BIPM : from expert knowledge to metadata driven BI workflows : scaling up data analytics processes by embedding expert knowledge

Leijssen, R.M.P.

Award date:
2015

BIPM: From expert knowledge to metadata driven BI workflows

Scaling up data analytics processes by embedding expert knowledge

Master's Thesis

Rob Leijssen

Eindhoven University of Technology
Department of Mathematics and Computer Science

Supervisor:
dr. ir. B.F. van Dongen

Examination Committee:
dr. ir. B.F. van Dongen
prof. dr. ir. P.W.P.J. Grefen
ir. A.T.M. Zegers

Eindhoven, September 2015

Abstract

Enterprise Business Intelligence solutions focus on the process of extracting knowledge from data. Innovative solutions rapidly evolve over time. As a result, documentation and standardization is lagging behind. Knowledge of the data flow resides implicitly as expert knowledge with the people that developed the solution. This dependency causes a bottleneck when scaling the value proposition towards a broader use, such as global solutions. This thesis investigates how to discover the process flow and data assumptions in expert knowledge dependent BI solutions. We present a methodology to significantly reduce expert knowledge dependence. This research project was based on a review of relevant literature in the domains of Business Process Management and Workflow Management. We validated the completeness and correctness of our methodology using a representative case study at KPMG Advisory, in the domain of tax analytics. We propose a platform-independent methodology – namely Business Intelligence Process Management (BIPM) – consisting of five phases: stakeholder analyses, process flow elicitation, identification of assumptions, transforming assumptions to process activity parameters and implementation. The outcome of BIPM is a metadata driven BI workflow definition that is implementable on a workflow engine with relatively little effort.

Keywords: Business Intelligence, ETL, expert knowledge, discovery, modelling

Preface

This thesis is the result of my graduation project, which is the final chapter my study Business Information Systems at Eindhoven University of Technology. My project has been conducted in collaboration with KPMG Advisory, Eindhoven.

I would firstly like to thank my daily supervisors, dr. Boudewijn van Dongen and Alexander Zegers for their valuable advice, discussions and continued support over the last 6 months. The collaboration during my master project has been a pleasant and a great learning experience. I would also like to thank prof. dr. Paul Grefen for serving on my committee and our interesting discussions. I am grateful to Stephan Janssen of KPMG, who enabled me to pursue my passion in a global initiative: bringing innovative scientific solutions to industry, delivering impact through reusability and optimizing processes.

The design project on the Tax Intelligence Solution challenged me in following a multidisciplinary approach, combining research from the domains Business Process Management, Workflow Management and Software Architecture. My visit to the Microsoft Campus in Redmond, USA was a great experience. Working there with KPMG's Global Data & Analytics team brought me lots of inspiration and the ability to make impact with solutions on a global scale.

Next to the my learning experience on academic research and the development of high quality software in industry, I enjoyed the challenging table soccer games with colleagues from KPMG Eindhoven.

A special thanks goes to my parents, who supported me throughout my study. In addition, I would like to thank Leanne, Joep, Stefan, René and Max for valuable discussions and giving feedback to this thesis.

Rob Leijssen
Eindhoven, September 2015

Contents

List of figures

List of tables

Abbreviations

Abbreviation	Term
BI	Business Intelligence
ETL	Extract, Transform, Load
RQ	Research question
BIPM	Business Intelligence Process Management
BPM	Business Process Management
PDAF	Process Discovery Application Framework
EPC	Event-driven process chain
UML	Unified modelling language
BPMS	Business Process Management Systems
SWFMS	Scientific Workflow Management Systems
PaaS	Platform as a Service
TIS	Tax Intelligence Solution
VAT	Value Added Tax

1. Introduction

Business Intelligence (BI) has gradually evolved over time. In the 90s, reports were constructed by senior management using ad-hoc queries. This is a suitable approach for Manufacturing Resource Planning (MRP) and early Enterprise Resource Planning (ERP) systems, but was challenging when systems became bigger (*increase of volume)* and more detailed (*increase of complexity*). In the late 80s, the concept of *data warehousing* was introduced by IBM (Devlin & Murphy, 1988). This enables fast and efficient handling and processing of large amounts of data, but requires more structure in the process of getting insights from data. Instead of directly querying on operational databases, a separate analytical environment is set up. Data from operational systems is *extracted*, *transformed* and *loaded* into this analytical environment, also known as ETL processing.

Innovative data analytic propositions often have their origin in a small proof-of-concept, e.g. as a collection of *Stored Procedures*. Once the enterprise recognizes their potential value, they ask for more features and extensions. These enterprise BI solutions rapidly evolve over time, documentation and standardization is lagging behind. Knowledge of the data flow resides implicitly as *expert knowledge* with the people that developed the solution. This causes a bottleneck when the enterprise requires to scale the BI value proposition towards a broader use (e.g. as global solution).

In this thesis, Business Intelligence Process Management (BIPM) is presented. BIPM is a methodology to reduce the dependence of expert knowledge significantly, and thereby enabling innovative BI solutions to scale up to the global corporate level. Our methodology is based on two design success principles; 1) making expert knowledge explicit and 2) delivering flexibility through configurability. The result of BIPM is a metadata driven BI workflow definition that is implementable on a workflow engine with relatively little effort.

1.1 Problem definition

The problem is defined in the following way:

How to reduce dependence on expert knowledge in order to scale up BI solutions?

1.2 Research questions

From our problem description, the following research goal was determined:

Develop a methodology to reduce expert knowledge dependence in BI solutions

In order to reach this goal, we constructed the following core research questions (RQs):

RQ1. Which main stakeholders and interests can be identified in the process of reducing expert knowