Movement in a State Space**  
Bider I., Jalali A., Ohlsson J.;  
*In 2nd International Workshop on Adaptive Case Management and Other Non-Workflow Approaches to BPM (AdaptiveCM)*, (2013), Springer.  
DOI: 10.1007/978-3-642-41033-8\_22
- [15] **Supporting Enactment of Aspect Oriented Business Process Models: An approach to separate cross-cutting concerns in action**  
Jalali A.  
*Licentiate thesis*, (2013), Stockholm University.
- [16] **Enacting Aspect Oriented Business Process Models**  
Jalali A.;  
*ISBN:3639868110,9783639868111*, (2014), Globe On-Demand, LLC.

- [17] **Creating Self-Managed Cross-Professional Teams with Metaphoric Business Process Support Systems**  
Bider I., Jalali A., Söderström D.;  
*In the 16th International Conference on Business Process Modeling, Development, and Support (BPMDS)*, (2015), Springer;  
DOI: 10.1007/978-3-319-19237-6\_2
- [18] **Limiting Variety by Standardizing and Controlling Knowledge Intensive Processes;**  
Bider I., Jalali A.;  
*In 20th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOC Workshops)*, (2016), IEEE;  
DOI: 10.1109/EDOCW.2016.7584366
- [19] **Reflections on the use of Chord Diagrams in Social Network Visualization in Process Mining;**  
Jalali A.;  
*In IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, (2016), IEEE;  
DOI: 10.1109/RCIS.2016.7549281
- [20] **Supporting Social Network Analysis using Chord Diagram in Process Mining;**  
Jalali A.;  
*In 15th International Conference on Perspectives in Business Informatics Research (BIR)*, (2016), Springer;  
DOI: 10.1007/978-3-319-45321-7\_2

---

Reprints were made with permission from the publishers.

# Contents

<b>Abstract</b>	<b>v</b>
<b>Sammanfattning</b>	<b>vii</b>
<b>List of Publications</b>	<b>xi</b>
<b>Acknowledgements</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>19</b>
1.1 BPM lifecycle . . . . .	20
1.2 Business process formulation . . . . .	21
1.2.1 Imperative vs. declarative process models . . . . .	21
1.2.2 Modularization techniques . . . . .	23
1.3 Problem formulation . . . . .	24
1.3.1 Problem area . . . . .	24
1.3.2 Research questions . . . . .	28
1.3.3 Expected artifacts . . . . .	29
1.4 Disposition . . . . .	31
<b>2 Extended Background</b>	<b>33</b>
2.1 History of BPM . . . . .	33
2.1.1 Traditions behind BPM . . . . .	34
2.1.2 The emergence of BPM . . . . .	35
2.2 Workflow Management Systems . . . . .	37
2.2.1 Supporting flexibility in BPM . . . . .	39
2.3 The Aspect-Oriented Paradigm . . . . .	40
2.3.1 Aspect-Oriented Programming . . . . .	41
2.3.2 Aspect-Oriented service composition . . . . .	42
2.3.3 Aspect Orientation in Business Process Modeling . . . . .	42

<b>3</b>	<b>Research Methodology</b>	<b>47</b>
3.1	Design Science . . . . .	47
3.2	Research methods . . . . .	50
3.2.1	Formalizing Aspect-Oriented Business Process Models	50
3.2.2	Analysing Aspect-Oriented Business Process Models .	53
3.2.3	Enacting Aspect-Oriented Business Process Models .	55
3.2.4	Assessing Aspect-Oriented Business Process Models .	60
3.2.5	Supporting Agile Business Process Development . . .	61
3.2.6	Declarative Aspect-Oriented Business Process Models	62
3.2.7	Hybrid Aspect-Oriented Business Process Models . .	64
3.2.8	Discovering Aspect-Oriented Business Process Models	65
<b>4</b>	<b>Result</b>	<b>67</b>
4.1	Formalizing AO-BPM . . . . .	67
4.2	Analyzing Aspect-Oriented Business Process Models . . . . .	69
4.3	Enacting Aspect-Oriented Business Process Models . . . . .	73
4.3.1	Monitoring . . . . .	73
4.3.2	Running . . . . .	75
4.4	Assessing Aspect-Oriented Business Process Models . . . . .	81
4.5	Supporting Agile Business Process Development . . . . .	82
4.6	Declarative Aspect-Oriented Business Process Models . . . . .	85
4.7	Hybrid Aspect-Oriented Business Process Models . . . . .	90
4.8	Discovering Aspect-Oriented Business Process Models . . . . .	91
<b>5</b>	<b>Conclusion and Future work</b>	<b>93</b>
5.1	Conclusion . . . . .	93
5.2	Future research . . . . .	95
	<b>References</b>	<b>xcvii</b>
	<b>Included Papers</b>	<b>ciii</b>

# Acknowledgements

I would like to acknowledge those people who helped me in the last five years to come to this stage to finish my Ph.D. dissertation.

First, I would like to thank my supervisor Paul Johansson for his tolerance, help, and availability from the start of my study as a Ph.D. student. He was always available when I needed help, and he gave me enough freedom to learn and experience how to become an independent researcher.

Second, I would like to thank Petia Wohed who helped me to start my Ph.D. study. I would thank her for all her advice, even if I understood them a long time later.

Next, I would like to thank all my co-authors who helped me in different papers. I would like to thank Chun Ouyang for her contributions and help and hosting me at Queensland University of Technology. I would like to thank Ilia Bider, with whom I had good discussions and collaborations that helped me to broaden my perspective on Business Process Management. I also like to thank him for all his help and assistance. I would like to thank Hajo A. Reijers for his comments and help. I would also like to thank Fabrizio Maria Maggi not only for his help but also for his friendly and supportive attitude. It was a pleasure to work with him.

In addition, I would like to thank Henrik Boström for his valuable comments on the first draft of my thesis, which helped me to increase the quality of my thesis significantly. I also like to thank Jelena Zdravkovic not only for her comments on the first draft of the thesis but also for her help and support during my study.

I would also like to thank two of my best friends who helped me during my research. I would like to thank Saman Mesgari for his time and support in conducting a case study. I would also like to thank Marjan Taheri not only for helping me in doing a case study but also for being my best friend who always listened to me and gave me valuable advice.

I would also like to thank Martin Henkel not only for helpful hints and discussion during lunches but also to motivate me to speak in Swedish and be a good friend. I also like to thank Rahim Rahmani for his supportive and friendly advice. I also like to thank Eric-Oluf Svee for his openness, helps and discussions that we had during finalizing our theses.

Moreover, I would like to thank all my colleagues at the Department of

Computer and Systems Sciences at Stockholm University who helped me while I have studied there. I especially thank all my colleagues in the Information Systems unit who provided a very friendly environment and supported me.

Last but not least, I thank my parents. Indeed, I cannot find words to express my gratitude to them. I can only dedicate this work to them with all my love.

# 1. Introduction

Business Process Management is an area of research that aims at making business processes more effective and efficient in organizations. A *business process* is defined as a set of activities performed to achieve an organizational goal [21]. These activities address different concerns for fulfilling different goals. The set of activities of a business process and the organizational settings that are required to perform those activities can be considered as the boundary of that business process. *Business Process Management (BPM)* is defined as a collection of techniques to support designing, administrating, configuring, enacting, and analyzing business processes [21].

There are usually many people involved in running a business process, and there are different concerns, such as security, that should be considered while running a business process. These concerns can be met by means of a set of activities that can be performed in a business process to fulfill different goals. Some of these concerns are not limited to one business process; instead, there are many processes that should comply with these concerns. These concerns cross over the boundary of a business process, so they should be considered in many business processes. The encapsulation of these concerns can hinder the design and support of business processes because they not only increase the complexity of the process models but also make it cumbersome to change them. Such concerns are known as cross-cutting concerns, and are usually present in business processes. Charfi et al. list examples of such concerns as dealing with compliance, auditing, business monitoring, accounting, billing, authorization, privacy, and separation of duties [22].

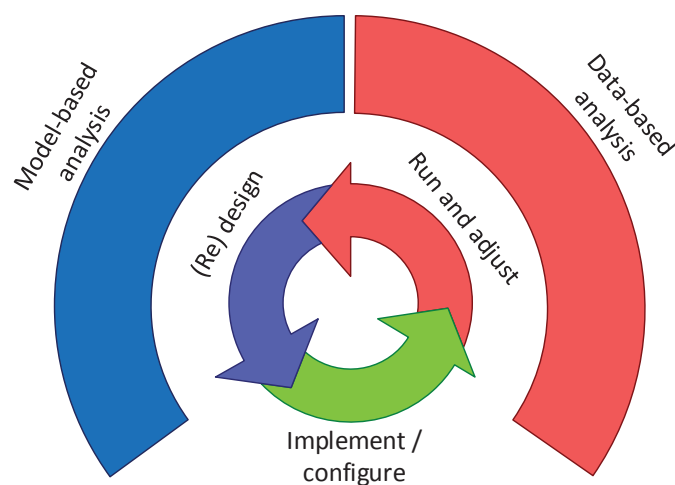
This thesis is about supporting the separation of cross-cutting concerns in Business Process Management. This separation is supported by defining techniques to enable managing the separation of cross-cutting concerns in the BPM area. These techniques can enhance the capability of organizations to manage their cross-cutting concerns in a more effective and efficient way. This section starts by introducing the BPM Lifecycle, which defines different levels of support for business processes. Then, it presents different approaches to formulating a business process. Thereafter, it introduces the research questions, and it lists the contributions. Finally, it summarizes the structure of the rest of the thesis.

## 1.1 BPM lifecycle

Supporting the management of business processes in organizations includes the support for different phases, including *identifying, designing, analyzing, configuring, implementing, executing, adjusting, monitoring* and *evaluating* business processes. One procedure that specifies how business processes can be managed through these phases is called BPM lifecycle. There have been several BPM lifecycles proposed by organizations and researchers based on their aims, e.g., [21; 23; 24]. Although they have some differences, they cover four fundamental activities that are needed to manage business processes, i.e., Model, Analyze, Manage, and Enact [24].

The BPM lifecycle proposed by van der Aalst [24] focuses more on supporting the management of business processes using IT artifacts (see Figure 1.1). It contains a core cycle that includes three phases, i.e., i) (re)design; ii) implement/configure; and iii) run and adjust. In addition, it has two complementary phases, called model-based and data-based analysis.

A process modeler designs a business process in the design phase, and the result is a process model. This model can be analyzed (model-based analysis), so new insights can be used to redesign the business process. The model can also be used to implement and configure business processes. Finally, the process can be run, and it can be adjusted according to new requirements. The enactment of a business process can produce data, which can be used for analysis at the data-based analysis phase. The new insights can be used to redesign the process model or to adjust the process model to work more efficiently.



**Figure 1.1:** The BPM Lifecycle, taken from [24]

## 1.2 Business process formulation

As mentioned, a business process is defined as a set of activities, and these activities can be enacted in different orders, which are defined by some constraints. A *business process model* is a specification of a business process including a set of activities and the execution constraints between them [21]. The process model plays a significant role in managing business processes because it is almost considered as a starting point for managing business processes. Business process models can easily become too complex in a way that exceeds the human capability to understand and manage them. The models can become very complex if the modeling notation does not support the formulation of the business process adequately, or the business process is itself complex.

Business process models range from the very flexible to the very rigid. Therefore, it is important to choose the right technique to model them. There are two approaches that aim at supporting rigid and flexible business processes, called Imperative and Declarative [25].

Moreover, business processes can involve many concerns that make the process models very complex. Separation of concerns is an approach to deal with the complexity of process models [26]. To separate different concerns, different researchers have proposed different modularization techniques to encapsulate these concerns.

### 1.2.1 Imperative vs. declarative process models

There are two approaches to modeling business processes: *Imperative* and *Declarative* [25].

*Imperative modeling* aims to formulate business processes by specifying mandatory constraints that should be considered while enacting a business process, i.e., the order of activities that should be followed in a business process. Therefore, they are suitable for specifying rigid end-to-end business processes. Many notations for imperative modeling have been proposed to support the definition of this sort of business processes, e.g., Petri nets [27], Business Process Model and Notation (BPMN) [28], Yet Another Workflow Language (YAWL) [29], Unified Modeling Language (UML) [30], Event-driven Process Chain (EPC), etc. In general, they show activities in a business process as some boxes, and they define the constraints using flows and sometimes some more visual elements. These notations are good candidates for modeling workflow based processes.

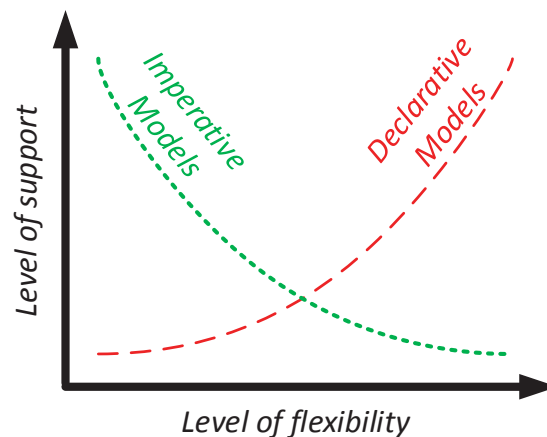
Although these models provide a good level of support for rigid business processes, they might not be very effective for designing flexible business processes. In flexible business processes, there are many ways to perform the

activities to realize the goal of the business process. Formulating this sort of processes using imperative modeling techniques can result in a huge number of flows connecting different activities to each other. These models are also known as spaghetti models, e.g., [31; 32]. The huge number of flows hinders the comprehensibility of the process models by people. Thus, a different modeling approach is required to model these business processes.

*Declarative modeling* aims to formulate business processes by mainly specifying forbidden constraints. These constraints should not happen while enacting a business process. Indeed, any order of activities can happen unless violating some specified constraints. Declarative modeling notations have been proposed to support the definition of this sort of business process, e.g., [25; 33]. These notations are good candidates for modeling knowledge-intensive processes in which people have more freedom to run the business processes based on their expertise.

Figure 1.2 shows the degree of support that these modeling techniques offer for different business processes. The horizontal axis displays the level of flexibility of the business processes, and the vertical axis displays the degree of support that can be offered by a business process modeling notation. The green and red dashed lines show the degree of support that imperative and declarative models provide, respectively. As can be seen, imperative models can provide better support for rigid (less flexible) business processes; while declarative models can offer better support for flexible business processes [25].

There are also some processes that cannot be effectively specified either by imperative constraints alone or by purely declarative constraints. Instead, a combination of these constraints defines how those processes should work. The techniques that aim at specifying processes using a combination of imperative and declarative constraints are called *hybrid modeling* techniques [34; 35].



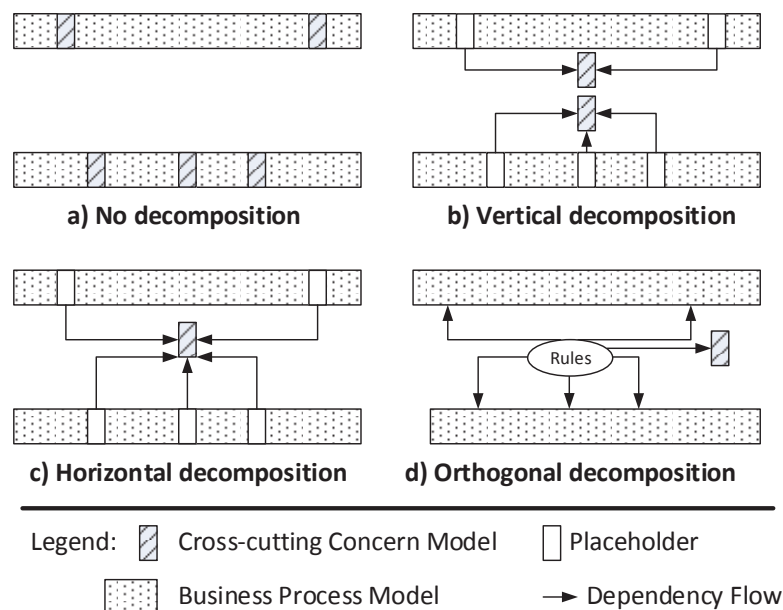
**Figure 1.2:** Imperative vs. declarative business processes

## 1.2.2 Modularization techniques

Dealing with complexity in business process modeling is a challenge that a process designer needs to face whether following imperative or declarative business process modeling. Modularization techniques are means to separate the different concerns of a phenomenon so as to facilitate coping with its complexity. La Rosa et al. categorize the modularization techniques in the BPM area into three classes: Vertical, Horizontal and Orthogonal [26]. Figure 1.3 shows an abstract representation of these techniques in general. It shows business process models and the concerns that should be considered in the processes.

Figure 1.3.a. shows a scenario where no decomposition is applied. As can be seen, the concern should be re-modelled several times in each process model. This repetition adds several costs, including design and maintenance. To avoid such problems, the concerns should be encapsulated in a way that is re-usable.

Figure 1.3.b. shows a scenario where vertical decomposition is applied. This decomposition suits a situation where the concerns are limited to one process model, and the process designer knows the places in the process models where that concern should be considered. As can be seen, the concern is encapsulated in a separate module for each process model, and is called by adding a placeholder in the process model. Although the encapsulated module can be re-used several times within the scope of a process model, the concern should be re-modeled for every business process model. This still hinders its



**Figure 1.3:** An abstract representation of modularization techniques for BPM

re-usability if the domain of the concern is not limited to one process model.

Figure 1.3.c. shows a scenario where horizontal decomposition is applied. This decomposition suits a situation where the concerns are not limited to one process model, and the process designer knows the places in the process models where the concern should be considered. As can be seen, the concern is encapsulated in a separate module, and is called by adding a placeholder in the process models. It is very effective decomposition technique, but also costly to develop because the process designer should know the places in the process models where the concern should be considered.

Figure 1.3.d. shows a scenario where orthogonal decomposition is applied. This decomposition suits a situation where the concerns are not limited to one process model, and the process designer does not know the places in the process models where the concern should be considered. As can be seen, the concern is encapsulated in a separate module. The concern is imposed in process models when some conditions are satisfied. These conditions can be articulated using some rules that specify the places in the process models where the concern should be applied. In this way, if the condition of implementing the concern is changed, the process designer need only change these rules. This modularization technique can be used to support flexibility in business processes, where the place of the flexibility cannot be known beforehand. Worklet exception service is an example of an orthogonal modularization technique to support flexibility in business processes [36].

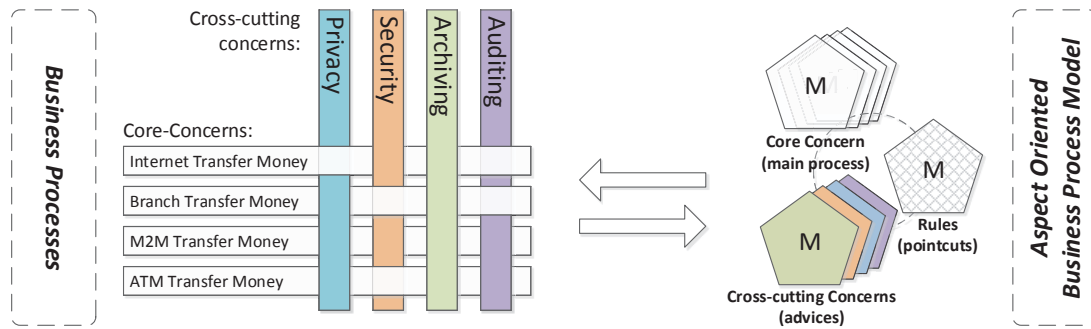
## 1.3 Problem formulation

This section introduces the area of the problem, the research goal and the research questions that this thesis aims to address.

### 1.3.1 Problem area

Business processes should comply with many concerns, and some of these concerns are not limited to one process model. To explain these concerns and their relations to process models, we use a fictitious case for transferring money, from the banking domain, which can be seen in Figure 1.4. The left-hand side of the figure shows an abstract representation of the concerns that are not restricted to one process model. In this figure, the business process models and concerns are represented by horizontal and vertical rectangles, respectively.

There are different processes specifying alternative ways a customer can transfer their money, e.g., i) via the Internet (captured in Internet Transfer Money process); ii) at a branch of the bank (Branch Transfer Money process); iii) using the mobile to mobile transfer service (M2M Transfer Money



**Figure 1.4:** Cross-cutting concerns in business processes

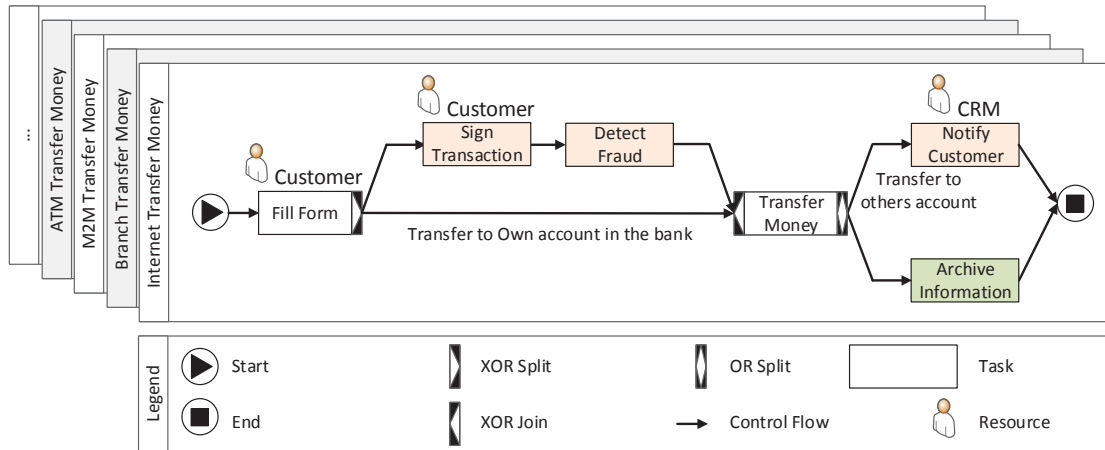
process); iv) through an Automated Teller Machine (ATM Transfer Money process); etc. There are also some cross-cutting concerns that might be applicable in some of these processes, such as privacy, security, archiving and auditing.

The visual expression that a business process must address a concern is that the concern crosses over it. For example, the Internet Transfer Money process should comply with the Security and Archiving cross-cutting concerns. The concerns in this figure that are not limited to one business process, such as security, are examples of *cross-cutting concerns*.

A *security* concern states that additional actions should be taken if the transaction is nominated as inadequately secured. The definition of such a nomination varies in each of these processes. For example, such a circumstance is defined in the Internet Transfer Money process when the customer wants to transfer money to another's account.

In the Branch Transfer Money or ATM Transfer Money process, such an investigation is not needed since the identity of the customer is verified by a clerk or by Personal Identification Number. In the M2M Transfer Money process, such an investigation is needed when the customer wants to transfer money to unregistered numbers or the amount to transfer exceeds the limit set by the customer as the default. The extra security considerations involve a sequence of actions, such as the customer's signing the transaction, the system's investigating whether the transaction is a potential fraud (through the Sign Transaction and Detect Fraud tasks, respectively) before the money can be transferred. Moreover, the Customer Relationship Management (CRM) department should notify the customer about the transaction through the Notify Customer task after the money transfer.

As an *archiving* policy, all transactions should be archived after the money has been transferred (through the Archive Information task). Both policies are illustrated using the Internet Transfer Money process in Fig. 1.5, where the customer should first carry out the Fill Form task, and then move



**Figure 1.5:** Transfer Money processes

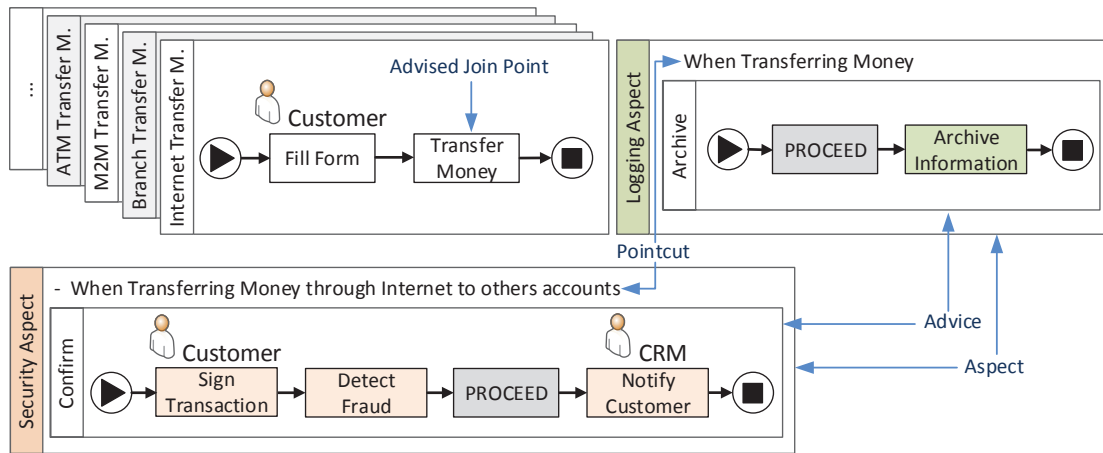
on to the Transfer Money task.

There are different sort of cross-cutting concerns which have been reported to be relevant for managing business processes, e.g., compliance, accounting, billing, monitoring, authorization, separation of duties [22], security, auditing, informativeness [37].

Encapsulating these concerns using vertical or horizontal modularization technique can introduce two problems called *scattering* and *tangling* [38]. The *scattering* problem can happen if these concerns are not encapsulated, or if they are encapsulated using the vertical decomposition technique. In this way, they should repeatedly be specified in several business process models, so their specifications are scattered through the system. The *tangling* problem can happen if these concerns are encapsulated using the horizontal decomposition technique. In this way, if the condition under which these concerns should be applied in different business processes is changed, all affected process models should be identified and changed. Therefore, the process models are tangled with the implementation of these concerns.

The orthogonal decomposition technique can solve the scattering and tangling problems, where aspect orientation is a technique to address these problems [38]. This technique was initially used in the programming area, where different aspect-oriented programming languages had been developed to address these problems, e.g. [39]. It has also been used in other areas, like requirement engineering and service composition [40–43]. In the BPM area, Aspect-Oriented Business Process Modeling (AO-BPM) is a modeling technique to support the design of such process models and their cross-cutting concerns. Some researchers have proposed Aspect-Oriented Business Process Modeling approaches to address the scattering and tangling problems, e.g., [22; 44].

The right-hand side of Figure 1.4 shows the elements of an AO-BPM tech-



**Figure 1.6:** Transfer Money processes: Through Aspect-Oriented Business Process Modeling

nique in general. The realizations of cross-cutting concerns are encapsulated into modules called *advices*, so the process models contain the core functionalities of the business processes, called *core-concerns*. Each core concern is encapsulated as a process model, and rules are encapsulated in separate modules that specify how these two modules should be connected to each other. These rules are called *pointcuts*. The encapsulation of some related advices and pointcuts, named *aspect*, specifies when and how a cross-cutting concern should be considered in core concerns.

Fig. 1.6 depicts how modularization of the processes in Fig. 1.5 is envisaged. In essence, the tasks related to the security aspect are extracted in a process called *Confirm*. Similarly, the tasks required by the logging aspect are encapsulated in the process *Archive*. These processes are related to the core processes (e.g., *Internet Transfer Money*, *Branch Transfer Money*) via certain rules, which can relate these concerns to the tasks in the core concerns. The tasks in core concerns are called *join points*. The join point which is related to an advice is called an *advised join point*, e.g. *Transfer Money* task. Advices may have a special activity named *PROCEED*, which acts as a placeholder to show the execution place of an advised join point. At runtime, advices will be triggered when the states fulfilling pointcuts are reached.

Such modularization may bring the following benefits. Firstly, it increases the re-usability since the concerns are defined once and used in several processes. Secondly, it facilitates maintenance, as a change to a concern only needs to be specified in one place. Thirdly, it reduces the complexity of business process models because i) it can “hide” unnecessary information from some stakeholders while displaying it in a concise form to the relevant stakeholders; ii) it often reduces the overall number of tasks; and iii) it may even eliminate the need for some control-flow elements like splits and joins that are

used to relate the core and cross-cutting concerns, as such relation would already be specified in the connection rules between the main process and cross-cutting concerns.

Despite attempts to develop different imperative aspect-oriented business process modeling approaches [22; 44–47], there is no formal definition of the syntax and semantics for such models. The same limitation applies to the configuration, implementation, and enactment of the models, where some publications have implemented some artifacts to support the enactment of aspect-oriented business process models [48; 49] without having a formal foundation. Formal foundations are necessary for a process model because they i) remove ambiguity from interpreting the model, and ii) increase the potential for their analysis [50].

As a result of the lack of a formal foundation for imperative aspect-oriented business process models, there is no analysis technique for these models and the data that can be generated by enacting these business processes. The result is a lack of supporting aspect orientation for the whole BPM lifecycle. In addition, there has been no research investigating the design of declarative or hybrid aspect-oriented business process models. This thesis aims to address these limitations to support the separation of cross-cutting concerns for the whole BPM lifecycle.

### 1.3.2 Research questions

The main question that this thesis aims to answer is how the separation of cross-cutting concerns can be supported using aspect orientation in business process management using imperative models. In addition, this thesis aims to explore how this separation can be supported using declarative and hybrid models. Lastly, this thesis aims to investigate how these models can be discovered from event logs.

Therefore, the research questions can be defined as follows.

1. How should the separation of cross-cutting concerns be supported by means of imperative formal aspect-oriented business process models based on aspect-oriented modularization?
  - 1.1. How should the design of imperative aspect-oriented business process models be formalized based on current informal aspect-oriented techniques?
  - 1.2. How should imperative aspect-oriented business process models be analyzed to check design time problems in business process models?

- 1.3. How should the enactment procedure (including configuration/Implementation, running and adjustment) of imperative aspect-oriented business process models be formalized?
- 1.4. How should imperative aspect-oriented business process models be assessed in terms of supporting the separation of cross-cutting concerns?
- 1.5. How can aspect-oriented business process management support agile business process development?
2. How should the separation of cross-cutting concerns be supported by means of declarative aspect-oriented business process models?
3. How should the separation of cross-cutting concerns be supported by means of hybrid aspect-oriented business process models?
4. How should aspect-oriented business process models be discovered (mined) by analyzing historical data obtained from enacting business processes?

In this thesis, AOBPM is used as an acronym for Aspect Oriented Business Process Management, while AO-BPM is used as an acronym for Aspect-Oriented Business Process Modeling.

### 1.3.3 Expected artifacts

Table 1.1 lists the expected artifacts as a result of this thesis to address the mentioned research questions. The first column shows the included papers, and the second column indicates the artifacts which are developed in that paper. Other columns shows the research questions, and the relation between each artifact and the related research question is marked in the table.

These are the artifacts which are developed in each paper:

- In [1], we defined the formal definition for aspect-oriented business process model and notation, named *AO-BPMN 2.0* in the table.
- In [2], we defined the formal definition for multi-perspective business process models (*MP-BPM*), an algorithm to support monitoring process instances (*MonitoringALG*), a rule editor to define monitoring rules (*RuleEditor*) and a service in YAWL that monitors business process instances (*Monitoring Service*) in the table.
- In [3], we defined a *Mining Method* to discover aspect-oriented business process models from log files.

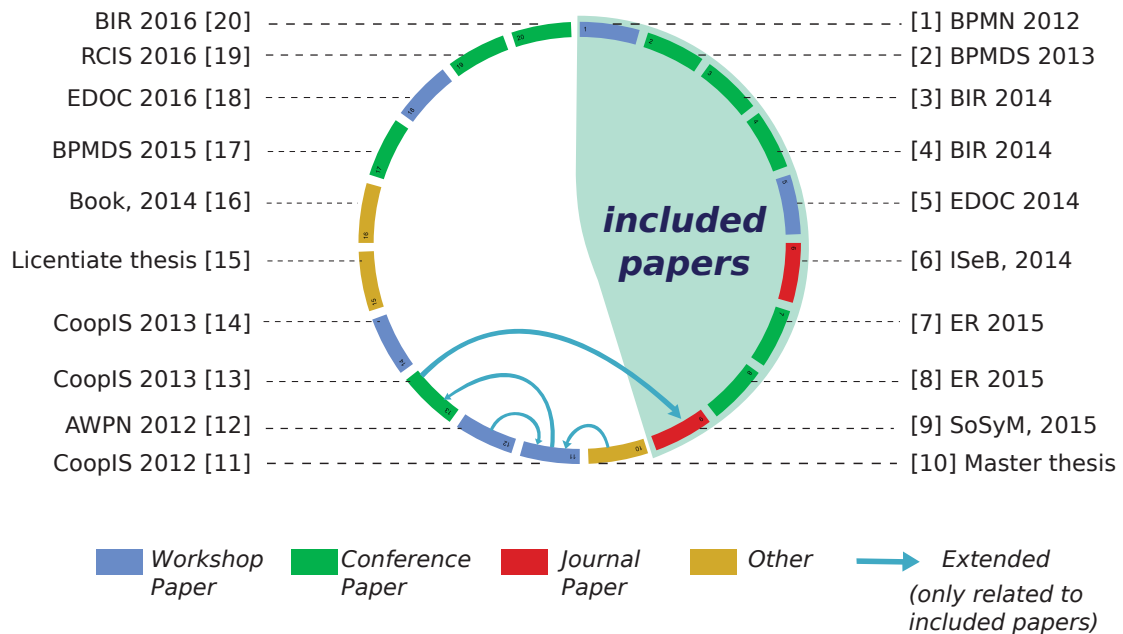
**Table 1.1:** Contributions and research questions

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[1]	AO-BPMN 2.0	✓							
[2]	MP-BPM	✓							
	MonitoringALG			✓					
	RuleEditor			✓					
	Monitoring Service			✓					
[3]	Mining Method								✓
[4]	Assessment Framework				✓				
[5]	Declarative AO-BPM						✓		
[6]	Agile Guideline					✓			
[7]	AO-Petri nets	✓							
	Static Weaving ALG		✓						
	AO-Petri nets Editor	✓							
	Static Weaver		✓						
[8]	Hybrid AO-BPM							✓	
[9]	AO-BPM	✓							
	Dynamic Weaving			✓					
	AO-YAWL Editor	✓							
	Aspect Service			✓					

- In [4], we defined an *Assessment Framework* to enable assessing aspect-oriented business process management approaches.
- In [5], we defined a declarative aspect-oriented business process modeling language, named *Declarative AO-BPM* in the table.
- In [6], we defined a guideline and its application to show how aspect-oriented business process management can support agile business process development, named *Agile Guideline* in the table.
- In [7], we defined a formal definition of an executable aspect-oriented business process modeling language using Petri nets (*AO-Petri nets*). We also defined an algorithm that converts these models to traditional models which can be analyzed (*Static Weaving ALG*). We implemented an editor for defining these models (*AO-Petri nets Editor*). We also implemented an algorithm to support weaving aspect-oriented business process models (*Static Weaver*).

- In [8], we defined a hybrid aspect-oriented business process modeling approach, named *Hybrid AO-BPM* in the table.
- In [9], we defined formal definitions for aspect-oriented business process models (*AO-BPM*) supported by a tool (*AO-YAWL Editor*). We also defined how the configuration and enactment of these models should be performed through *Dynamic Weaving*, which is supported by a service developed in YAWL (*Aspect Service*).

Figure 1.7 shows both the included and the excluded publications. An arrow between two publication  $A \rightarrow B$  shows that publication  $B$  is an extension of publication  $A$ . The extension relation is only shown for those related to included publications.



**Figure 1.7:** The list of publications

## 1.4 Disposition

The rest of this thesis is organized as follows. Chapter 2 presents and discusses the extended background. Chapter 3 presents the research methodology. Chapter 4 summarizes the results. Chapter 5 concludes the thesis and outlines future research.



## 2. Extended Background

This section summarizes the history behind Business Process Management (BPM). In addition, it describes how the aspect-oriented paradigm emerged in this area, based on advances in the programming area.

### 2.1 History of BPM

People have practiced managing business processes for as long as they have been running businesses throughout history, but the use of machines to automate business processes can be traced back to the industrial revolution. In this period, people enhanced their capabilities by shifting the traditional way of working from the handheld production concept to the machine one, starting around 1760. This transition resulted in the invention of a vast number of artifacts and new ways of managing works to improve efficiency. As an example, Henry Ford employed a new approach to simplify work by assigning specific tasks to each group of workers in 1903, so they could do their jobs more efficiently. He defined the whole production line as a sequential process including several tasks. This simple process-based thinking revolutionized the automobile manufacturing industry [51]. Indeed, it was a starting point to define business processes as we know them today, i.e., a business process is a set of activities that are performed to realize a business goal in an organization [21].

As a result, people invented new approaches to control the quality of work and improve processes. As a famous example, Frederick Winslow Taylor proposed a systematic approach, called the Principles of Scientific Management, in 1911, offering a systematic approach to identifying the best way of performing a task, controlling the system, and measuring the output. These works are based on the idea of Work Simplification, which is considered as a foundation for new disciplines like Business Process Management (BPM) [51]. BPM is defined as a set of concepts, methods, and techniques to support the management of business processes [21].

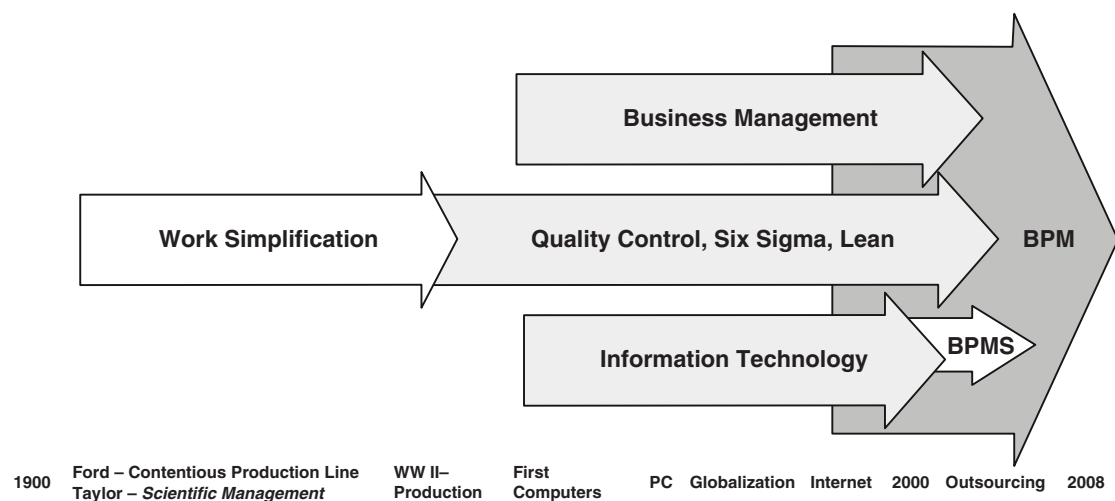
### 2.1.1 Traditions behind BPM

Work simplification inspired three traditions that are considered as foundations for the BPM tradition, i.e., the traditions of quality control, business management, and Information Technology (IT) [51]. Figure 2.1 shows how the BPM tradition has evolved based on these traditions over time.

*The quality control tradition* can be traced back to 1946, when the American Society for Quality (ASQ) was established. However, it did not much influence the American market until 1980, when they realized they were losing market share to Japanese companies due to the latter's use of better quality control. This tradition has been employed in Japanese companies very well, where a corporation like Toyota invented its method for its production system, called Lean. Apart from Lean, other methods have also been developed to control the quality of processes, like Total Quality Management (TQM), Six Sigma, Capability Maturity Models, and Lean Six Sigma. These traditions have been followed mostly by engineers and quality control specialists [51].

*The business management tradition* started around the 1980s when the US firms realized they were losing the automobile market to the Europeans and Japanese due to quality issues. The result affected how managers in the US refocused on reducing prices and improving production quality to improve the overall performance of the firm. The focus is on aligning strategy with the means of realizing them to achieve corporate goals. There are many approaches, proposed by different researchers, in this tradition, including Geary Rummler's approach, Michael Porter's approach, Balanced Scorecard, and Business Process Reengineering [51].

Rummler's approach focuses on processes and the related elements of the business environment that impact on process performance. Michael Porter pro-



**Figure 2.1:** Business Process Management Evolution, taken from [51]

poses an approach in which he considers the processes of a company as a value chain that should be aligned with the corporation's strategy to achieve a competitive advantage. He divided the processes into core and support processes. Core processes produce values for the business that are supported by support processes. A balance scorecard is a measurement approach based on operations in an organization, proposed by Robert S. Kaplan and David P. Norton. This approach links the strategy to processes and people. Business Process Reengineering convinced many managers to rethink their business strategy. This method is based on value chains, and it focuses on employing IT to refine the whole process. Indeed, it is common to the business management and IT traditions [51].

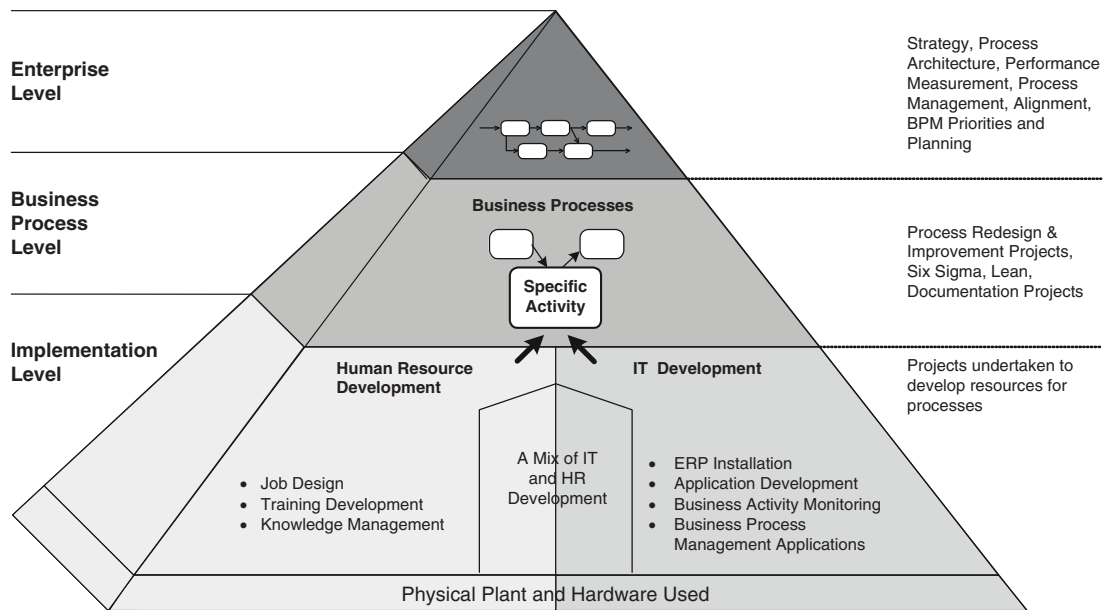
*The IT tradition* started in the 1960s. It focuses on using IT to support business processes. Different strategies have been proposed in this tradition, including Business Process Reengineering, Enterprise Resource Planning Applications, CASE and Process Modeling Tools, Expert Systems and Business Rules. This trend had continued with the development of generic information systems which are driven by process models to coordinate the enactment of business processes. These systems are known as Business Process Management Systems (BPMSs) [21].

As a result of all the inventions and software that have been developed and used in this tradition, executives have found that there is no distinct border between the business model and the latest IT technology. Indeed, they realized that IT should be considered as an essence of the company's strategy to achieve success [51].

### 2.1.2 The emergence of BPM

BPM has emerged as a combination of three early traditions to align different levels of businesses to their strategies. The business process trends Pyramid is a framework that is widely used to describe various business process activities that happen in a business at different levels (see Figure 2.2). This framework contains three levels, i.e., the Enterprise Level, Business Process Level and Implementation Level [51].

The Enterprise Level focuses on *identifying* processes and aligning them with the business strategy. The result is a process architecture that facilitates aligning the business strategy with value chains and high-level processes. The result enables process prioritization and planning, which empowers decision makers to decide how to better manage the business and its processes. At this level, there are areas of research like BPM with a focus on people and culture, BPM maturity measures, Social BPM, BPM and Crowdsourcing, Process innovation, Human-centric processes and knowledge-intensive pro-



**Figure 2.2:** The business process trends pyramid, taken from [51]

cesses, BPM strategic alignment and governance, Process architectures and value chains, and Process execution architectures.

The Process Level focuses on the *design* and *analysis* of the processes within the scope of the process architecture. As a result, the processes are documented and redesigned at this level. Lean and Six Sigma are two examples of this approach that can be used to improve processes in organizations. Moreover, there are many business process modeling notations and analysis techniques that have been invented to facilitate the efforts that are performed at this level. At this level, there are some areas of research like BPM lifecycle management, Process modeling languages, Formal methods in BPM, Management of process model collections such as querying, refactoring, similarity search, and versioning, Process variability and configuration, and Process Mining.

The Implementation Level focuses on executing the defined process models in the organization. It requires development efforts in both the Human Resource and IT sections of the organization. Examples of development efforts in Human Resource section include Job Design, Training Development, Knowledge Management, etc. Examples of development efforts in IT section include Application Development, Business Activity Monitoring, Business Process Management Applications, etc. The result is the *configuration, implementation, execution, adjusting, monitoring* and *evaluating* business processes in the organization. At this level, there are some areas of research like BPM systems, Case management, Adaptive and context-aware process execution, Management of process execution aspects (e.g., resources, data), and

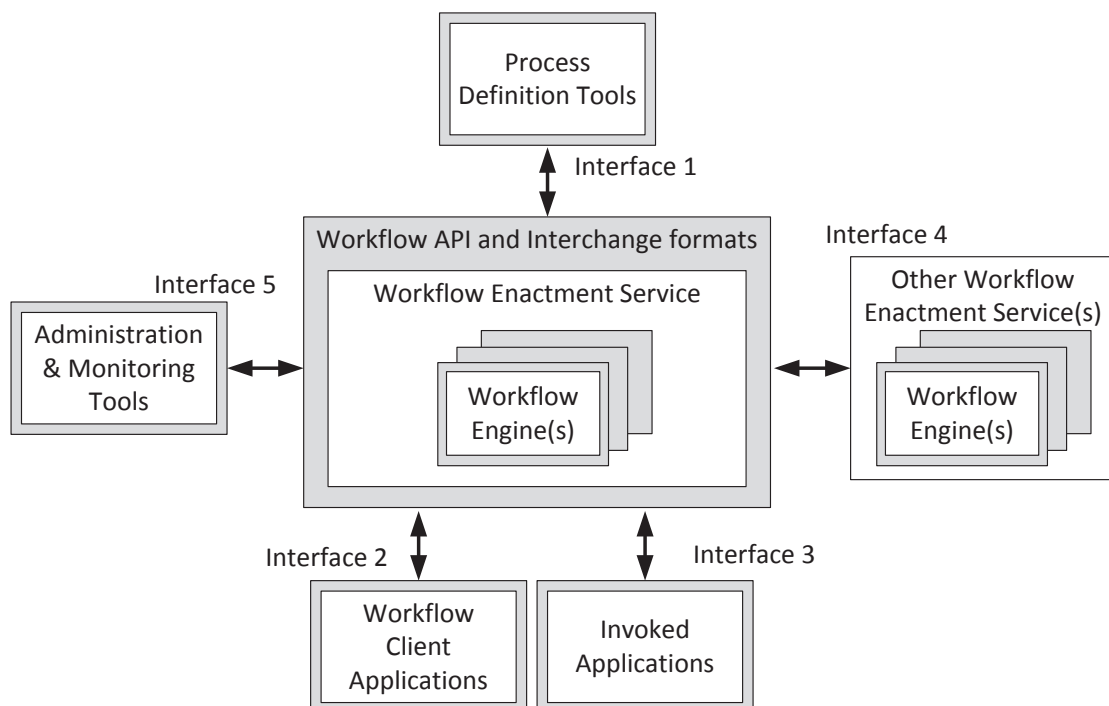
Process dashboards, analytics and visualization of big process data.

This thesis focuses on supporting the management of business processes using information systems, so it will next introduce some basic definitions and terms required for the subsequent chapters.

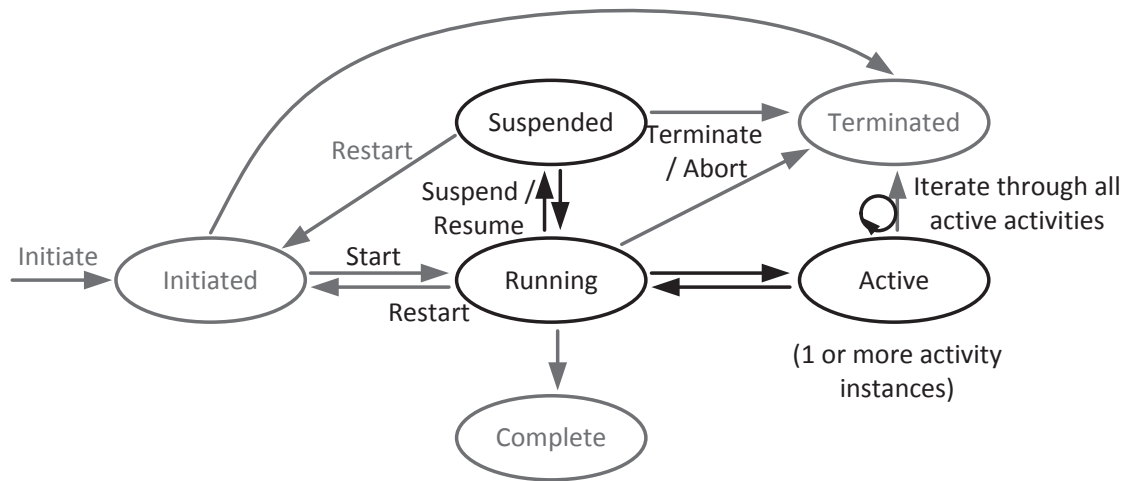
## 2.2 Workflow Management Systems

The implementation of business processes has long been supported by tools called Workflow Management Systems (WfMS). These systems support the design of business processes. They also support the configuration, implementation, execution, monitoring and adjustment of business process models. A generic model that specifies a standard for the architecture and functionalities of a WfMS is defined by the Object Management Group (OMG), called the Workflow Reference Model [52]. This model describes the major functional components, interfaces, and information interchange flows of a general WfMS. Figure 2.3 shows the major components and interfaces of a general WfMS.

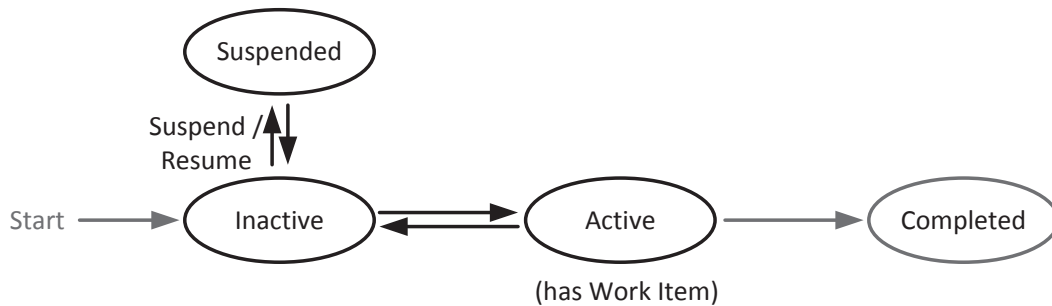
Each component of a WfMS is drawn as a rectangle, and their interfaces are drawn around them. The Workflow API and Interchange formats is a layer which acts as a unified service interface supporting functionalities for other interfaces to work with the Workflow Enactment Service. The Workflow Enactment Service is responsible for managing and executing



**Figure 2.3:** The Workflow Reference Model, taken from [52]



**Figure 2.4:** Workflow Reference Model: Example state transitions for a process instance, taken from [52]



**Figure 2.5:** Workflow Reference Model: Example state transitions for a task instance, taken from [52]

workflow instances. In our case, this service manages and runs all the main processes and related advice. The *Workflow Engine* provides a runtime environment for a workflow instance. It is responsible for interpreting a process instance, controlling its state, managing its task execution, managing data, executing supervisory actions for controlling processes, etc.

A process instance can have different states, and they are varied in different systems. A basic state transition of a process instance has *Initiated*, *Running*, *Suspended*, *Active*, *Terminated* and *Complete* states [52] (see Fig. 2.4). Moreover, a task instance can have different states. Such states are general in different systems like *Inactive*, *Suspended*, *Active* (also named *Started* [53] or *Running* [54]) and *Completed* [52] (see Fig. 2.5).

The ability to move between the *Running*, *Suspended* and *Active* states in a process instance and between the *Active* and *Suspended* states in a task instance enables developers to extend the functionality of a WfMS by implementing services that collaborate with the engine. The functionality of a WfMS can be extended to support different sorts of modularizations in busi-

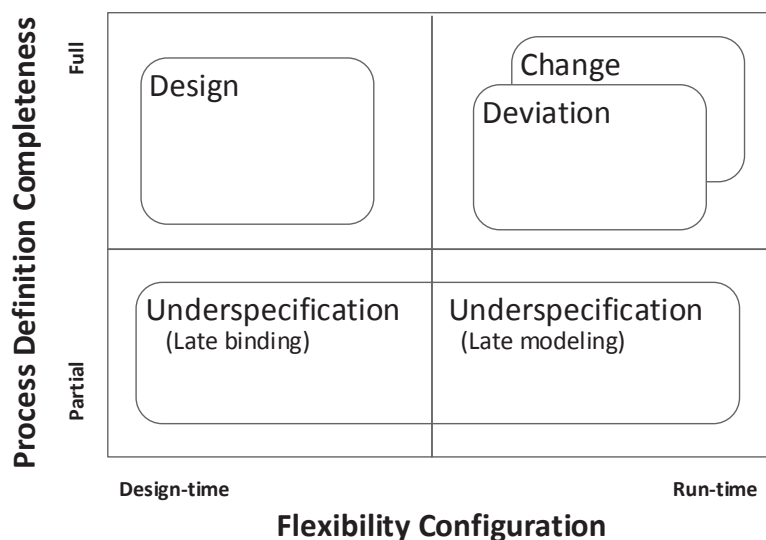
ness processes. Each modularization technique can support different sorts of flexibility in business processes.

### 2.2.1 Supporting flexibility in BPM

Schonenberg et al. [55] define four types of process flexibilities based on two criteria: i) if the process definition is complete; and ii) if the solution to support the flexibility is configured at design time or runtime (see Figure 2.6).

If the process definition is complete, and if the solution to support the flexibility is configured at design time, the flexibility is defined as Flexibility by Design [55] (see Figure 2.6). This sort of the flexibility can be supported if the point of flexibility, the solutions, and how the solutions should be applied are completely known at design time, so the solution can be captured within the process model. This means that the process definition is complete, and the flexibility is handled at design time. This kind of flexibility is capable of providing a solution for different situations that could happen in a process model. For example, what should be done if the payment has failed, or what sort of actions should be taken if there are no resources available to answer customers? This sort of flexibility is limited to the design phase of the BPM life-cycle. This means that it cannot fulfill the goal of flexibility at runtime. The traditional knowledge transformation model supports this kind of flexibility.

If the process definition is not complete, and if the solution to support the flexibility is configured at design time, the flexibility is defined as Flexibility by Underspecification (Late binding) [55] (see Figure 2.6). There are situations where the point of flexibility and the solutions are known at design time, but the decision regarding how the solutions should be applied is not



**Figure 2.6:** Process flexibility types, taken from [55]

known at design time. In these situations, process models can contain a placeholder to specify the point of flexibility, and the solution is bound to these points at runtime. This means that the process model is partially specified, the solutions are modeled at design time, but they are bound later. This approach enables the reusability of solutions in other process models as well, since they are not modeled in a particular process model.

If the process definition is not complete, and if the solution to support the flexibility is configured at runtime, the flexibility is defined as Flexibility by Underspecification (Late modeling) [55] (see Figure 2.6). In this situation, the point of flexibility is known at design time, but the solutions are not known at design time. Clearly, when the solution is not known, the knowledge of how to apply it is not available either. In these situations, the process models can contain a placeholder to specify the point of flexibility, and the solution can be modeled later (when the process model is enacted). This means that the process model is partially specified, and the solutions are modeled at runtime. This approach enables the reusability of solutions in other process models as well since they are not modeled in a particular process model.

If the process definition is complete, and if the solution to support the flexibility is configured at runtime, the flexibility is defined as Flexibility by Change or Deviation [55] (see Figure 2.6). This flexibility suits to situations in which neither the point of flexibility nor the solutions are known at design time. In these situations, the process models cannot specify the flexible point, so their specifications are complete. Two approaches can be taken in such circumstances: i) change the process specification at runtime and add a solution to support flexibility; ii) do not change the process specification, but enforce its execution to deviate from what is specified in the process specification [55]. The first approach is called Flexibility by Change, and the second approach is called Flexibility by Deviation. To extend the functionality of a WfMS to support aspect orientation, we should know the requirements that such a modularization technique requires.

## 2.3 The Aspect-Oriented Paradigm

Aspect-orientation is a paradigm that has emerged in information systems to facilitate the separation of cross-cutting concerns. Therefore, this section starts by introducing how this paradigm evolved in the BPM area by borrowing definitions and semantics from the programming area. Thus, the following sub-sections introduce Aspect-oriented Programming, aspect-oriented service composition, and aspect-oriented business process modeling.

### 2.3.1 Aspect-Oriented Programming

Separation of concerns has long been an important issue in programming, and different modularization techniques have been developed to support the separation of different concerns. Some concerns are scattered through the system, which makes different modules of the systems tangled to them if they are encapsulated using traditional modularization techniques. These concerns are known as cross-cutting concerns, and Aspect-Oriented Programming (AOP) has been introduced to support the encapsulation of these concerns [56]. These concerns can be high-level, such as the security and quality of service, or low-level, such as caching and buffering [56]. They can represent functional requirements of the system, such as features or business rules, or non-functional (systemic) requirements, such as synchronization and transaction management [56].

AOP aims at enabling programmers to develop the main functionality of the system without worrying about the definition of cross-cutting concerns. This feature is called obliviousness [38]. It also aims at enabling them to encapsulate the specification of cross-cutting concerns in different modules, called advice. Finally, AOP aims at providing a mechanism to support relating the advice to the main functionality of the system. Thus, it is a form of orthogonal modularization that is introduced in Chapter 1. This mechanism includes an approach to quantifying elements from the functionality of the system (called join points) and writing assertions that relate the advice to those elements. These assertions are called pointcuts. An encapsulation of advices and pointcuts that specify how a cross-cutting concern should be considered in a system is known as an aspect [56].

Filman et al. introduce quantification and obliviousness as two main features of an AOP approach [38]. They state that “AOP can be understood as the desire to make quantified statements about the behavior of programs, and to have these quantifications hold over programs written by oblivious programmers”, and they give an example of such a statement: “In programs P, whenever condition C arises, perform action A.” [38]. In this example, the assertion of condition C refers to *Quantification*. The definition of how actions should interact with each other is called an *Interface*, and the approach that defines how the program P and the action A should be executed is called *Weaving*.

Quantification can be considered along two dimensions. It can be static or dynamic, and it can be defined as clear-box or black-box [38]. Static/dynamic quantification supports asserting statements based on the information about the functionality of the program that is available at design/runtime respectively. Clear-box quantification supports asserting statements based on the structure of the program, so it requires having access to the source code. However,

Black-box quantification supports asserting statements based on the public interfaces that are defined in the program, so it does not require having access to the source code [38].

Weaving can also be static or dynamic. Static weaving defines a process that merges all pieces of an aspect-oriented program, so the result is a traditional code that can be compiled or executed in the traditional way. Dynamic weaving requires that the compilers or interpreters be aware of the semantics of the aspect-oriented program, so they can adapt the code while running the system.

### 2.3.2 Aspect-Oriented service composition

The need to separate cross-cutting concerns in service composition area has motivated researchers to introduce aspect orientation in this area. Applying aspect orientation in service composition aims at increasing re-usability, reducing the cost of maintenance, and facilitating handling system complexity. Such an approach is investigated and implemented as an extension to Business Process Execution Language (BPEL) [57], called Aspect-Oriented for BPEL (AO4BPEL) [41–43]. This extension is defined based on the SOAP message life-cycle [43] and focuses on service composition.

### 2.3.3 Aspect Orientation in Business Process Modeling

The successful experience of implementing aspect orientation in service composition (AO4BPEL) and the needs in encapsulating cross-cutting concerns in business process modeling inspired researchers to develop techniques to support aspect orientation in Business Process Modeling. Here, we introduce some of these techniques briefly.

Wang et al. [45] introduced a Concern-Oriented Business Process Modeling approach to enable aspect-oriented business process modeling. They claim that quantification and obliviousness [38] are “too strong and hard to consumable for business people”, so they introduced an approach that enables encapsulating these concerns into separate modules. These modules can be executed in parallel with process models. Each process can invoke different concerns using an element called “Losing Control”. The process model can receive the result of the invoked concern through another element called “Gaining Control”. Although they solve the scattering problem by encapsulating cross-cutting concerns, they do not solve the tangling problem since they introduce losing and gaining control elements to be added to the main concern. The approach supports the definition of before, after and around advice. In addition, this approach supports the definition of parallel and nested advice indirectly. However, this approach does not provide any quantification. This approach

can be used for defining, identifying, extracting, assembling and weaving concerns in relation to process models. They have also implemented a prototype as a “proof of concept” to support modeling concern oriented business process models.

Shankardass [46] aims at proposing an extension to BPMN to encapsulate cross-cutting concerns into modules, called “Aspect wrapper”. “Aspect wrapper” solves the scattering problem. To relate these modules to the main process models, a new element is proposed, called “Aspect dot”, which can be placed before and/or after the activities in the main process model. These aspect dots can be related to the aspect wrappers through new elements, called “Aspect flow”. As a result of having these elements, this approach does not solve the tangling problem. The position of the aspect dots determines the type and order of the aspects which are related to the process model. This means that if an advice dot is placed before or after an activity, the related aspect wrapper will be considered as before or after the aspect, respectively. However, if an aspect wrapper is related (through a flow called a “forked aspect flow”) to aspect dots which are placed before and after an activity, the aspect wrapper will be considered as an around aspect. Each activity can have different aspect dots, so their orders determine the precedence of the aspects that should be considered. The approach also supports the definition of parallel advice indirectly. However, this approach neither supports the definition of nested advice nor develops an editor to support the definition of aspect-oriented business process models. It should be noted that it was not clear from the text if the tool (called AMAP) is implemented or only proposed, so we assume that it is only proposed.

Charfi et al. [22] proposed an aspect-oriented extension of BPMN (AO4BPMN). We refer to this approach as AO4BPMN 1.0 to avoid confusion with the extended versions which were proposed subsequently. This approach considers activities and events as join points. It defines pointcuts as constructs for selecting join points which are annotated with text. Advices are defined as subprocesses that encapsulate the functionality of a cross-cutting concern. An advice can have a special activity, called Proceed. This activity can be used to determine the order of the advice, i.e., before, after or around. If the advice does not have a Proceed activity, the type of advice should be specified. Aspects are modeled using pools, which are collections of advice and pointcuts. This approach is supported by a graphical editor, and is used as a foundation for other approaches to extend the level of support for separating cross-cutting concerns in the BPM area. This approach does not provide any quantification.

Cappelli et al. [44] proposed an approach to separate cross-cutting concerns based on BPMN to solve the scattering and tangling problems. They support the definition of an advice to be specified before or after an advised join point. This approach is implemented as an extension to Oryx [58], called

CrossOryx [44], that supports the design of aspect-oriented business process models. The tool supports quantifications over Control-flow, task and data perspectives. It also enables quantifications based on combinations of these perspectives, and it does not consider any of these perspectives as dominant—which is a strong point in quantification.

AO4BPMN 1.0 lacks a concrete pointcut language and weaving mechanism. Jabeen et al. [47] extended AO4BPMN 1.0 by developing a pointcut language using Xtext, and their quantification supports control-flow and task perspectives. They also propose a pseudocode for a weaver to support the weaving of aspect-oriented business process models. However, the actual implementation of the tool is not completed, and it is planned to be done in the future.

Patiniotakis et al. [49] extended the AO4BPMN 1.0 to support aspect composition and weaving. They called their approach AO4BPMN2.0. In addition to three types of aspects (before, after and around), they introduced two other types: replace and bypass. In the replacing scenario, a join point can be replaced by another activity. In the bypass scenario, the join point can be skipped and not executed. They reported that they defined pointcuts using Situation-Action-Networks. The details about their definitions are not clear, yet it seems that they support quantifications over control-flow and task perspectives. Moreover, this approach is supported by an editor and a dynamic weaver that enable designing and enacting aspect-oriented business process models. Their approach also supports the definition of parallelism and precedence in the definition of advice indirectly.

As mentioned earlier, AO4BPMN 1.0 lacks a concrete pointcut language and weaving mechanism. Moreover, it is not based on BPMN 2.0, so it does not have execution semantics. Witteborg et al. [48] extended the approach with a concrete pointcut language and weaving mechanism and to support BPMN 2.0 elements. They call the new extension AO4BPMN 2.0. To avoid confusion with the extension proposed by Patiniotakis et al. [49], we will refer to their approach by their names. The pointcut language is defined using OCL. According to their UML conceptual diagram and example, the pointcut can be composed based on FlowNode elements. These elements are Activity, Event, Gateway, CallActivity, Task and SubProcess. Moreover, the pointcut can be composed based on the Lane information which can be accessed through FlowNode elements. This means that this approach can support the quantification of conditions based on control-flow, task and resource perspectives. In addition, a combination of these perspectives can be used when composing pointcut using different OCL operators. However, it seems that it cannot quantify conditions based on DataObject since their join point does not have any relation to this object in their conceptual diagram. It should be mentioned that only support-

ing the composition of pointcuts using the resource perspective is supported by this approach so far. Furthermore, this approach does not consider any of the business process perspectives as the dominant perspective when defining the pointcuts, which is a strong point in quantification. Their approach is also supported by a tool. They also support the design and enactment of aspect-oriented business process models. The enactment is supported by offering static weaving.

Moreover, different applications of aspect orientation in BPM have been investigated in the literature. Machado et al. [59] investigate how aspect orientation can help to manage process variability. Santos et al. [60] discuss some open issues in aspect-oriented business process modeling, which are classified into three categories: i) how to model them (called method); ii) how to document them (called model); and iii) how to use them (called management). They also investigate how goals can be used to identify aspects in process models [61], and how aspects can be identified using a set of combinations of heuristics [62]. Furthermore, they propose a way to represent the process ownership for aspect-oriented business process models [63]. In addition, Souza et al. [37] propose an approach for identifying services using aspect-oriented business process modeling.

In addition to modeling techniques, some researchers have investigated analysis techniques to query aspects from process models [64–66].



# 3. Research Methodology

This chapter aims to explain and discuss how the research questions can be answered.

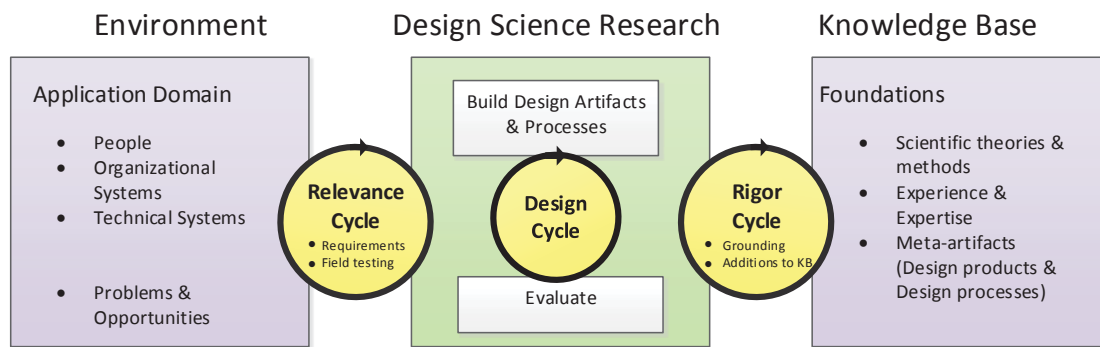
Considering that the tool support is absolutely vital for supporting aspect-oriented modularization [67], this sort of research cannot be solved solely by an empirical study. Indeed, we need the development of tools that enable us to manage cross-cutting concerns in the BPM area. Using these tools, it is then possible to conduct some empirical studies to investigate different aspects of using this technique in different contexts. Thus, the core contribution of this thesis is about developing and demonstrating artifacts to support aspect-oriented modularization in the BPM area. “Design science ... [ in the area of information systems and IT] aims to create novel artifacts in the form of models, methods, and systems that support people in developing, using, and maintaining IT solutions” [68]. Therefore, this thesis falls into the area of design science.

## 3.1 Design Science

According to Hevner and Chatterjee (2010), “Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence” [69].

Design science research aims to improve the environment by building a new artifact. The environment can be considered as the application domain consisting of People, Organizational Systems, and Technical Systems, based on which some insights about problems and opportunities can be discovered (see Figure 3.1). The insight can be expressed in terms of the requirements that justify the need for a new artifact. A new artifact can be built based on the existing body of knowledge as a foundation. This foundation can be in the form of Scientific Theories & Methods, Experience & Expertise, and Meta-Artifacts.

Based on the requirements and the existing body of knowledge, a new artifact can be built. Building an artifact is an incremental process that requires the evaluation of the artifact in each phase. The activities of building and



**Figure 3.1:** Design Science Research Cycles, adopted from [70]

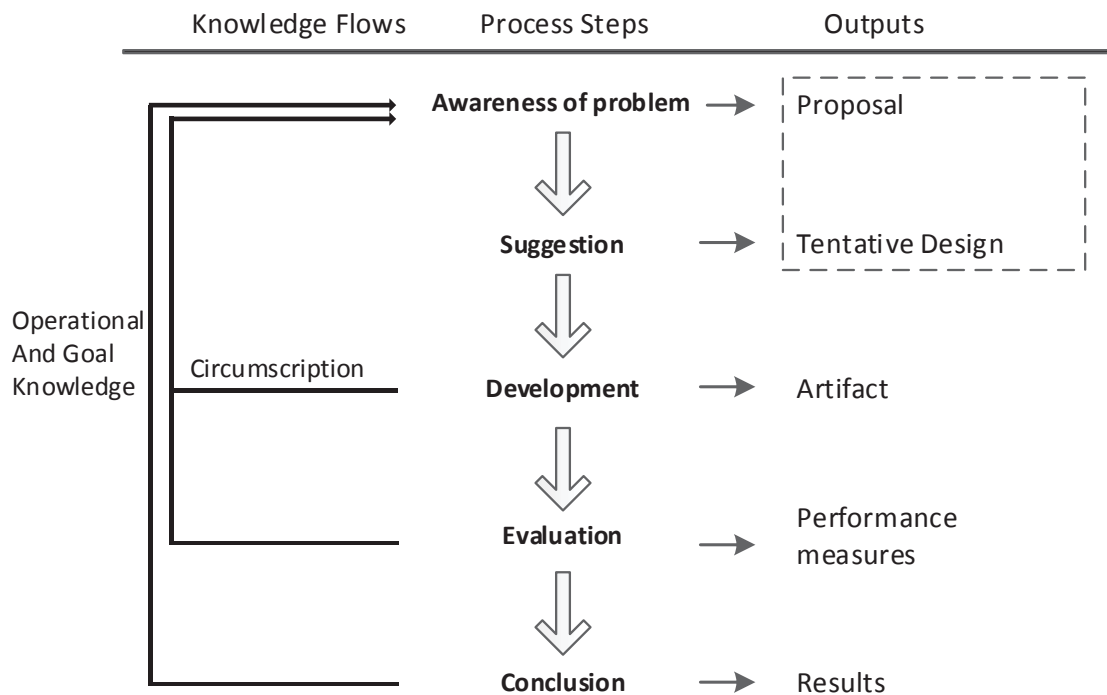
evaluating that result in a new artifact are called the *design cycle*. Design science research usually documents the process of building the artifact. The new artifact can be tested in the environment, and it also extends the existing body of knowledge. Therefore, there are two more cycles that complement the design science cycle, called the relevance cycle and the rigor cycle.

The relevance cycle gathers requirements from the environment to build an artifact and to test the artifact in the environment. The rigour cycle grounds a new artifact on an existing knowledge base and extends the current knowledge by inventing a new artifact. These cycles are identified by Hevner, [70].

A design cycle supports the creation of a new design science artifact. In general, this cycle follows the following reasoning, which has been presented in Figure 3.2, known as the Design Science Research Framework [71]. The central part of the figure shows the different activities that can be performed in a design cycle and their flow. The left side of the figure shows the flow of knowledge between the different activities of the cycle. The right side of the figure specifies the different outputs that are generated in each activity.

The cycle starts by an *Awareness of problem*, where a problem or need is discovered from the environment. The result is a proposal for creating a new artifact to solve the problem. Then, a researcher can suggest a tentative design. An artifact can be developed based on the design. The development can change the understanding of the researcher of the needs, and the whole process can be repeated from the beginning. The artifact can be evaluated, which results in measuring its performance. The result can also change the understanding of the researcher about the needs, and the whole process can be repeated from the beginning. Finally, the evaluation can help a researcher to draw a conclusion regarding the research results. The new result can also initiate new requirements, and the whole process can be repeated from the beginning [71].

Many research choices can be made for each activity. In this thesis, this framework will be employed to build different artifacts. However, each artifact



**Figure 3.2:** Design Science Research Framework, taken from [71]

requires a different choice of method, so the configuration of this framework is different for each artifact. The configuration of the framework is elaborated separately for every paper in related sections.

There are different sorts of artifacts that can be defined in the design cycle. Design science aims to produce artifacts in the form of *constructs*, *models*, *methods*, or *instantiations* [72].

*Constructs* are definitional knowledge in the form of terms, notations, definitions, and concepts that support the formulation of problems and possible solutions, e.g., Unified Modeling Language (UML) [68]. The entity-relationship model is another example of a construct that represents the semantics of data [72].

*Models* are abstractions and representations of possible solutions to some identified problems, and they facilitate the construction of artifacts [68].

*Methods* are prescriptive knowledge specifying how an artifact can be created. This knowledge can, for example, be defined in terms of algorithms and practices [68; 72]. Methods define processes specifying how to search the solution space to solve problems. They can be represented formally (e.g., using algorithms), informally (e.g., using textual descriptions of “best practice”), or some combination [72].

*Instantiations* are systems with embedded knowledge that can be employed in practice, e.g., an implementation of a search algorithm in Java [68]. They show the fact that constructs, models, or methods can be implemented, so they

demonstrate the feasibility of those construct by providing a so-called “proof by construction” [72].

## 3.2 Research methods

This thesis follows the design science research framework to invent new artifacts to address the specified research questions. Figure 3.3 shows the artifacts that have been developed for different research questions. The following subsections explain how the design science research framework is used to answer each research question. The choice of method is justified in each subsection.

### 3.2.1 Formalizing Aspect-Oriented Business Process Models

This section discusses the method based on which we formalized syntax for imperative aspect-oriented business process models to answer the following research question:

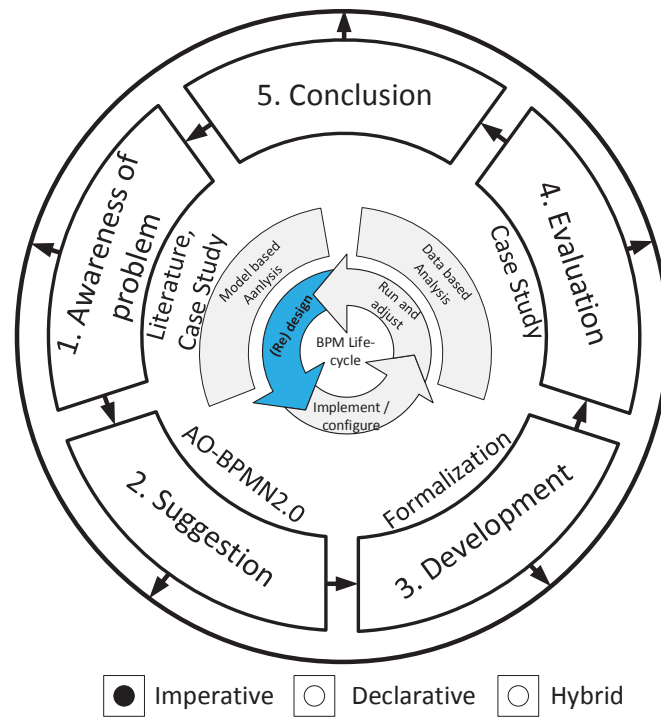
- How should the design of imperative aspect-oriented business process models be formalized based on current informal aspect-oriented techniques?

As mentioned earlier in Chapter 2, there are different constructs defined informally to support the separation of cross-cutting concerns in imperative version of normative and descriptive business process modeling, e.g., [22; 44–47]. However, there is no formal definition as a construct that specifies different pieces of an aspect-oriented business process modeling language. A formal definition is important since it enables one interpretation of the modeling notations, and removes any possible ambiguity.

In [1], we addressed this gap by developing a new artifact. Figure 3.4 shows the application of the design science research framework which is used in that paper. The phases are illustrated in a cycle, and the arrows show the possibilities for moving from one phase to another. In the middle of this figure, we show the BPM lifecycle. The phase of the lifecycle which is addressed by that paper is colored in the figure, which is the design phase for that paper. As is shown below the figure, this approach is relevant for imperative business process modeling. The same settings will be followed by other figures in the following sub-sections.

We have compiled requirements from previous publications (specifically AO4PBPMN [22]), and we have formalized the so-called AO-BPM using BPMN notation [1], labelled as *AO-BPMN 2.0* in Figure 3.3. The artifact is a construct, and it is evaluated using a case study in the banking domain. This





**Figure 3.4:** Application of design science research framework in [1]

artifact removes ambiguity in interpreting different elements of normative and descriptive aspect-oriented business process models based on BPMN notation.

As can be seen from Figure 3.4, we obtained an awareness of this problem through a literature study and a case study. We suggested an extension to BPMN 2.0 to support aspect-oriented business process modeling based on previous research. We formalized our suggestion and evaluated the result through a case study. The decisions for the selection of the methods in each phase are justified as follows.

We became aware of this problem by studying the current literature and applying current techniques to different cases, where we realized there was a gap in supporting the definition of precedence when designing aspect-oriented business process models. We also realized that there was a gap as to the formal definitions for such models. This problem can also be discovered through interviewing experts who have currently applied this technique to real cases. However, it might be very difficult to find such people since the area is new and these techniques have not yet been widely employed in the industry.

We suggested and developed our approach as an extension to BPMN notation. Indeed, there are different business process modeling notations, like Petri nets [27], BPMN [28], YAWL [29], UML [30], EPC, etc. We chose BPMN because it is one of the most widely used notations in modeling business processes. There are also many tools available to support designers in defining their process models in BPMN. In addition, previous research in

aspect-oriented business process modeling are mainly defined based on this notation. Despite selecting BPMN, our conceptualization proposed here is general. Indeed, it can be adapted to extend other modeling notations, such as UML Activity Diagrams, EPC, YAWL, etc.

We evaluated our artifact through a case study in the banking domain. The evaluation was performed in the form of a demonstration where we modeled the business processes in a bank through both BPMN and our approach. To identify and design fairly simple yet representative processes with at least a couple of cross-cutting concerns, we conducted an interview with a domain expert from the bank. Detailed information about the process was derived from a follow-up interview with the same domain expert. The interview was unstructured. The processes could not be discovered through structured or semi-structured interview techniques because we did not have any knowledge about these processes with which to formulate the relevant questions. Thus, the unstructured interview technique was used. Although it is possible to evaluate the approach using expert opinion as an alternative method, a case study is better due to the possibility of showing the relevance of the approach that is used.

We selected small size processes containing approximately 20 activities in our case study to learn the process quickly to be able to present it to an audience that is less knowledgeable about the domain. This selection could have affected our study, in that it could produce small and separated advice. In addition, there are two main limitations to this investigation, that requires additional research to address the related research question completely. First, it did not define formally an executable aspect-oriented business process model, so it was still not possible to define an analysis technique for such models. Second, it was language dependent, so it was not general in a way that can be applied to all imperative business process models. These two limitations were addressed in the subsequent papers.

### 3.2.2 Analysing Aspect-Oriented Business Process Models

This section discusses the method based on which we support the analysis of imperative aspect-oriented business process models to answer the following research questions:

- How should the design of imperative aspect-oriented business process models be formalized based on current informal aspect-oriented techniques? <sup>1</sup>

<sup>1</sup>In [7], we focused on formalizing the definition of an executable version of imperative aspect-oriented business process models.

- How should imperative aspect-oriented business process models be analyzed to check design time problems in business process models?

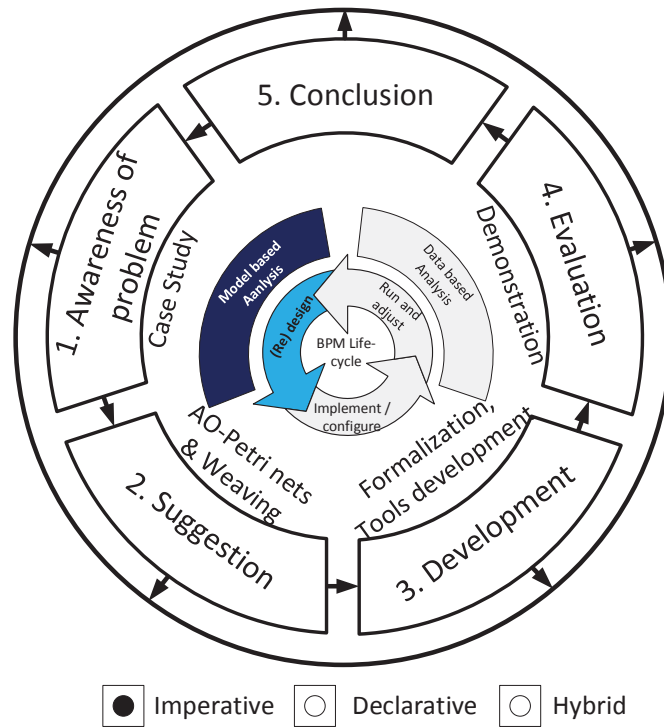
Despite different constructs which are defined to support designing descriptive and normative versions of imperative aspect-oriented business process models, there is no research that supports designing executable versions of these models. The lack of executable models results in a gap in the support for analyzing such models.

In [7], we addressed this gap by developing new artifacts. Figure 3.5 shows the use of the design science research framework in that paper. As can be seen from this figure, we became aware of this problem through modeling different cases. We formulated the formal definitions of executable aspect-oriented business process models using Petri nets [9], referred to as *AO-Petri nets* in Figure 3.3. This artifact is a construct for supporting the definition of executable imperative aspect-oriented business process models. We also developed an algorithm in the form of a method that converts these models into traditional Petri nets models, which can be analyzed using different techniques. It is referred to as *Static Weaving ALG* in Figure 3.3.

In addition, we developed artifacts in the form of instantiations to support designing these models and weaving them into traditional Petri nets models, so they can be analyzed using the existing analysis techniques. These artifacts are called *AO-Petri nets Editor & Static Weaver* (see Figure 3.3). These artifacts are demonstrated using an example, and they support the analysis of imperative aspect-oriented business process models for the first time. The choice of the method in each phase is justified as follows.

We became aware of this problem through case studies in [1; 9] in which we realized that we could not investigate the soundness of our models. It is very difficult for people to analyze a business process model manually, so there is a need for supporting analysis of these models using software. This gap can also be discovered through a systematic literature review, where different phases of support for aspect-oriented business process management can be compared to current publications.

There are some executable business process modeling languages based on which we could define our extension to support executable aspect-oriented business process models, e.g., Petri nets [36], YAWL [29] and BPMN 2.0 [28]. Petri net is a modeling language with a formal definition and semantics. This language is also used to design business processes, and it enables the definition of executable models that can be interpreted by software [24]. It also has three elements, so it is easier to define an analysis technique than it is for BPMN and YAWL, which have many more elements. Therefore, we chose this notation to define executable aspect-oriented business process models.



**Figure 3.5:** Application of design science research framework in [7]

There are different versions of Petri nets, suited to different purposes: Coloured Petri Nets [73], Workflow nets [74], etc. Workflow nets [74] are defined to support the definition of executable business processes. There are different analysis techniques defined based on workflow nets, e.g., Woflan [75]. Therefore, we defined our extension based on workflow nets. We defined an algorithm to support merging the different elements of our executable aspect-oriented business process models into a traditional process model [7]. The result is a traditional workflow net that can be analyzed using existing analysis techniques.

There are different methods of evaluating these artifacts, e.g., expert opinion, case study, and demonstration. The case study is a better method, since it also shows the relevance of the artifact. We demonstrated this artifact through an example. This demonstration shows the ability of these artifacts to reveal design time problems. The details are specified in the included paper, [7]. More evaluation can be performed in the future to study other aspects, such as its relevance, usefulness, etc.

### 3.2.3 Enacting Aspect-Oriented Business Process Models

As mentioned in Chapter 2, supporting aspect orientation means quantifying programmatic assertions over modules written by people who are oblivious to such assertions [38], and there are two approaches to enacting these mod-

els: static weaving and dynamic weaving. In the previous sub-section, we explained how we supported the analysis and enactment of these models through static weaving. In this section, we explain how dynamic weaving can support the enactment of these models.

Enacting aspect-oriented business process models implies quantifying programmatic assertions through *monitoring* the modules and imposing advice when needed through *dynamic weaving*. Thus, this sub-section explain how both monitoring and dynamic weaving can be performed in the BPM area by presenting the papers [2; 9]. To define this mechanism, we need a general definition of aspect-oriented business process model, which is independent of any notation. Thus, we also formulate such definitions, based on which we can define our approach. In particular, we aim to answer the following research questions in this sub-section:

- How should the design of imperative aspect-oriented business process models be formalized based on current informal aspect-oriented techniques? <sup>1</sup>
- How should the enactment procedure (including configuration/Implementation, running and adjustment) of imperative aspect-oriented business process models be formalized?

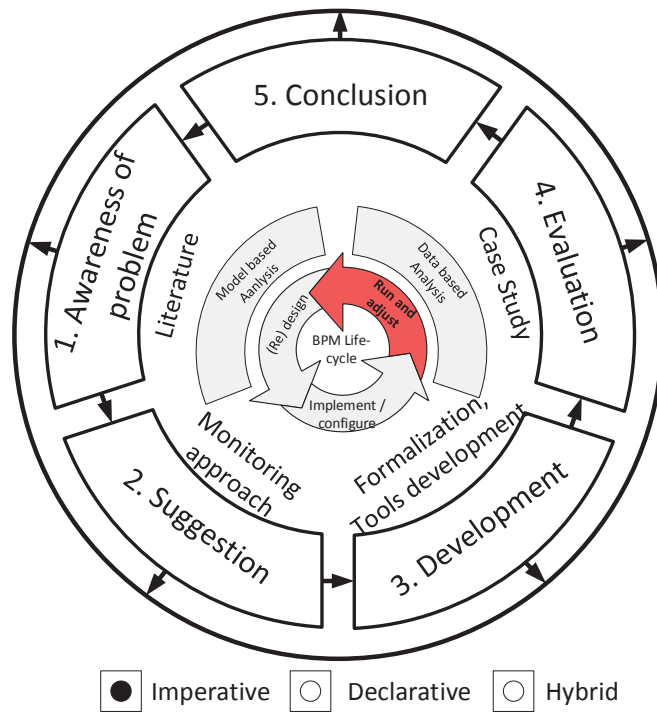
The following sub-sections explain our artifacts for supporting the monitoring and enactment mechanism.

## Monitoring

In this section, we present the method based on which we define how we can monitor instances of imperative business processes at runtime. There are different perspectives for the definition of business process models, and supporting quantification based on different perspectives plays an important role in defining the monitoring method. Figure 3.6 shows the application in [2] of the design science research framework to support monitoring.

In that paper, we formulated the formal definition of the syntax of a process model (as a construct, viz., *MP-BPM* in Figure 3.3), based on which we defined a monitoring algorithm (as a method, viz. *Monitoring ALG* in Figure 3.3). This approach was also implemented (as an instantiation, viz., *RuleEditor & Monitoring Service* in Figure 3.3) to support the definition of the rules and the monitoring of the business process instances. The approach and artifacts were evaluated using a case study. The details are specified in [2]. The choice of the selected method in each phase is justified as follows.

<sup>1</sup>In [9], we focused on formalizing the definition of a general imperative aspect-oriented business process model.



**Figure 3.6:** Application of design science research framework in [2]

To design an effective dynamic weaving solution, a strong quantification mechanism plays an important role. By reviewing the existing literature on aspect-oriented business process management, we realized that in the BPM area there were no formal definitions for such a mechanism.

To suggest a monitoring method, we need to know the monitorable points when enacting a business process instance. Thus, we defined these points based on the definition of the case and workitem definitions in the Workflow Management Reference Model [52]. To suggest a monitoring method, we need a general formal definition of the concept of a business process model. Thus, we defined the formalization. We also need to define how process instances are enacted, in general, so we investigated this issue by considering the specification of a general workflow management system, i.e., the workflow reference model [52]. This comparison helped us to define a monitoring algorithm.

To support such monitoring, we needed to implement our approach in a workflow management system. There are different workflow management systems on the basis of which we could have developed our artifact. We selected YAWL because it is open source. It is also designed based on Service Oriented Architecture, so it is possible to develop a monitoring service on top of it. It also complies with the workflow reference model.

We evaluated our approach through a case study. We interviewed a business expert, and we designed a process model. The interview was unstructured since we had no previous knowledge about the domain. Then, we implemented

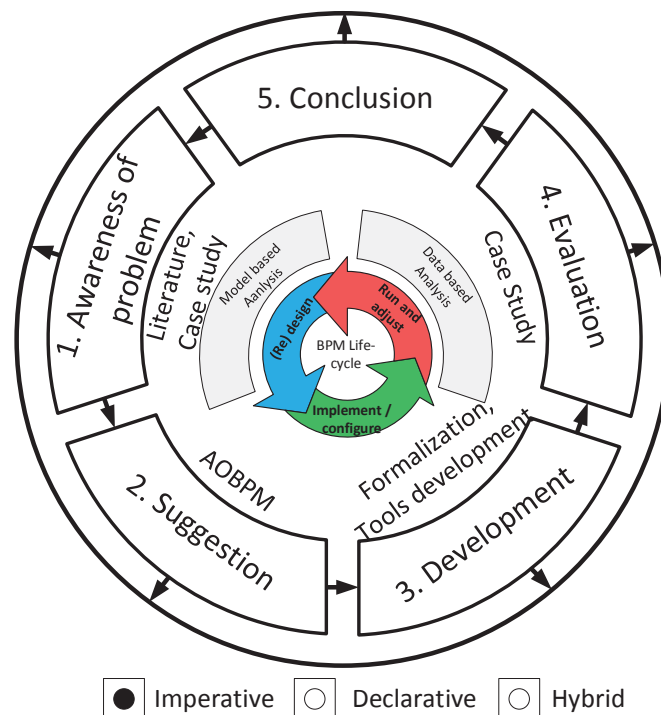
monitoring rules and compared the traditional technique with our approach. As an alternative method, we could have evaluated the approach using expert opinion. Expert opinion is a less effective method for evaluating our approach because it cannot evaluate the relevance of our approach based on a real case.

## Running

In this section, we present the method we used to define how to enact imperative aspect-oriented business process models through dynamic weaving. Figure 3.7 shows the application in [9] of the design science research framework to support the enactment. We prepared both the formal definition of the syntax for a general aspect-oriented business process model and that of the execution semantics for such models. In addition, we implemented the instantiation of our approach in a workflow management system. The results were evaluated through a case study. The details are specified in the included paper, [9]. The decision for the selected method in each phase is justified as follows.

We became aware of this problem through our case study in [1]. We also realized, after a review of the relevant literature, that there was no existing formal definition for supporting the enactment of these models. As an alternative method, one could have detected this gap by interviewing an expert.

To support the core cycle of the BPM lifecycle, we proposed an approach called AOBPM. We formulated a general formal definition of the syntax and



**Figure 3.7:** Application of design science research framework in [9]

execution semantics for the aspect-oriented business process model. The formal definitions were developed as a construct, labelled *AO-BPM* in Figure 3.3. These modeling constructs support the definition of descriptive and normative aspect-oriented business process models. Then, we defined an artifact as a model to define the architecture, components, and specification of a system that supports the enactment of an aspect-oriented business process model, labelled *Dynamic Weaving* in Figure 3.3. This artifact defines the operational semantics for enacting aspect-oriented business process models using Coloured Petri Nets (CPN). CPN was selected because it has a supporting environment where models can be designed, executed, simulated and verified. We implemented a service as an instantiation of the *CPN Model*. The instantiation was developed in YAWL (labelled *Aspect Service* in Figure 3.3), and it extends the functionality of YAWL to supports the enactment of aspect-oriented business process models [9]. Another artifact in the form of an instantiation is also developed (labelled *AO-YAWL Editor* in Figure 3.3) to support definitions of aspect-oriented business process models in YAWL. YAWL was selected because of its advantages that we mentioned in the previous section.

We evaluated the artifacts through a case study. We interviewed the same expert as for preparing [1], for the same processes. We refined the processes, and we also discovered another process that had to be complied with some cross-cutting concerns. The interview was unstructured. There are other ways to have studied this case, such as observation. In that method, a researcher should have access to the field and explore the processes on their own. It is possible to discover processes by studying documents of different organizations, but that is very time-consuming. We implemented, configured, enacted, monitored and adjusted the discovered processes. The result shows that our approach makes it easier to deal with the complexity of cross-cutting concerns by encapsulating them through aspect-oriented business process modeling and enacting them.

Regarding measuring the complexity, there are many different metrics for measuring the complexity of business process models [76]. One basic metric is counting the Number Of Activities (NOA) in a process model, such as the number of Lines of Code (LOC) metric in programming. This metric is popular due to its simplicity, yet it has been criticized as being blind to measuring the functionality and only focusing on one parameter, viz., the size of the module.

This limitation is intended to be addressed by taking the control-flow structure into account when measuring the complexity of the process models by two metrics, viz., “Number Of Activities and Control-flow” (NOAC) and “Number Of Activities, Joins, and Splits” (NOAJS). NOAC can only be applied to well-structured process models [77], while NOAJS can be applied to any

process model. This technique does not consider the number of paths that a module contains. This limitation is addressed by McCabe through a metric called McCabe's Cyclomatic Complexity (MCC) [78]. MCC is adopted for measuring process complexity by Cardoso, known as Control-Flow Complexity (CFC) [79]. This metric measures the complexity of a process model based on the number of XOR, OR, and AND Splits, in such a way that

- For every XOR Split with  $n$  outcomes, the complexity of the process model should be increased by  $n$ .
- For every AND Split, the complexity of the process model should be increased by one.
- For every OR Split with  $n$  outcomes, the complexity of the process model should be increased by  $2^n - 1$ .

There are also other metrics for measuring the complexity of a process model, which have not been used in this research. To measure the complexity of our AS-IS and To-BE process models, we only consider the metrics NOA, NOAJ and CFC, which are commonly used for measuring the complexity of process models based on the Control-flow perspective.

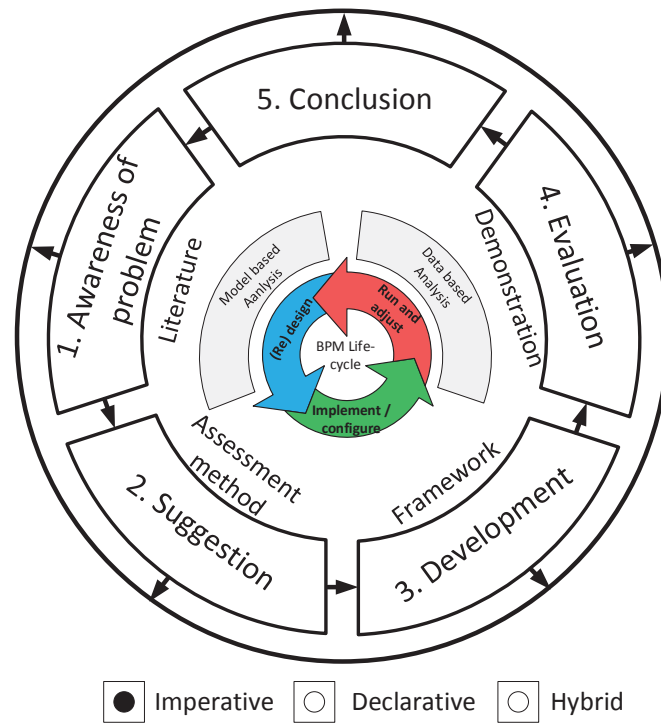
### 3.2.4 Assessing Aspect-Oriented Business Process Models

There are different approaches to supporting imperative aspect-oriented business process modeling, yet there is no method that supports comparing them. Thus, we addressed this gap by developing a new artifact in [4], which address the following research question:

- How should imperative aspect-oriented business process models be assessed in terms of supporting the separation of cross-cutting concerns?

Figure 3.8 shows the application in that paper of the design science research framework. As can be seen from this figure, we became aware of this problem through studying the literature on aspect-oriented business process modeling.

Therefore, we defined a new method to support the comparison of these languages, labelled *Assessment Framework* in Figure 3.3. The method is presented as a framework based on which we can evaluate the strength of these approaches. The framework was tested based on the evaluation and comparison of different approaches through a literature review. This artifact supports the comparison of different AO-BPM languages, so it enables designers to choose the right language to model their business processes. The details are



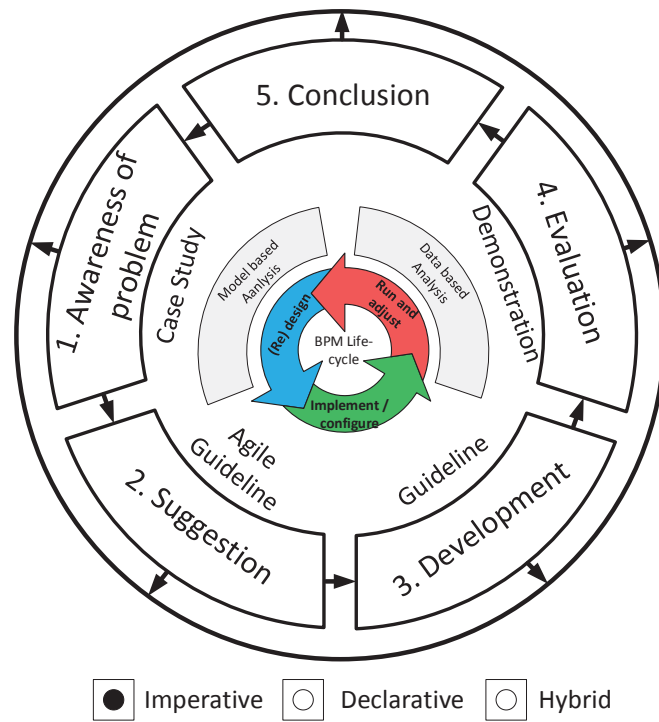
**Figure 3.8:** Application of design science research framework in [4]

specified in the included paper, [4]. The choice of the method selected for each phase is justified as follows.

We became aware of this problem when studying different aspect-oriented business process modeling approaches. We realized that there was no comparison or a method that supports comparing these approaches with each other. As an alternative, one could have noticed this problem during a case study in which the person would want to apply different methods. We compiled the requirements in the form of assessment criteria. We proposed a framework based on which we can compare these approaches with each other. This framework assesses the imperative aspect-oriented business process management approaches by comparing their ability to support the core cycle of the BPM lifecycle. The artifact was used to compare current aspect-oriented business process modeling techniques with each other. As an alternative approach, expert opinion could also have been used to evaluate this framework.

### 3.2.5 Supporting Agile Business Process Development

Separating concerns enables the gradual development of a business process. Indeed, it is possible to develop the core concern, and add others incrementally. The idea of supporting agile business process development through aspect-oriented business process management arose during our case study in [1]. We investigated this idea in [6], which is based on the framework proposed in [80].



**Figure 3.9:** Application of design science research framework in [6]

Thus, we answer the following research question in this section:

- How can aspect-oriented business process management support agile business process development?

Figure 3.9 shows the application in that paper of the design science research framework.

We became aware of the possibility of supporting agile business process development during our case study in [1]. Thus, we extended the framework proposed by Bider in [80], using which we showed how aspect-oriented business process management could support agile business process development. This guideline is labelled *Agile guideline* in Figure 3.3. The result is demonstrated by the use of our previous case study. The details are specified in the included paper, [6].

### 3.2.6 Declarative Aspect-Oriented Business Process Models

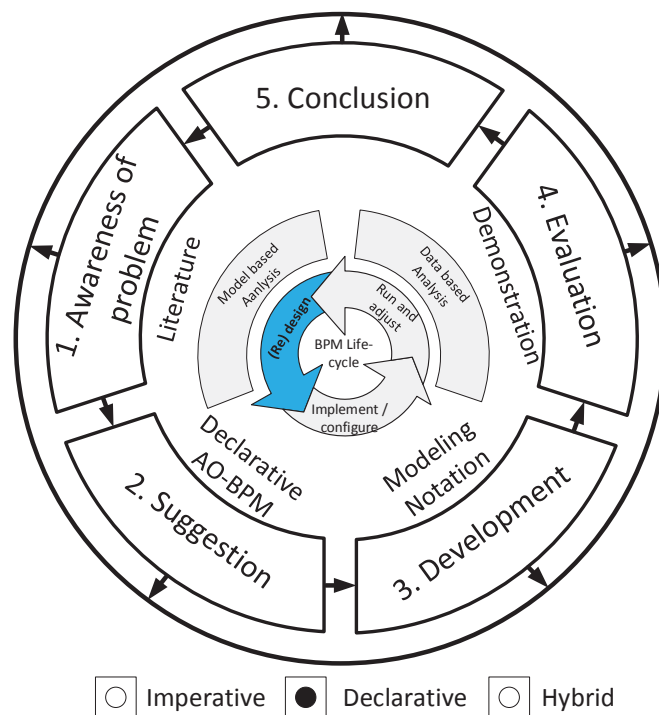
Traditionally, business processes are modeled through imperative modeling approaches where the focus has been rather on providing a high level of support than on a high level of flexibility, as discussed in Chapter 1. The emergence of declarative modeling approaches enabled departing from this trend by proposing constraint-based business process modeling, where the focus is on giving people more flexibility. This paradigm is quite new, and we have not found

anything in the literature proposing a modeling approach to support declarative aspect-oriented business process modeling to provide process participants more flexibility in managing cross-cutting concerns. Thus, we addressed this gap by developing a new artifact in [5], which addresses the following research question:

- How should the separation of cross-cutting concerns be supported by means of declarative aspect-oriented business process models?

Figure 3.10 shows the application in that paper of the design science research framework. As can be seen from this figure, we became aware of this problem through studying current declarative techniques and aspect-oriented business process modeling techniques.

To define how aspect-orientation can be supported to manage business processes using declarative models, we developed a new artifact in the form of a construct. We proposed an approach to support aspect orientation in declarative business processes in [5], labelled *Declarative AO-BPM* in Figure 3.3. We explained how current systems could be extended to support this approach through defining elements in terms of a construct. The idea was demonstrated and discussed based on a tool that supports adaptive case management. The details are specified in the included paper, [5]. The decision for the selected method in each phase is justified as follows.



**Figure 3.10:** Application of the design science research framework in [5]

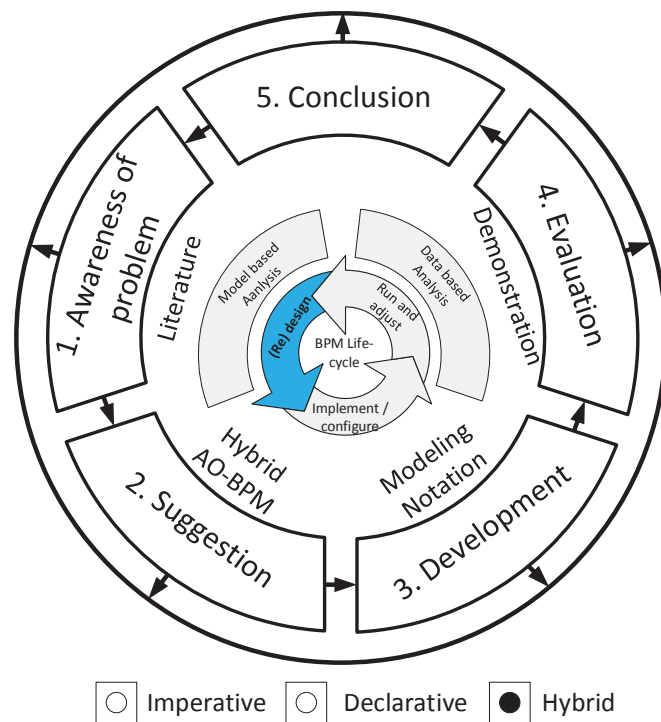
We became aware of this problem when studying different aspect-oriented business process modeling approaches and comparing them with current techniques in declarative aspect-oriented business process modeling. Thus, we proposed a new approach based on the definition of declarative business process modeling to define declarative aspect-oriented business process modeling. We demonstrated our approach through a proposal in an adaptive case management tool.

### 3.2.7 Hybrid Aspect-Oriented Business Process Models

As mentioned in Chapter 1, there is a new paradigm in business process modeling that aims to provide both support and flexibility by combining imperative and declarative models. It is called hybrid modeling. Since it is new in the BPM area, there is as yet no published research on how aspect orientation can be supported through these modeling techniques. Thus, in [8], we aimed to answer the following question.

- How should the separation of cross-cutting concerns be supported by means of hybrid aspect-oriented business process models?

Figure 3.11 shows the application in that paper of the design science research framework. As can be seen from this figure, we became aware of this



**Figure 3.11:** Application of design science research framework in [8]

problem through studying current declarative techniques and aspect-oriented business process modeling techniques.

In that paper, we defined how aspect-orientation can be supported to manage business processes using hybrid models by defining a new construct [8], labelled *Hybrid AO-BPM* in Figure 3.3. In the new approach, imperative advice can be considered for the imperative business processes through declarative rules. The approach was demonstrated through a simple business process. This was the first publication to introduce the use of hybrid models to support the separation of cross-cutting concerns. The details are specified in the included paper, [8].

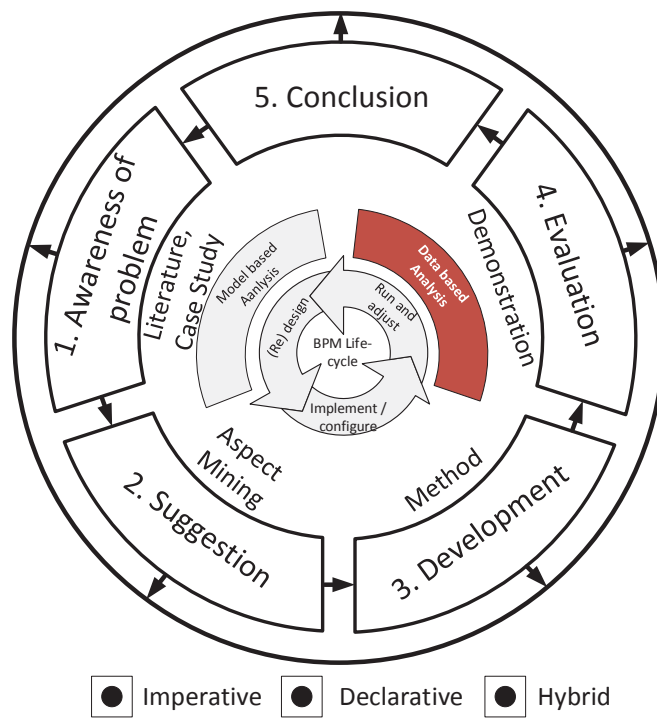
### 3.2.8 Discovering Aspect-Oriented Business Process Models

We have investigated different phases of the BPM lifecycle, except for data based analysis for supporting aspect-oriented business process modeling. In aspect-oriented programming, this area is called *aspect mining*. In [3], we investigated how we can introduce this area in aspect-oriented business process management. In particular, we answered the following research question in that paper:

- How should aspect-oriented business process models be discovered (mined) by analyzing historical data obtained from enacting business processes?

We defined a new method that supports discovering aspect-oriented business process models from event logs, labelled *Mining Method* in Figure 3.3. Figure 3.12 shows the application in that paper of the design science research framework. As can be seen from this figure, we became aware of this problem through studying the literature on aspect-oriented programming. In addition, we tried to discover the process based on the log files of our case study in [13].

To support the discovery of these models, we proposed a method, called aspect mining. We demonstrated the application of the method through applying it to the log files of processes from the banking domain. This artifact supports discovering aspect-oriented business process models from event logs. Therefore, it can reduce the cost of discovering these models through traditional techniques, such as interviews, by supporting their discovery from event logs.



**Figure 3.12:** Application of design science research framework in [3]

## 4. Result

In this chapter, we present a summary of results for each paper.

### 4.1 Formalizing AO-BPM

In this section, we summarize the artifact and results of [1]. The artifact which was developed in that paper is specified in Table 4.1.

**Table 4.1:** The list of artifacts in [1]

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[1]	AO-BPMN 2.0	✓							

We defined the requirements for capturing multiple concerns by studying the existing literature on aspect-oriented modularization and adapting them into the BPM area. The summary of these requirements are listed here:

R1 It should be possible to identify and encapsulate concerns simultaneously.

This requirement implies that both parallel and sequential order of execution should be considered when relating cross-cutting concerns to process models. This means that a process designer should be able to define multiple concerns for a join point in a process model. (S)he should also be able to define the precedence when relating cross-cutting concerns to the process models.

R2 It should be possible to identify and add concerns incrementally at any time during the development lifecycle.

R3 Developers should not be required to know the details of the concerns that do not affect their particular activities.

R4 It must be possible to represent and manage overlapping and interacting concerns.

R5 A mechanism should be defined to support the integration of aspect-oriented business process models to enable their interpretation and enactment by a software system.

We also elaborated on aspect-oriented process modeling by extending AO4BPMN [22]. We formalized our approach by extending the BPMN formal definition in order to remove ambiguity in the definition of normative and descriptive aspect-oriented business process models.

Finally, we demonstrated the relevance of the approach through a case study. A summary of the findings from the case study is given below.

- Aspect orientation enables the documentation of more knowledge in business process modeling by capturing the relation between cross-cutting concerns and specific activities.
- Our approach supports the separation of several concerns that can be defined for an activity in a process model.
- Our approach supports the definition of an ordering of the different cross-cutting concerns.
- Aspect orientation can facilitate dealing with the complexity of business processes by reducing the complexity of models through separating different concerns.
- Aspect orientation can increase the reusability of models since the separated concerns can be re-used several times in the same and other processes.
- Aspect orientation can facilitate the maintenance of the system because the changes only need to be applied in one place. It also removes the overhead of finding scattered concerns in different process models.
- Aspect orientation can support agile business process development by supporting an incremental development of business processes, i.e., the ability to add or change aspects some time after the development of the main process.

The details about the results can be found in [1].

## 4.2 Analyzing Aspect-Oriented Business Process Models

In this section, we summarize the artifacts and results from [7], which makes a formal definition and semantics for executable aspect-oriented business process models. It also supports the analysis of these models by defining a new algorithm to weave the pieces of these models together at design time. The artifacts which were developed in that paper are listed in Table 4.2.

**Table 4.2:** The list of artifacts from [7]

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[7]	AO-Petri nets	✓							
	Static Weaving ALG		✓						
	AO-Petri nets Editor	✓							
	Static Weaver		✓						

We summarize the definitions here, which are based on the definitions of Workflow nets. The details can be found in [7].

**Definition** P/T-nets [81; 82]: A Place/Transition net (P/T-net) is a tuple  $N = (P^N, T^N, F^N)$ ,<sup>1</sup> where:

- $P^N$  is a finite set of *places*;
- $T^N$  is a finite set of *transitions*, such that  $(P^N \cap T^N = \emptyset)$ ; and
- $F^N \subseteq (P^N \times T^N) \cup (T^N \times P^N)$  is a set of *directed arcs*, called the flow relation.

A place is represented by a circle, and a transition is represented by a rectangle. A flow is represented by a directed arrow connecting a place to a transition, or vice versa. The *preset* of an element  $x \in P^N \cup T^N$  in the net  $N$  is defined as  $\bullet^N x = \{\forall y \in (P^N \cup T^N) | (y, x) \in F^N\}$ . The *postset* of an element  $x \in P^N \cup T^N$  in the net  $N$  is defined as  $x^N \bullet = \{\forall y \in (P^N \cup T^N) | (x, y) \in F^N\}$ .  $P$ ,  $T$  and  $F$  represent the universe of all places, transitions and flows, respectively.

The state of a system is determined by a distribution of tokens in the places of the net. The distribution of tokens in the places of a net is called the *marking*

<sup>1</sup>The superscript  $N$  for a set specifies the net that contains the elements of the set.

of the net. The transitions of the net can change the markings of the net. A transition is *enabled* iff there is at least one token in every input place. The result of executing a transition is a new marking, where a token is removed from every input place of the transition, and a token is added to every output place of the transition. The initial state of the system is defined by an initial marking. A marking is called *reachable* iff there exists a sequence of enabled transitions whose firing leads from the initial marking to that marking. The sequence of all executed transition from a marking to another is called the *firing sequence*.

A net is *weakly connected*, or simply connected, iff for every two nodes in the net there are a number of flows and nodes that connect one of the nodes to the other. The net is *strongly connected* iff for every two nodes in the net, there are a number of flows and nodes that connect each of them to the other. A marked net is *bounded* iff the set of reachable markings is finite. It is *safe* iff, for any reachable marking, the number of tokens in every place is less than or equal to one. A transition in the net is called *dead* iff there is no reachable markings such that the transition would be enabled. The net is called *live* iff for every reachable marking  $M'$  and every transition  $t$ , there is another marking  $M''$  which is reachable from  $M'$  which enables  $t$ .

**Definition** Workflow nets [81; 82]: Let  $N = (P^N, T^N, F^N)$  be a P/T-net and  $\bar{t}$  an element not in  $P^N \cup T^N$ .  $N$  is a workflow net (WF-net) iff:  $P^N$  contains an input place  $i$  such that  $\bullet^N i = \emptyset$ ;  $P^N$  contains an output place  $o$  such that  $o^N \bullet = \emptyset$ ;  $\bar{N} = (P^N, T^N \cup \{\bar{t}\}, F^N \cup \{(o, \bar{t}), (\bar{t}, i)\})$  is strongly connected.

A workflow net is sound iff the following four properties hold. i) Safeness: the net with the initial given marking is safe; ii) Proper completion: for any marking that is reachable from the initial marking and has a token in the output place, there is no other token in other places; iii) Option to complete: for any marking that is reachable from the initial marking, there is a reachable marking that has a token in the output place (I call this marking the end marking); and iv) Absence of dead tasks: the net contains no dead transitions.

**Definition** Core net: A core net is a workflow net that encapsulates the core functionality of a business process. The set of all core nets is denoted by  $\mathcal{C}$ . Every task in a core net is called a join point.

**Definition** Advice: An advice  $W = (P^W, T^W \cup \mathbb{P}(\{t_{proceed}^W\}), F^W)$ , is a workflow net that encapsulates a cross-cutting functionality, where:

- $W \notin \mathcal{C}$ . The advice net is not a part of core nets, and

- $t_{proceed}^W$  is a special task in the advice net which is a place holder for a join point in a core net.  $\mathbb{P}$  denotes the power set, so an advice can have zero or one place holder.

An advice  $W$  has zero or one  $t_{proceed}^W$  place holders. An advice with one place holder is called an around advice. The set of all around advices is denoted by  $\mathcal{A}_{\square}^{\bullet}$ . An advice with no place holders is called a free advice. The set of all free advices is denoted by  $\mathcal{A}_{\square}^{\circ}$ . The set of all advices is denoted by  $\mathcal{A} = \mathcal{A}_{\square}^{\bullet} \cup \mathcal{A}_{\square}^{\circ}$ , where  $\mathcal{A}_{\square}^{\bullet} \cap \mathcal{A}_{\square}^{\circ} = \emptyset$ .

**Definition** Pointcut: A pointcut is a function that relates a join point to a set of advice. There are three sort of pointcuts: before, after and around, denoted by  $\mathcal{P}_{\square}$ ,  $\mathcal{P}_{\square\circ}$  and  $\mathcal{P}_{\square}^{\bullet}$ , respectively. The set of all pointcuts is denoted by  $\mathcal{P} = \mathcal{P}_{\square} \cup \mathcal{P}_{\square\circ} \cup \mathcal{P}_{\square}^{\bullet}$ .

- A before pointcut is a function  $\mathcal{P}_{\square} : T \rightarrow 2^{\mathcal{A}_{\square}^{\circ}}$ .
- An after pointcut is a function  $\mathcal{P}_{\square\circ} : T \rightarrow 2^{\mathcal{A}_{\square}^{\circ}}$ .
- An around pointcut is a function  $\mathcal{P}_{\square}^{\bullet} : T \rightarrow 2^{\mathcal{A}_{\square}^{\bullet}}$ .

This means that a before and an after pointcut can relate a set of free advices to a join point, and an around pointcut can relate a set of around advices to a join point. It is also possible that a pointcut relates an empty set of advices to a join point. The join points that are related to at least one advice are called advised join points.

**Definition** Aspect Oriented nets: An Aspect Oriented net (AO-net) is a tuple  $AO = (C, \mathcal{A}, \mathcal{P})$ , where:  $C \in \mathcal{C}$  is a core net.

**Definition** AO-net Soundness: An AO-net  $AO = (C, \mathcal{A}, \mathcal{P})$  is sound iff  $W = staticWeaving(AO)$  is sound. Algorithm 1 defines *staticWeaving*<sup>1</sup>.

This algorithm gets an AO-net and converts it to a Workflow net, so the nets can be analyzed by the techniques available for Workflow nets. It weaves each related advice to an advised join point in the core net. To discover the advised join points, the algorithm investigates whether every task in the core net is an advised join point. If a task  $t$  is an advised join point, the algorithm should weave i) a “before advice” before the task; ii) an “around advice” before and after the task, and iii) an “after advice” after the task. Therefore, it adds

<sup>1</sup>This is a refined version of the static weaving algorithm in [7].

---

**Algorithm 1:** Algorithm for weaving an AO-net

---

```
1 Algorithm staticWeaving( $AO = (C = (T^C, P^C, F^C), \mathcal{A}, \mathcal{P})$ )
2    $W \leftarrow C$ ;
3   foreach  $task\ t\ in\ T^W$  do
4     if  $\mathcal{P}_\bullet(t) \neq \{\}$  then
5        $T^W \leftarrow T^W \cup \{t_{t.split}^W\} \cup \{t_{t.join}^W\}$ ;
6     else
7       if  $\mathcal{P}_{\square}(t) \neq \{\}$  then
8          $T^W \leftarrow T^W \cup \{t_{t.split}^W\}$ ;
9       if  $\mathcal{P}_{\square\circ}(t) \neq \{\}$  then
10         $T^W \leftarrow T^W \cup \{t_{t.join}^W\}$ ;
11      foreach  $Advice\ A'\ in\ \mathcal{P}_{\square}(t)$  do
12         $W \leftarrow addAdvice(W, A', t_{t.split}^W, t)$ ;
13      foreach  $Advice\ A'\ in\ \mathcal{P}_{\square\circ}(t)$  do
14         $W \leftarrow addAdvice(W, A', t, t_{t.join}^W)$ ;
15      foreach  $Advice\ A'\ in\ \mathcal{P}_\bullet(t)$  do
16         $W \leftarrow addAdvice(W, A', t_{t.split}^W, t_{t.join}^W)$ ;
17        foreach  $Place\ p\ in\ \bullet t_{t.proceed}^A$  do
18           $F^W \leftarrow F^W \cup \{(p, t)\} / \{(p, t_{t.proceed}^A)\}$ ;
19        foreach  $Place\ p\ in\ t_{t.proceed}^A \bullet$  do
20           $F^W \leftarrow F^W \cup \{(t, p)\} / \{(t_{t.proceed}^A, p)\}$ ;
21         $T^W \leftarrow T^W / t_{t.proceed}^A$ ;
22    return  $W$ ;
23 Function addAdvice( $W \in \mathcal{C}, B \in \mathcal{A}, initiator \in T^W, terminator \in T^W$ )
24    $A \leftarrow B$ ;
25    $T^W \leftarrow T^W \cup T^A$ ;
26    $P^W \leftarrow P^W \cup P^A$ ;
27    $F^W \leftarrow F^W \cup F^A \cup \{(initiator, i^A)\} \cup \{(o^A, terminator)\}$ ;
28   return  $W$ ;
```

---

a helper split and joins tasks for weaving advice. Thereafter, it weaves before, after and around advices to the net. To weave an around advice, the algorithm should also replace the proceed to the advised join point. Finally, the algorithm returns the woven net. The function *addAdvice* relates an advice to an advised join point, which is used in weaving before, after and around advices.

These definitions were used as a blueprint to implement a tool that supports the definition of aspect-oriented business process models. They are also used to

implement the static weaver to merge these models into traditional workflow nets. The tools were used to evaluate the approach through an example in which the tools enable the analysis of aspect-oriented business process models, and it also enables finding design problems in such models. The details can be found in [7].

### 4.3 Enacting Aspect-Oriented Business Process Models

The results from enacting aspect-oriented business process models are explained in the following subsections about monitoring and running. In the monitoring section, we explain the results that show how different perspectives of business process models can be monitored at runtime. In the running section, we explain how the aspect-oriented business process models can be implemented, configured, run and adjusted at runtime.

#### 4.3.1 Monitoring

In this section, we summarize the artifacts and results of [2]. The artifacts which were developed are listed in Table 4.3.

**Table 4.3:** The list of artifacts in [2]

		Research Questions							
		Imperative					Declarative	Hybrid	Mining
Papers	Artifacts	Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[2]	MP-BPM	✓							
	MonitoringALG			✓					
	RuleEditor			✓					
	Monitoring Service			✓					

We defined the observable points when running business process instances by investigating the case and workitem lifecycles in Workflow management Reference Model [52] (see Figure 2.4 & 2.5). As a result, we defined the relation between the level, states and perspectives in business process models, which is something that can enable us to define monitoring rules. Figure 4.1 shows a summary of these points.

As can be seen from this figure, there are different points in an instance of a business process at which we can monitor the instance from a different perspective. To support the definition of monitoring rules and mechanisms, we

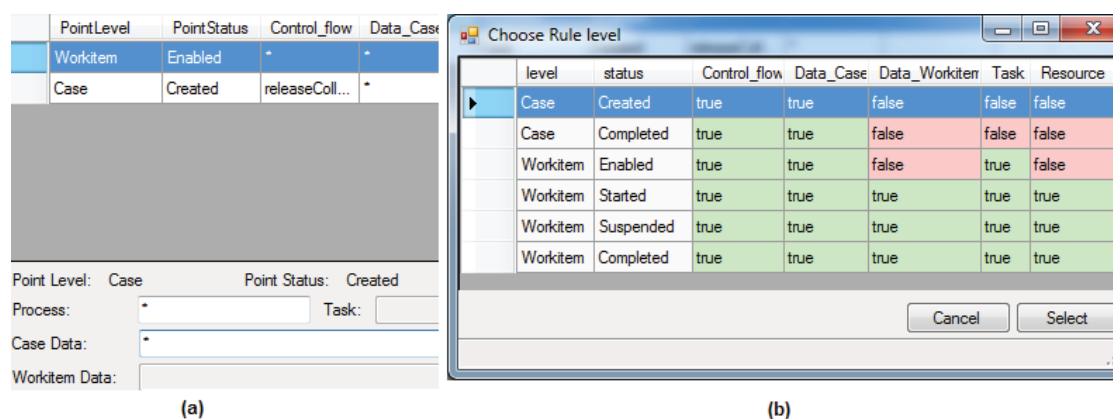
Status	Perspectives				
	Control-flow	Data		Task	Resource
Case Monitoring Lifecycle		Case level	Workitem level		
Created	+	+(r)	-	-	-
Suspended	+	+	-	-	-
Failed	+	+	-	-	-
Completed	+	+(w)	-	-	-
Workitem Monitoring Lifecycle					
Enabled	+	+	-	+	-
Created	+	+	+(r)	+	-
offered to a single resource	+	+	+	+	+
offered to multiple resources	+	+	+	+	+
allocated to a single resource	+	+	+	+	+
Started	+	+	+	+	+
Suspended	+	+	+	+	+
Failed	+	+	+	+	+
Completed	+	+	+(w)	+	+

**Figure 4.1:** The relation between level, states and perspectives

formalized the definition of a general business process model and monitoring mechanism in [2].

These definitions were used as a blueprint to implement a toolset to support the definition and monitoring of business process instances. The toolset includes a rule editor to support defining the rules. It also has a service in YAWL that supports monitoring of the process instances in YAWL. Figure 4.2 shows a screen shot of the rule editor.

Lastly, we investigated the validity and relevancy of the artifacts through a banking case study. The case study showed that different perspectives of a business process instance can be monitored at runtime based on this approach. Details about the results can be found in [2].



**Figure 4.2:** The implemented Rule Editor supporting definition of monitoring rules

### 4.3.2 Running

In this section, we summarize the artifacts and results of [9]. The artifacts which were developed in that paper are listed in Table 4.4.

**Table 4.4:** The list of artifacts in [9]

		Research Questions							
		Imperative					Declarative	Hybrid	Mining
Papers	Artifacts	Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[9]	AO-BPM	✓							
	Dynamic Weaving			✓					
	AO-YAWL Editor	✓							
	Aspect Service			✓					

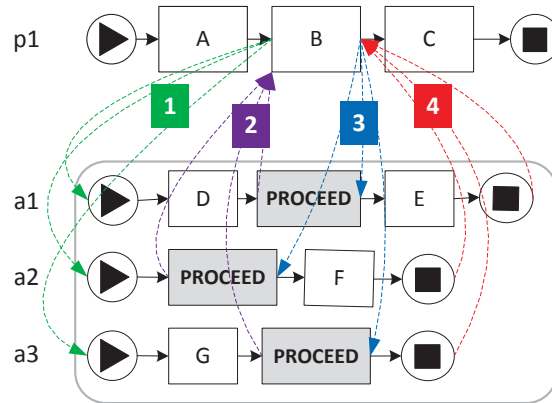
To define the formal operational semantics of aspect-oriented business process models at runtime, we formalized a general definition for aspect-oriented business process models independent of any specific notation.

In this section, we summarize the results of [9], that makes the formal definitions and defines the formal operational semantics for executable aspect-oriented business process models. It clarifies how the implementation, configuration, run and adjustment of these models can be performed using a business process management system.

The details of the syntax of a general aspect-oriented business process model can be found in [9]. Here, we give an overview of the solution, which can be found in more detail in [9].

In order to describe the operational semantics of our weaver, Fig. 4.3 depicts an abstract example for a process model ( $p1$ ) with three advices. This process has three tasks ( $A$ ,  $B$  and  $C$ ), where task  $B$  is related to three advices ( $a1$ ,  $a2$  and  $a3$ ).  $a1$  is an around advice,  $a2$  is an after advice, and  $a3$  is a before advice.

The weaver should enforce the correct execution order of the tasks. In this example, task  $A$  should be followed by a set of tasks in related advices, which should be executed before the advised join point. This set includes all tasks in advices which are positioned before the PROCEED placeholder, viz., tasks  $D$  and  $G$ . The PROCEED holds the place of task  $B$  in this example, so this task should be executed next. The tasks in the advice which are positioned after the PROCEED placeholders,  $E$  and  $F$  should follow task  $B$ . Finally, task  $C$  in the main process should be executed. Using regular expressions, this can be written as  $A(D||G)B(E||F)C$ .

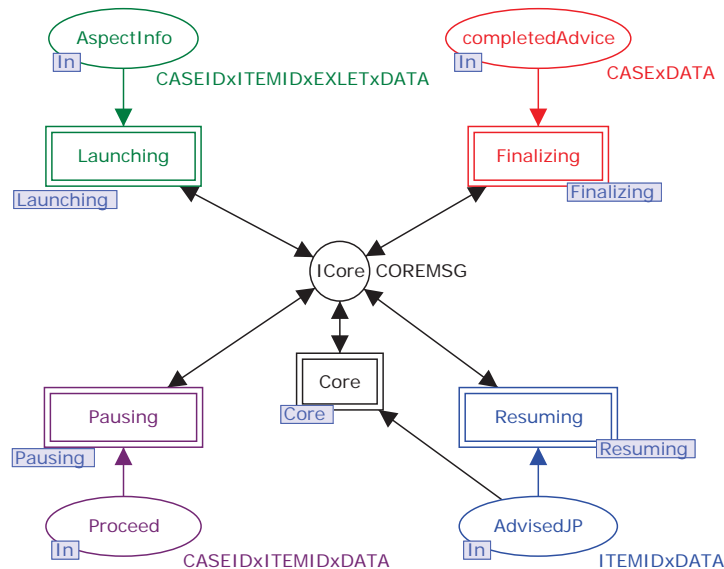


**Figure 4.3:** Dynamic weaving of aspects

The weaving process contains two activities, called *join point selection* and *advice injection*. The first activity selects advised join points by evaluating all join points. The second activity imposes advice in appropriate places regarding the advised join point, i.e., before, after or around it. In this example, all tasks shall be evaluated in the join point selection. The result of the evaluation is task *B*, which has three advices.

To define a general operational semantics for these models, we developed a model using Colored Petri nets. The model contains four main phases, using which an executable aspect-oriented business process model can be enacted, viz., Launching, Pausing, Resuming and Finalizing (see Figure 4.4).

**1** *Launching*: the weaver launches and enforces the enactment of all valid advices before the enactment of their associated advised join point.



**Figure 4.4:** Figure weaveAspect

Each valid advice is determined by the weaver through join point selection.

**2** *Pausing*: the weaver pauses the enactment of an advice when reaching the PROCEED placeholder. It also enables the enactment of the corresponding advised join point when all PROCEED placeholders are reached.

**3** *Resuming*: the weaver enforces the enactment of the rest of advices after enacting the advised join point. It also pauses the enactment of the rest of the main process.

**4** *Finalizing*: the weaver enables the enactment of the rest of the main process when all advices are completed.

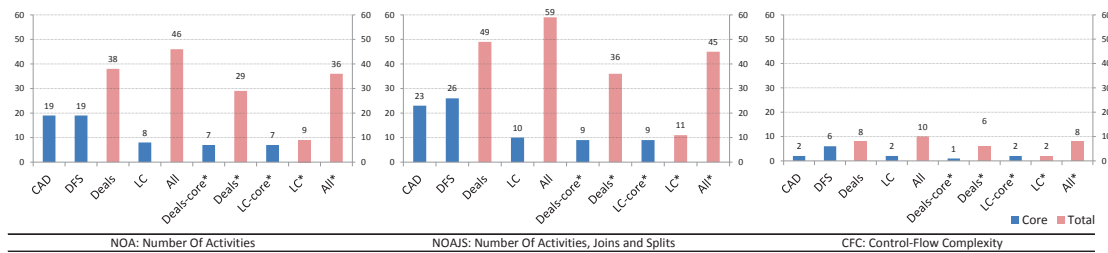
During all these steps, the data is also synchronized between the main process and its advices. These steps require a complex interaction between the weaver and the enactment engine. An overview of the interaction points between the weaver and the enactment engine can be found in [9].

This model can be used as a blueprint for implementing services in Business Process Management Systems to support the implementation, configuration, running and adjustment of aspect-oriented business process models. To show the implementability of such service, we used this model to implement a service in YAWL. In addition, we developed a designer that supports the definition of these models in YAWL. A discussion of the architecture and implementation can be found in [9].

The developed artifacts were used in a case study including three processes in which we measured how our approach supports reducing the complexity of the models. The result confirms the better re-usability, maintainability, documentation, handling process variations, and decreased complexity when employing aspect-oriented business process management.

We calculated the complexity of our AS-IS and To-BE process models using the NOA, NOAJIS and CFC metrics. Figure 4.5 shows the result of measuring the complexity for the AS-IS and TO-BE processes in our case study. *For AS-IS processes*, it shows the complexity of the processes Change Asset Deal (CAD), Deal For Speculation (DFS) and Issue LC (LC). The overall complexity of the Deal processes is shown by Deals, and the complexity of all processes is shown by All. *For TO-BE processes*, it shows the complexity of the Deal core process (Deal-core\*) and the Issue LC core process (LC-core\*). The complexity of the Deal process considering all related advice is shown by Deals\*.

The complexity of the Issue LC process considering all related advice is shown by LC\*, and the complexity of all processes with all advice is shown by All\*. In general, it can be said that the complexity is reduced considerably for the deal processes and all processes. The overall size of the set of models is decreased since the repetitive parts in both processes are modeled once. However, the complexity is not reduced much for the LC process: it even increases if we compare LC with LC\* using the NOA and NOAJS metrics. The reason is that this process is only related to one simple concern with only two tasks, and the process itself is not complex. It should be noted that the process models in the case study are fairly small, so the expected effect of applying aspect-oriented modularization to larger process models will be even more significant.



**Figure 4.5:** Measuring the complexity of AS-IS and TO-BE processes

We also compared our approach to previous ones in terms of supporting different aspects of business process management. We considered four dimensions for the comparison of the different approaches, namely *Crosscutting*, *Quantification*, *Advice Relations* and *Phases Support*.

The *Crosscutting* dimension aims at investigating whether an approach solves the cross-cutting problem, so it investigates whether the proposed technique solves the scattering and tangling problems.

The *Quantification* dimension aims at investigating whether an approach supports quantification. For supporting quantification, having a *Tool Support* is a must. Moreover, the quantification can be based on the different perspectives involved in defining a business process model. We consider *Control-flow*, *Task*, *Data* and *Resource* perspective since they are the minimal perspectives involved in defining a business process model. In addition, quantification can be based on combinations of these perspectives. Finally, it is important that quantifying the conditions does not impose a specific dominant dimension [83]. This means that, for example, a user should be able to define a condition based on a resource perspective without being forced to map it to other perspectives. Such a condition can be expressed as “If a junior dealer performs an activity, do ...”. If a user has to define it for each task, then the quantification itself imposes a scattering of the definitions of the conditions.

The *Advice Relations* dimension aims at investigating how well supported

by this approach is the definition of the different relations between the advices and the main process models. Advices can be defined *Before*, *After* or *Around* an advised join point. Moreover, advice can be defined for an advised join point in parallel to each other. It is possible for advice to be related to another advice, i.e., *Nested*. For example, a security concern itself might need an archive concern to be considered. Finally, advices can have different precedences when defined for an advised join point. For example, a security concern might have higher priority of being applied than an archiving concern.

The *Phases Support* dimension aims at investigating whether an approach supports the separation of cross-cutting concerns in the different phases of the BPM life-cycle. This approach can support the separation of cross-cutting concerns at design time if there is a *Designer* (tool) which enables designing models according to the approach. The models can be woven at design time (*Static Weaving*), or they can be woven at runtime (*Dynamic Weaving*). Business process models can be descriptive, normative or executable [24]. Descriptive models specify what is happening in a business process. Normative models only specify what should happen in a business process. Executable models are those who has a proper execution semantics, so a Business Process Management System can enact them. Static weaving can be performed for all types of models, but the result of static weaving for executable models is the only one that can be enacted. However, weaving models at runtime means that the models can be enacted. Thus, we considered another measure called *Enactment* to see whether the approach supports the enactment of woven models.

The *Phases Support* dimension is not only limited to design and enactment. Indeed, it should be investigated whether the approach supports adjustment as well. Three adjustments can be considered when running an aspect-oriented business process model: Backward, Forward, and Backward–Forward Adjustments [4]. *Backward Adjustment* requires adjusting the running cases and running advices. *Forward Adjustment* requires adjusting the advices for new cases. *Backward–Forward Adjustment* requires adjusting new advices for the running cases. We did not consider the Joinpoint Model (i.e., Signature Exposure in [4]) as an another dimension since all the approaches in the AO-BPM area are based on the control-flow perspective.

The aspect-oriented approaches that support modelling control-flow, task, data and resource perspectives are evaluated based on the defined criteria. These approaches include those proposed by Wang et al. [45], Shankardass [46], Charfi et al. [22] (AO4BPMN 1.0), Cappelli et al. [44], Jabeen et al. [47], Jalali et al. [1] (AOBPMN), Patiniotakis et al. [49], Witteborg et al. [48] and the approach proposed in [9].

Fig. 4.6 shows the result of the evaluation, where the columns of each row represent the different features of an aspect-oriented business process model-

	Crosscutting		Quantification						Advice Relations						Phases Support								
	Scattering	Tangling	Tool Support	Control-flow	Task	Data	Resource	Combinations	Domination	Before	After	Around	Parallel	Nested	Precedence	Designer	Static Weaving	Dynamic Weaving	Enactment	fwd adj	bckwrtd adj	fw-bck	
Wang et al.	+									+	+	+	+	+	+	+							
Shankardass (1)	+									+	+	+	+	+	+	+							
Charfi et al. (AO4BPMN 1.0)	+	+								+	+	+	+	+	+	+							
Cappelli et al. (AO-BPMN)	+	+	+	+	+			+		+	+	+	+	+	+	+							
Jabeen et al.	+	+	+	+	+					+	+	+	+	+	+	+							
Jalali et al. (AOBPMN)	+	+								+	+	+	+	+	+	+							
Patiniotakis et al. (2)	+	+	+	+	+					+	+	+	+	+	+	+			+	+	+	+	+
Witteborg et al.	+	+	+	+	+					+	+	+	+	+	+	+			+	+	+	+	+
Jalali et al. (Aspect Service)	+	+	+	+	+					+	+	+	+	+	+	+			+	+	+	+	+

+ : Directly Supported  
 ! : Indirectly Supported

(1) It is not clear if the tool (called AMAP) is implemented or only proposed.

(2) The details are not given, so it is not possible to judge about different properties.

**Figure 4.6:** Features supported by different approaches

ing approach. The “+” sign in a cell indicates that the approach supports the specific feature directly, and “!” indicates it is supported indirectly.

As can be seen from the figure, our approach solves the scattering and tangling problems. *For Quantification criteria*, it provides a tool to support quantifications based on the control-flow, task and data perspectives. It also supports quantifications based on the combination of these perspectives. However, it does not currently support quantification over the resource perspective. *For Advice Relations criteria*, it also considers the control-flow as a dominant dimension when defining pointcuts. Our approach supports the definition of advice before, around, and after an advised join point. It also supports the definition of parallel advice for an advised join point. However, it does not support nested or precedence requirements. *For Phases Support criteria*, our approach supports the design of aspect-oriented business process models using an editor. It supports the enactment of these models through dynamic weaving. In addition, it supports *forward* and *forward-backward* adjustments. Details about the results can be found in [9].

## 4.4 Assessing Aspect-Oriented Business Process Models

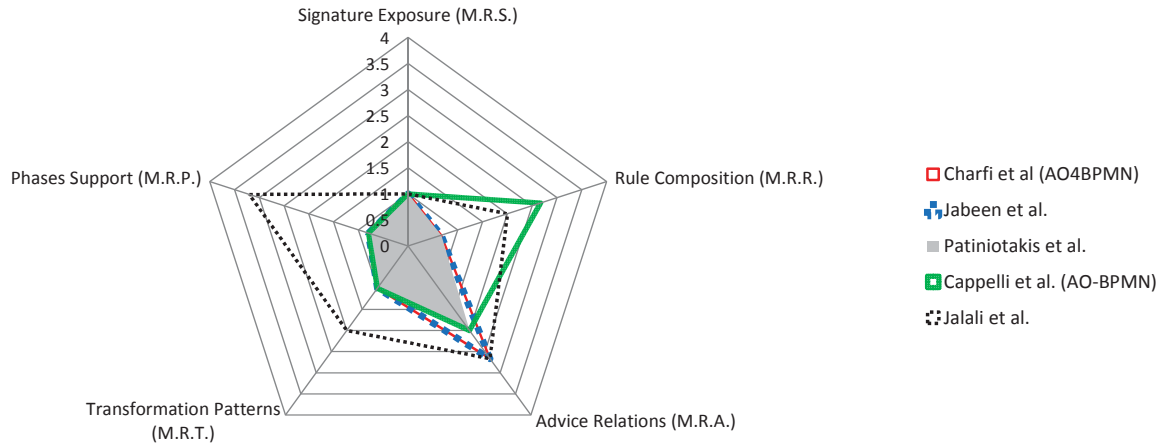
In this section, we summarize the artifact and results of [4]. The artifact which was developed in that paper is specified in Table 4.5.

**Table 4.5:** The list of artifacts in [4]

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[4]	Assessment Framework				✓				

To support the assessment of the aspect-oriented business process management approaches, we compiled the requirements from the existing literature. These requirements are categorized into two sets: basic requirements and measurement requirements.

*Basic requirements* define an including and excluding criteria based on which we can recognize whether an approach can support aspect-oriented business process management. The criteria are composed of three requirements, viz., if the approach supports i) the definition of control-flow, data and resource perspective; ii) addressing the scattering problem, and iii) addressing the tangling problem.



**Figure 4.7:** EvaluationResult

*Measurement requirements* define the criteria based on which we can measure the level of support offered by an aspect-oriented business process management approach. These requirements can help to assess the maturity of an aspect-oriented business process modeling approach, and they are defined based on three issues: i) the strength of the join point selection, ii) the strength of the advice injection and iii) the available support for different phases in the BPM lifecycle. The join point section depends on two sub-issues: *Signature Exposure* and *Rule Composition*. The advice injection depends on two sub-issues: *Pointcut Definitions* and *Transformation Patterns*. Therefore, we categorized these requirements into five dimensions: Signature Exposure, Rule Composition, Advice Relations, Transformation Patterns and Phases Support.

These requirements for each dimension are also defined in more detail in [4]. Each dimension was then measured based on a weighted average of its requirements, and the result was mapped to the scale between 0 and 4. The result is then illustrated in a pentagon that enables the comparison of the results.

The evaluation framework was then used to evaluate different aspect-oriented business process management approach. The result can be seen in Figure 4.7. The details about the evaluated approach and discussions can be found [4].

## 4.5 Supporting Agile Business Process Development

In this section, we summarize the artifact and results of [6]. The artifact which was developed in that paper is specified in Table 4.6.

In [6], we defined how aspect oriented business process management can support agile business process development. That paper extends previous work [80], which elaborated on the agile business process development lifecycle by extending the knowledge transformation perspective along the lines suggested by Nonaka [84]. We realized that having a tool that supports enacting business

**Table 4.6:** The list of artifacts in [6]

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[6]	Agile Guideline					✓			

processes is a requirement for supporting agile business process development. Up to that point, we only have supported the enactment of imperative aspect-oriented business process models, so evaluating the support for agile aspect-oriented business process development was limited to imperative models. We defined a template for assessing tools that support agile business process development. We demonstrated the use of this tool on two different approaches. The approach that is relevant to this thesis is the assessment of how aspect-oriented business process management can support agile business process development. Figure 4.8 shows a summary of this assessment.

As can be seen from this figure, Aspect Oriented Business Process Management partially supports creating the first version of a system in the shortest possible time. This feature is supported due to the separation of some concerns that can be added later. However, the main skeleton cannot be developed in an agile way, so this feature is partially supported. This approach supports full separation of the data and resource perspectives, since they are supported by YAWL notation.

It partially supports *what* and *how* concerns. The *what* concern can be supported by the definition of aspects, which can be related to different goals. It should be mentioned that the notion of underspecification used in aspect-oriented business process modeling partially support this aspect since they can be reconfigured later. The *how* aspect is supported due to the definition of pointcuts. The change in pointcuts that can be made at runtime enables the re-configuration of how different aspects should be considered while enacting a business process.

The *in what order* aspect cannot be supported, since the orders of overall activities are defined in the main process, and the order of aspects can only be defined through pointcuts. Thus, the changes in the order of the main activities still require changes in the process models. The imperative aspect-oriented business process development only supports the definition of the obligatory cross-cutting concerns, so it does not support the *Changing Modality* features, which is something that requires the definition of optional and prohibited con-

Features	Level of support	Comments
1. Creating the first version of a system in the shortest possible time	Partial support	The Aspect Service enables process developers to design the skeleton of the business process. The skeleton consists of the main business process functionality, without cross-cutting concerns like security, privacy, etc. Cross-cutting concerns can be added in later phases.
2. Separation of concerns		Some types of separations below are supported by YAWL, on which Aspect Service has been built, others are implemented by Aspect Service itself.
a <i>What</i>	Partial separation	Process sub-goals are not explicitly modeled in YAWL; however, they may be depicted through underspecified tasks.
b <i>How</i>	Partial separation	Through underspecification tasks supported by YAWL, see above.
c <i>In what order</i>	No separation	As long as Aspect Service is dependent on YAWL, this kind of separation cannot be achieved. Switching to Declare, however, could help in achieving this separation.
d <i>Data</i>	Full separation	The data perspective is separated from process, task and resource perspectives in YAWL.
e <i>By whom</i>	Full separation	The users and organizational model are defined in YAWL separately from process models. Different users can be involved in different processes, and concerns and each process and concern can be handled by different users.
f <i>Changing Modality</i>	No support	Only one sort of Modality is supported - obligation. The definition of aspects makes them necessary to be followed based on specified conditions. Thus, there is no possibility to change the modality.
g <i>Other concerns, e.g. security</i>	Supported	Aspect Service fully supports separation of additional concerns like security, privacy, logging, etc., which is the main objective of the aspect oriented business process management.
3. Usability for non-technical people	No Support	Aspect Service needs to be configured by process administrator who should be quite knowledgeable in the BPM area. Therefore, non-technical users cannot introduce changes themselves.
4. Having an executable model	Supported	Aspect Service has a formal executable semantics in Colored Petri Nets. Also, it has been implemented on the top of YAWL that is based on the concept of executable business process models.

**Figure 4.8:** Assessment of the support for agile business process development by Aspect Service [9]

straints.

This approach completely supports the separation of other concerns, e.g., security. Actually, this is the main goal of this modeling technique. Currently, this approach is not very suitable for non-technical people due to knowledge required for its configuration and enactment. It also completely supports the definition of executable models. Details about the results can be found in [6].

## 4.6 Declarative Aspect-Oriented Business Process Models

In this section, we summarize the artifact and results of [5]. The artifact which was developed in that paper is specified in Table 4.7.

**Table 4.7:** The list of artifacts in [5]

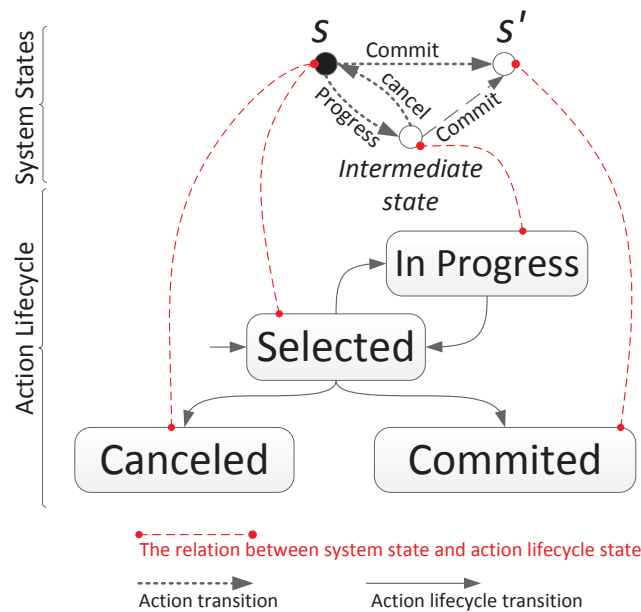
Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[5]	Declarative AO-BPM						✓		

In [5], we defined an approach which specifies how aspect-oriented business process modeling can be supported through declarative business process modeling to enable supporting flexibility in managing business processes. Like other aspect-oriented approaches, this one also includes two mechanisms for both the join point selection and advice injection.

To select a join point, we need to track changes in the state space of a process. Thus, we need to control the movement between states in the system in a way to produce an information history, capture the results of the execution of the tasks, notify people about available tasks, etc. The movement is controlled based on three possible paths: obligatory, optional, and prohibited. To inject advice, the system should be able to alter the potential paths based on the evaluation of the pointcut rules. Thus, we need to define a lifecycle for tasks in such a flexible environment.

Fig. 4.9 shows the states of the action lifecycle and their relations to the system state. An action can be selected, which is shown by the Selected state in the action lifecycle. Selecting an action does not change the state of the system visible to other actions. The results achieved by the selected action are preserved in a temporary, or intermediate state of the system. As soon as the temporal state of the system is changed by the action, the selected action is

considered to be in the In Progress state in the action lifecycle. The selected action can be canceled or committed, which is shown using the Canceled and Committed states in the action lifecycle. The cancellation of an action brings the state of the system to its original state, viz.,  $S$  in the figure. However, the commitment of an action moves the state of the system forward, viz.,  $S'$  in the figure. Furthermore, selecting an action already in progress can be considered as the cancellation of the previous in progress action and selecting it again as a new action. This lifecycle enables the system's moving between states based on the commitment and or cancellation of different actions.



**Figure 4.9:** Action Lifecycle

Selecting actions is governed by rules that allow or prohibit the movement to the state aimed at by the action. We developed a set of declarative rules that support the definition of pointcuts in these systems, which are categorized into Relation and Negation constraints. Relation constraints restrict the ordering of actions by imposing some restrictions on a trace of states. Negation constraints specify negative relations between actions, so they are appropriate to define the prohibited paths in the state space.

Fig. 4.10 shows the relation constraints. A short description of each constraint is specified below.

1. *Responded existence.* The system can commit the join point if and only if action A has occurred before it, i.e., action A is defined as mandatory in the path that goes through the state to be committed. If neither is true, the system should (a) insert rules that make A mandatory after the commitment and commit, or (b) prevent commitment of the join point

Constraint	Meaning	Graphical representation
1. responded existence	if the pointcut rule is met then A should occur either before or after the join point completion	
2. response	if the pointcut rule is met then eventually A occurs after the completion of the join point	
3. alternate response	if the pointcut rule is met then eventually A occurs after the completion of the join point without other occurrences of the join point in between	
4. chain response	if the pointcut rule is met then then A occurs in the next position after completion of the join point	
5. precedence	if the pointcut rule is met then A occurs before completion of the join point	
6. alternate precedence	if the pointcut rule is met then A occurs before completion of the join point without other occurrences of the join point in between	
7. chain precedence	if the pointcut rule is met then A occurs in the next position before the completion of the join point	

**Figure 4.10: Relation Constraints**

- to enforce action A to happen before the commitment, or (c) prevent commitment and require changes in the intermediate (temporary) state that would make A mandatory or allow insertion of the rules that makes A mandatory.
2. *Response restriction.* The system can commit the join point if and only if action A has occurred before it, or action A is defined as mandatory in the path that goes through the state to be committed. If none of the conditions are true, the system should (a) insert rules that make A mandatory after the commitment and commit, or (b) prevent commitment of the join point to enforce action A to happen before the commitment, or (c) prevent commitment and require changes in the intermediate (temporary) state that would make A mandatory or allow insertion of the rules that makes A mandatory.
  3. *Alternate response restriction.* The system should follow the response restriction scenario, except it should also ensure having a prohibition of the same action as the join point's action until action A happens. In case such a prohibition does not exist, the system should add it or not commit the transaction until such a prohibition becomes natural or can be added.
  4. *Chain response restriction.* The system should follow the response restriction scenario, except it should also ensure having a prohibition of any action until action A happens. In case such a prohibition does not exist, the system should add it or not commit the transaction until such a prohibition becomes natural or can be added.
  5. *Precedence restriction.* The system can allow the commitment of the join point if and only if action A has occurred before it. Otherwise, the system should prevent the commitment of the join point to enforce action A to happen before the commitment.
  6. *Alternate precedence restriction.* The system should follow the precedence restriction scenario, except it should also check whether the same action as the join point's action has already happened after action A. In the latter case, the system should cancel the join point's action.
  7. *Chain precedence restriction.* The system can allow the commitment of the join point if and only if action A has occurred exactly before it. Otherwise, the system should prevent commitment of the join point to enforce action A to happen exactly before the commitment.

Fig. 4.11 shows the negation constraints. A short description of each constraint is specified below.

Constraint	Meaning	Graphical representation
1. not co-existence	if the pointcut rule is met then the join point and A cannot occur together	
2. not succession	if the pointcut rule is met then A cannot eventually occur after completion of the join point	
3. not chain succession	if the pointcut rule is met then A cannot occur in the next position after the completion of the join point	

**Figure 4.11:** Relation Constraints

1. *Non-coexistence constraint.* The system should check whether action A has been executed before or not. In case the action has been executed before, the system should cancel the action. Otherwise, the system should check whether a prohibition rule for action A will be in place after commitment. If yes, the action can be committed. Otherwise, the system should add such a prohibition, or, if this is not possible, require changes in the intermediate (temporary) state that would make A prohibited or allow the insertion of a rule that makes A prohibited.
2. *Non-succession constraint.* The system should check whether a prohibition rule for action A will be in place after commitment. If yes, the action can be committed. Otherwise, the system should add such a prohibition, or, if this is not possible, require changes in the intermediate (temporary) state that would make A prohibited or allow insertion of a rule that makes A prohibited.
3. *Non chain-succession constraint.* The system should check whether a prohibition rule for the immediate execution of action A will be in place after commitment. If yes, the action can be committed. Otherwise, the system should add such a prohibition, or, if this is not possible, require changes in the intermediate (temporary) state that would make immediate execution of action A prohibited or allows the insertion of a rule that makes the immediate execution of action A prohibited.

The feasibility of this approach was demonstrated through a possible future development in the *iPB* tool. The details can be found in [5].

## 4.7 Hybrid Aspect-Oriented Business Process Models

In this section, we summarize the artifact and results of [8]. The artifact which was developed in that paper is specified in Table 4.8.

**Table 4.8:** The list of artifacts in [8]

		Research Questions							
		Imperative					Declarative	Hybrid	Mining
Papers	Artifacts	Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[8]	Hybrid AO-BPM							✓	

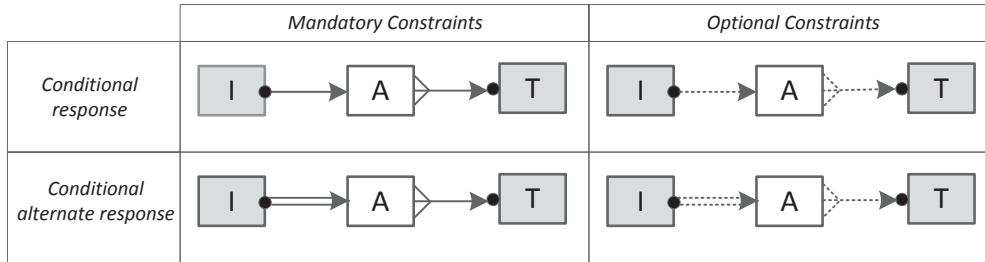
In [8], we proposed an approach to design aspect-oriented business process models using hybrid business process modeling to provide both support and flexibility in managing business processes. Previous work could not support the definition of cross-cutting concerns which need to be enacted in parallel with other parts of the process. Neither could they support the definition of optional cross-cutting concerns.

Therefore, we defined a new modeling language to fulfill this gap. In our approach, the main and cross-cutting concerns are specified using imperative models, and the pointcuts are specified using declarative models. Thus, the approach is hybrid.

In this approach, we defined two placeholders known as initiator and terminator, labelled “I” and “T” in Figure 4.12. There is also another placeholder for the advice, which is labelled “A” in the figure. The mandatory concerns are defined using solid lines; optional ones are defined using dashed lines. We extended the definition of pointcuts to the following:

- support the definition of an *initiator* and a *terminator*. An initiator–terminator pair allows us to define the interval within which the advice can or must be executed. The initiator can refer to the start of a process instance or an instance of any task in the main process. The terminator can refer to the end of a process instance or an instance of any task in the main process;
- support the definition of both optional and mandatory advice.

An instance of the initiator can be followed by a terminator with or without other occurrences of the initiator in between. Thus, we can define the advice to be executed either i) for every occurrence of the initiator until the occurrence of



**Figure 4.12:** Proposed approach for definition of pointcuts

the terminator, or ii) for the first occurrence of the initiator until the occurrence of the terminator. The first option is called an *alternate response* scenario and the second one a *response* scenario.

Considering that advice can be defined as optional or mandatory, pointcuts can be defined through four declarative rules, called i) the mandatory conditional response (mcr); ii) the mandatory conditional alternate response (mcar); iii) the optional conditional response (ocr); and iv) the optional conditional alternate response (ocar). The graphical representations of these rules are illustrated in Figure 4.12.

We demonstrated the use of this technique on a simplified business process from the education domain. The result shows that our approach can deal with the definition of i) cross-cutting concerns that can be defined in parallel with (a part of) a process model; and ii) both mandatory and optional cross-cutting concerns. The details can be found in [8].

## 4.8 Discovering Aspect-Oriented Business Process Models

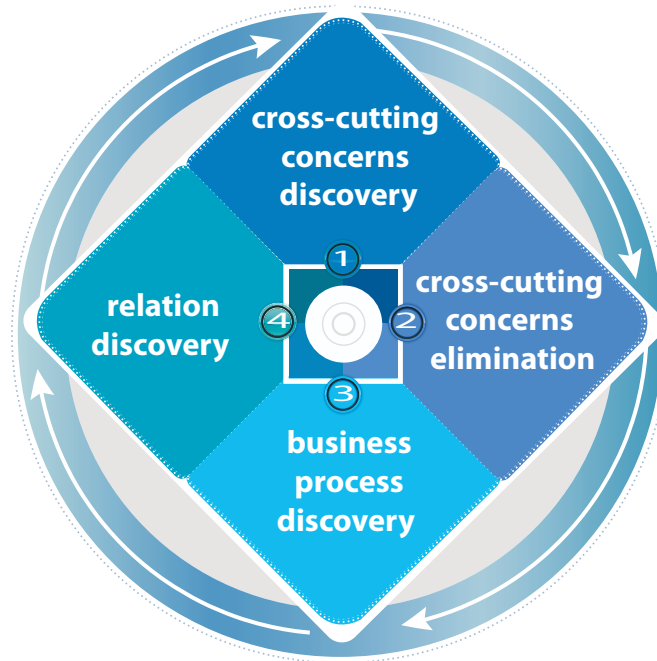
In this section, we summarize the artifact and results of [3]. The artifact which was developed in that paper is specified in Table 4.9.

**Table 4.9:** The list of artifacts in [3]

Papers	Artifacts	Research Questions							
		Imperative					Declarative	Hybrid	Mining
		Design	Analysis	Enactment	Assessment	Agility	Design	Design	
[3]	Mining Method								✓

In [3], we proposed a method to discover process models from event logs. The method includes four steps: i) cross-cutting concerns discovery, ii) elimination of cross-cutting concerns, iii) business process discovery, and iv) re-

lation discovery (see Figure 4.13). This method can be followed manually or automatically. Its automation requires the development of algorithms and tools, so we present the method of that paper manually.



**Figure 4.13:** Aspect Mining Approach for BPM

This method was applied to a log file resulting from enacting aspect-oriented business process models based on a banking case study. Details about the results can be found in [3].

# 5. Conclusion and Future work

## 5.1 Conclusion

This thesis has investigated how aspect orientation can be supported in business process management using information technology. It has mainly focused on how this support can be given based on *imperative* business process models. It has also investigated how the level of support can be extended using *declarative* and *hybrid* models. It also defined how these models can be discovered from event logs. The thesis has used the framework of design science, and the result has been presented as a set of artifacts and their demonstrations fulfilling different aspects of supporting aspect orientation in the Business Process Management (BPM) area.

The support of *imperative* aspect-oriented business process models has been defined based on different aspects, including how these models can be designed, be analyzed, be enacted, be monitored, be assessed, and support agility in the BPM area.

The *design* of an aspect-oriented business process model was investigated by developing artifacts in the forms of constructs and instantiations that support modeling aspect-oriented business processes. To define the elements of a descriptive or normative aspect-oriented business process modeling approach, a construct in the form of formalizing the syntax of a language using BPMN is developed that defines all the elements required to design such a model. The construct was evaluated through a case study. In addition, to define the elements of an executable aspect-oriented business process modeling approach, a construct in the form of formalizing the syntax of a Petri net was developed, which defines all elements required to design an executable version of such model. This construct enables the interpretation of these models by information systems. An aspect-oriented Petri net designer was developed as an instantiation to prove the applicability of this artifact, which was used subsequently to give an example for the purpose of analysis. Furthermore, a general and language-independent formal definition of the syntax of such a process model was defined to facilitate the definition of an executable model based on which an Aspect-Oriented Business Process Management Systems could be built. This artifact was used to develop an editor that supports designing

aspect-oriented business process models in YAWL, which was used in a banking case study to design process models.

The *analysis* of an aspect-oriented business process model was carried out by developing artifacts in the forms of a method and an instantiation. The method was developed in the form of an algorithm for aspect-oriented business process models in the notation of Petri nets. The algorithm was implemented as an artifact that merges all fragmentations of the process models and results in a traditional process model. Therefore, the traditional analysis techniques could be applied. The result was demonstrated through a case that showed how problems like deadlock can be discovered through analyzing these models. These artifacts define the executable aspect-oriented business process modeling, and define a static weaving algorithm. The result enables the analysis of these models using existing analysis technique to remove design problems from these models.

The *enactment* of an aspect-oriented business process model was investigated by developing artifacts in the forms of a model and an instantiation. The model was developed using Colored Petri Nets, which was used as a blueprint to implement an instantiation in YAWL. The instantiation was used to evaluate the approach using a case study in the banking domain. The result enabled the formulation of aspect-oriented business process models. This shows how aspect-oriented business process management can support people in dealing with the complexity of process models.

The *monitoring* of an aspect-oriented business process model was investigated by developing artifacts in the forms of a construct, a method and instantiations. The basic elements were formalized as a construct that defines a general process model. The construct was used in the definition of a method in the form of an algorithm that specifies how business processes can be monitored by considering different perspectives involved in defining them. An instantiation of the algorithm was developed in YAWL, and was used to evaluate the approach through a case study. The result enabled the definition of better pointcuts to encapsulate cross-cutting concerns in business processes. The findings show that it improves the reusability and facilitates the maintenance of such systems.

The *assessment* of approaches that support aspect-oriented business process management were investigated by developing an artifact in the form of a method. That method investigates different aspects that an aspect-oriented business process management approach can support. The result was presented as a framework that enables comparing these approaches. The framework was applied to existing approaches, and the results were presented. The results show the pros and cons of existing approaches, which can be used as guidance to select the appropriate tool.

The discovery of aspect-oriented business process models from event logs, called *aspect mining*, was investigated by developing an artifact in the form of a method. This method defines how aspect-oriented business process models can be discovered from event logs with the help of existing process mining algorithms. The method was demonstrated by applying it to the event logs of a banking process model. The result shows that these models can be mined from event logs, which reduces the cost of discovering business processes.

Supporting *agile* business process development using aspect-oriented business process development has been investigated by developing an artifact in the form of a method. This method was defined based on a theory that describes how business processes can be developed in an agile way. This theory was developed by refining the theory of knowledge creation. The method was applied using the artifact that we developed for enacting aspect-oriented business process models in YAWL. The result shows how aspect-oriented business process management can support agile business process development.

Furthermore, it was investigated how the support for aspect-orientation can be extended by declarative and hybrid business process models.

The support of *declarative* aspect-oriented business process model was investigated by developing an artifact in the form of a construct. This construct defines elements and rules that can support declarative aspect-oriented business process management. This construct was demonstrated based on a potential application of this approach to an existing tool, called *iPB*. The result shows how this modularization technique can be used to encapsulate cross-cutting concerns in a more flexible environment.

Aspect-oriented business process management using *hybrid* models was investigated by developing artifacts in the form of a construct. That artifact was used to demonstrate how hybrid models can support the separation of cross-cutting concerns in the BPM area. The result enabled definitions of more constraints to encapsulate cross-cutting concerns in the BPM area, which can result in better reusability and maintenance.

## 5.2 Future research

This thesis has defined different aspects of supporting aspect orientation using information systems. The result not only provides support for aspect-oriented business process management, but it also opens many potential ways for future research.

These tools enable researchers to manage cross-cutting concerns in the BPM area, so it will be interesting to perform a more empirical study on how this approach can help organizations address different aspects.

We defined how cross-cutting concerns can be modeled using declarative and hybrid modeling approaches to support business processes in a more flexible environment. However, the support for other phases is still open for future investigation.

In aspect mining, we defined how aspect-oriented business process models can be discovered from event logs. Our finding opens an area that can be further investigated. Indeed, the development of algorithms that can support automation of discovering such models can be considered as an interesting future direction.

# References

- [1] A. Jalali, P. Wohed, and C. Ouyang, "Aspect oriented business process modelling with precedence," in *Business Process Model and Notation* (J. Mendling and M. Weidlich, eds.), vol. 125 of *Lecture Notes in Business Information Processing*, pp. 23–37, Springer Berlin Heidelberg, 2012. xi, 29, 30, 50, 52, 54, 58, 59, 61, 62, 67, 68, 79
- [2] A. Jalali and P. Johannesson, "Multi-perspective business process monitoring," in *Enterprise, Business-Process and Information Systems Modeling* (S. Nurcan, H. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin, and I. Bider, eds.), vol. 147 of *Lecture Notes in Business Information Processing*, pp. 199–213, Springer Berlin Heidelberg, 2013. xi, 29, 30, 56, 57, 73, 74
- [3] A. Jalali, "Aspect mining in business process management," in *Perspectives in Business Informatics Research* (B. Johansson, B. Andersson, and N. Holmberg, eds.), vol. 194 of *Lecture Notes in Business Information Processing*, pp. 246–260, Springer International Publishing, 2014. xi, 29, 30, 65, 66, 91, 92
- [4] A. Jalali, "Assessing aspect oriented approaches in business process management," in *Perspectives in Business Informatics Research* (B. Johansson, B. Andersson, and N. Holmberg, eds.), vol. 194 of *Lecture Notes in Business Information Processing*, pp. 231–245, Springer International Publishing, 2014. xi, 30, 60, 61, 79, 81, 82
- [5] A. Jalali and I. Bider, "Towards aspect oriented adaptive case management," in *Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 IEEE 18th International*, pp. 143–151, Sept 2014. xi, 30, 63, 85, 89
- [6] I. Bider and A. Jalali, "Agile business process development: why, how and when-applying nonaka's theory of knowledge transformation to business process development," *Information Systems and e-Business Management*, pp. 1–39, 2014. xii, 30, 61, 62, 82, 83, 85
- [7] A. Jalali, "Static weaving in aspect oriented business process management," in *Conceptual Modeling* (P. Johannesson, M. Lee, S. Liddle, A. Opdahl, and O. Pastor López, eds.), vol. 9381 of *Lecture Notes in Computer Science*, pp. 548–557, Springer International Publishing, 2015. xii, 30, 53, 54, 55, 69, 71, 73
- [8] A. Jalali, F. M. Maggi, and H. A. Reijers, *Enhancing Aspect-Oriented Business Process Modeling with Declarative Rules*, pp. 108–115. Springer International Publishing, 2015. xii, 30, 31, 64, 65, 90, 91
- [9] A. Jalali, C. Ouyang, P. Wohed, and P. Johannesson, "Supporting aspect orientation in business process management," *Software & Systems Modeling*, pp. 1–23, 2015. xii, 30, 31, 54, 56, 58, 59, 75, 77, 79, 81, 84
- [10] A. Jalali, "Foundation of Aspect Oriented Business Process Management," Master's thesis, Stockholm University, 2011. xiii
- [11] A. Jalali, P. Wohed, and C. Ouyang, *Operational Semantics of Aspects in Business Process Management*, pp. 649–653. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. xiii

- [12] A. Jalali, "Service oriented modularization using coloured petri nets," in *AWPN 2012: Algorithmen und Werkzeuge für Petrinetze Sep 13, 2012-Sep 14, 2012 University of Vienna, Austria*, 2012. xiii
- [13] A. Jalali, P. Wohed, C. Ouyang, and P. Johannesson, *Dynamic Weaving in Aspect Oriented Business Process Management*, pp. 2–20. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. xiii, 65
- [14] I. Bider, A. Jalali, and J. Ohlsson, *Adaptive Case Management as a Process of Construction of and Movement in a State Space*, pp. 155–165. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. xiii
- [15] A. Jalali, "Supporting enactment of aspect oriented business process models: an approach to separate cross-cutting concerns in action," 2013. xiii
- [16] A. Jalali, *Enacting Aspect Oriented Business Process Models*. Globe On-Demand, LLC, 2014. xiii
- [17] I. Bider, A. Jalali, and D. Söderström, *Creating Self-managed Cross-Professional Teams with Metaphoric Business Process Support Systems*, pp. 19–33. Springer International Publishing, 2015. xiv
- [18] I. Bider and A. Jalali, "Limiting variety by standardizing and controlling knowledge intensive processes," in *IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, IEEE, 2016. xiv
- [19] A. Jalali, "Reflections on the use of chord diagrams in social network visualization in process mining," in *Research Challenges in Information Science (RCIS)*, IEEE, 2016. xiv
- [20] A. Jalali, "Supporting social network analysis using chord diagram in process mining," in *Perspectives in Business Informatics Research* (V. Repa and T. Bruckner, eds.), vol. 261 of *Lecture Notes in Business Information Processing*, Springer International Publishing, 2016. xiv
- [21] M. Weske, *Business process management: concepts, languages, architectures*. Springer Science & Business Media, 2012. 19, 20, 21, 33, 35
- [22] A. Charfi, H. Müller, and M. Mezini, "Aspect-oriented business process modeling with ao4bpmn," in *Modelling Foundations and Applications* (T. Kühne, B. Selic, M. Gervais, and F. Terrier, eds.), vol. 6138 of *Lecture Notes in Computer Science*, pp. 48–61, Springer Berlin Heidelberg, 2010. 19, 26, 28, 43, 50, 68, 79
- [23] M. zur Muehlen and D. T.-Y. Ho, "Risk management in the bpm lifecycle," in *Business process management workshops*, pp. 454–466, Springer, 2006. 20
- [24] W. Aalst, "Business process management: A comprehensive survey," *ISRN Software Engineering*, vol. 2013, 2013. 20, 54, 79
- [25] M. Pestic, *Constraint-based workflow management systems: shifting control to users*. PhD thesis, Technische Universiteit Eindhoven, 2008. 21, 22
- [26] M. La Rosa, P. Wohed, J. Mendling, A. ter Hofstede, H. Reijers, and W. M. P. Van der Aalst, "Managing process model complexity via abstract syntax modifications," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 614–629, 2011. 21, 23
- [27] C. Petri, "Introduction to general net theory," in *Net theory and applications*, pp. 1–19, Springer, 1980. 21, 52
- [28] I. O. Object Management Group, "Business Process Model and Notation (BPMN)," tech. rep., Object Management Group, Inc. (OMG), 2013. 21, 52, 54
- [29] W. Aalst, L. Aldred, M. Dumas, and A. Hofstede, "Design and implementation of the yawl system," in *Advanced Information Systems Engineering* (A. Persson and J. Stirna, eds.), vol. 3084 of *Lecture Notes in Computer Science*, pp. 142–159, Springer Berlin Heidelberg, 2004. 21, 52, 54

- [30] J. Rumbaugh, I. Jacobson, and G. Booch, "The unified modeling language reference manual," 1999. 21, 52
- [31] C. W. Günther and W. M. P. van der Aalst, *Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics*, pp. 328–343. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. 22
- [32] W. M. P. van der Aalst and C. W. Gunther, "Finding structure in unstructured processes: The case for process mining," in *Application of Concurrency to System Design, 2007. ACSD 2007. Seventh International Conference on*, pp. 3–12, July 2007. 22
- [33] S. Debois, T. Hildebrandt, and T. Slaats, "Hierarchical declarative modelling with refinement and sub-processes," in *Business Process Management* (S. Sadiq, P. Soffer, and H. Völzer, eds.), vol. 8659 of *Lecture Notes in Computer Science*, pp. 18–33, Springer International Publishing, 2014. 22
- [34] M. Westergaard and T. Slaats, "Mixing paradigms for more comprehensible models," in *Business Process Management* (F. e. a. Daniel, ed.), vol. 8094 of *LNCS*, pp. 283–290, Springer, 2013. 22
- [35] W. Aalst, M. Adams, ter A.H.M. Hofstede, M. Pesic, and H. Schonenberg, "Flexibility as a service," in *Database Systems for Advanced Applications* (L. e. a. Chen, ed.), vol. 5667 of *LNCS*, pp. 319–333, Springer, 2009. 22
- [36] M. Adams, *Facilitating Dynamic Flexibility and Exception Handling for Workflows*. PhD thesis, Faculty of Information Technology, Queensland University of Technology, November 2007. 24, 54
- [37] A. Souza, C. Capelli, F. Santoro, L. Azevedo, J. Sampaio do Prado Leite, and T. Batista, "Service identification in aspect-oriented business process models," in *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*, pp. 164–174, 2011. 26, 45
- [38] R. E. Filman and D. P. Friedman, "Aspect-oriented programming is quantification and obliviousness," in *Workshop on Advanced separation of Concerns, OOPSLA*, vol. 2000, 2000. 26, 41, 42, 55
- [39] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. Loingtier, and J. Irwin, "Aspect-oriented programming," in *ECOOP'97 - Object-Oriented Programming* (M. Aksit and S. Matsuoka, eds.), vol. 1241 of *Lecture Notes in Computer Science*, pp. 220–242, Springer Berlin Heidelberg, 1997. 26
- [40] A. Rashid, A. Moreira, and J. Araújo, "Modularisation and composition of aspectual requirements," in *Proceedings of the 2nd international conference on aspect-oriented software development, AOSD '03*, (New York, NY, USA), pp. 11–20, ACM, 2003. 26
- [41] A. Charfi and M. Mezini, "Aspect-oriented web service composition with ao4bpel," in *Web Services* (L.-J. Zhang and M. Jeckle, eds.), vol. 3250 of *Lecture Notes in Computer Science*, pp. 168–182, Springer Berlin Heidelberg, 2004. 42
- [42] A. Charfi and M. Mezini, "Ao4bpel: An aspect-oriented extension to bpel," *World Wide Web*, vol. 10, no. 3, pp. 309–344, 2007.
- [43] A. Charfi, *Aspect-oriented workflow languages: AO4BPEL and applications*. PhD thesis, der Technischen Universität Darmstadt, Darmstadt, November 2007. 26, 42
- [44] C. Cappelli, F. Santoro, J. do Prado Leite, T. Batista, A. Medeiros, and C. Romeiro, "Reflections on the modularity of business process models: The case for introducing the aspect-oriented paradigm," *BPM Journal*, vol. 16, pp. 662–687, 2010. 26, 28, 43, 44, 50, 79
- [45] J. Wang, J. Zhu, H. Liang, and K. Xu, "Concern oriented business process modeling," in *e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on*, pp. 355–358, 2007. 42, 79
- [46] A. Shankardass, "The dynamic adaptation of an aspect oriented business process in a service oriented architecture platform," Master's thesis, Athabasca University, Athabasca, Alberta, 2009. 43, 79

- [47] A. Jabeen, S. Tariq, Q. Farooq, and Z. Malik, "A lightweight aspect modelling approach for bpmn," in *Multitopic Conference (INMIC), 2011 IEEE 14th International*, pp. 255–260, 2011. 28, 44, 50, 79
- [48] H. Witteborg, A. Charfi, D. Colomer Collell, and M. Mezini, "Weaving aspects and business processes through model transformation," in *Service-Oriented and Cloud Computing* (M. e. a. Villari, ed.), vol. 8745 of *LNCS*, pp. 47–61, Springer, 2014. 28, 44, 79
- [49] I. Patiniotakis, N. Papageorgiou, Y. Verginadis, D. Apostolou, and G. Mentzas, "An aspect oriented approach for implementing situational driven adaptation of bpmn2.0 workflows," in *Business Process Management Workshops* (M. Rosa and P. Soffer, eds.), vol. 132 of *Lecture Notes in Business Information Processing*, pp. 414–425, Springer Berlin Heidelberg, 2013. 28, 44, 79
- [50] W. Aalst, A. Hofstede, and M. Weske, "Business process management: A survey," in *Business Process Management* (W. Aalst and M. Weske, eds.), vol. 2678 of *Lecture Notes in Computer Science*, pp. 1–12, Springer Berlin Heidelberg, 2003. 28
- [51] P. Harmon, "The scope and evolution of business process management," in *Handbook on Business Process Management 1*, pp. 37–80, Springer, 2015. 33, 34, 35, 36
- [52] D. Hollingsworth, "Workflow management coalition - the workflow reference model," tech. rep., Workflow Management Coalition, Jan. 1995. 37, 38, 57, 73
- [53] N. Russell, W. Aalst, A. Hofstede, and D. Edmond, "Workflow resource patterns: Identification, representation and tool support," in *Advanced Information Systems Engineering* (O. Pastor and J. Falcão e Cunha, eds.), vol. 3520 of *Lecture Notes in Computer Science*, pp. 216–232, Springer Berlin Heidelberg, 2005. 38
- [54] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007. 38
- [55] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. van der Aalst, "Towards a taxonomy of process flexibility," in *CAiSE Forum*, vol. 344, pp. 81–84, 2008. 39, 40
- [56] T. Elrad, R. E. Filman, and A. Bader, "Aspect-oriented programming: Introduction," *Commun. ACM*, vol. 44, pp. 29–32, Oct. 2001. 41
- [57] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Business process execution language for web services," 2003. 42
- [58] G. Decker, H. Overdick, and M. Weske, "Oryx - an open modeling platform for the bpm community," in *Business Process Management* (M. Dumas, M. Reichert, and M.-C. Shan, eds.), vol. 5240 of *Lecture Notes in Computer Science*, pp. 382–385, Springer Berlin Heidelberg, 2008. 43
- [59] I. Machado, R. Bonifácio, V. Alves, L. Turnes, and G. Machado, "Managing variability in business processes: an aspect-oriented approach," in *Proceedings of the 2011 international workshop on Early aspects*, EA '11, (New York, NY, USA), pp. 25–30, ACM, 2011. 45
- [60] F. Santos, C. Cappelli, F. Santoro, J. do Prado Leite, and T. Batista, "Aspect-oriented business process modeling: analyzing open issues," *Business Process Management Journal*, vol. 18, no. 6, pp. 964–991, 2012. 45
- [61] F. Santos, J. Sampaio do Prado Leite, C. Cappelli, T. Batista, and F. Santoro, "Using goals to identify aspects in business process models," in *Proceedings of the 2011 international workshop on Early aspects*, EA '11, (New York, NY, USA), pp. 19–23, ACM, 2011. 45
- [62] F. Santos, C. Cappelli, F. Santoro, J. do Prado Leite, and T. Batista, "Analysis of heuristics to identify crosscutting concerns in business process models," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, (New York, NY, USA), pp. 1725–1726, ACM, 2012. 45

- [63] F. Santos, J. Prado Leite, C. Cappelli, F. Santoro, and T. Batista, "A proposal for ownership representation in the context of business process models," in *Enterprise, Business-Process and Information Systems Modeling* (I. Bider, T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer, and S. Wrycza, eds.), vol. 113 of *Lecture Notes in Business Information Processing*, pp. 61–75, Springer Berlin Heidelberg, 2012. 45
- [64] C. Francescomarino and P. Tonella, "Crosscutting concern documentation by visual query of business processes," in *Business Process Management Workshops* (D. Ardagna, M. Mecella, and J. Yang, eds.), vol. 17 of *Lecture Notes in Business Information Processing*, pp. 18–31, Springer Berlin Heidelberg, 2009. 45
- [65] C. Di Francescomarino and P. Tonella, "Crosscutting concern mining in business processes," *Software, IET*, vol. 5, no. 6, pp. 552–562, 2011.
- [66] C. Ghidini, C. Francescomarino, M. Rospoche, P. Tonella, and L. Serafini, "Semantics-based aspect-oriented management of exceptional flows in business processes," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 1, pp. 25–37, 2012. 45
- [67] D. Sabbah, "Aspects: From promise to reality," in *Proceedings of the 3rd International Conference on Aspect-oriented Software Development, AOSD '04*, (New York, NY, USA), pp. 1–2, ACM, 2004. 47
- [68] P. Johannesson and E. Perjons, *An introduction to design science*. Springer, 2014. 47, 49
- [69] A. Hevner and S. Chatterjee, "Introduction to design science research," in *Design Research in Information Systems*, vol. 22 of *Integrated Series in Information Systems*, Springer US, 2010. 47
- [70] A. R. Hevner, "A three cycle view of design science research," *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007. 48
- [71] V. Vaishnavi and W. Kuechler Jr, *Design science research methods and patterns: innovating information and communication technology*. CRC Press, 2007. 48, 49
- [72] R. H. von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004. 49, 50
- [73] K. Jensen, L. M. Kristensen, and L. Kristensen, *Coloured Petri nets: modelling and validation of concurrent systems*. Springer, 2009. 55
- [74] W. Aalst, "Workflow verification: Finding control-flow errors using petri-net-based techniques," in *Business Process Management* (W. Aalst, J. Desel, and A. Oberweis, eds.), vol. 1806 of *Lecture Notes in Computer Science*, pp. 161–183, Springer Berlin Heidelberg, 2000. 55
- [75] W. M. Van Der Aalst, "Woflan: a petri-net-based workflow analyzer," *Systems Analysis Modelling Simulation*, vol. 35, no. 3, pp. 345–358, 1999. 55
- [76] J. Cardoso, J. Mendling, G. Neumann, and H. Reijers, "A discourse on complexity of process models," in *Business Process Management Workshops* (J. Eder and S. Dustdar, eds.), vol. 4103 of *Lecture Notes in Computer Science*, pp. 117–128, Springer Berlin Heidelberg, 2006. 59
- [77] W. Aalst, "The application of petri nets to workflow management," *Journal of Circuits, Systems and Computers*, vol. 08, no. 01, pp. 21–66, 1998. 59
- [78] T. McCabe, "A complexity measure," *Software Engineering, IEEE Transactions on*, vol. SE-2, pp. 308–320, Dec 1976. 60
- [79] J. Cardoso, "Control-flow complexity measurement of processes and weyuker's properties," in *6th International Informatika Conference*, vol. 8, pp. 213–218, 2005. 60

- [80] I. bider, "Agile process development: Why, how and when. applying nonaka's theory of knowledge transformation to business process development," in *SIGBPS Workshop on Business Processes and Services (BPS'12)*, pp. 13–17, 2012. 61, 62, 82
- [81] W. Aalst, "Verification of workflow nets," in *Application and Theory of Petri Nets* (P. Azéma and G. Balbo, eds.), vol. 1248 of *LNCS*, pp. 407–426, Springer, 1997. 69, 70
- [82] W. Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," *Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, 2004. 69, 70
- [83] P. Tarr, H. Ossher, W. Harrison, and S. M. Sutton, Jr., "N degrees of separation: multi-dimensional separation of concerns," in *Proceedings of the 21st international conference on Software engineering, ICSE '99*, (New York, NY, USA), pp. 107–119, ACM, 1999. 78
- [84] I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization Science*, vol. 5, no. 1, pp. 14–37, 1994. 82

# Included Papers





ISBN: 978-91-7649-587-2  
ISSN 1101-8526



Stockholm  
University

Department of Computer and Systems Sciences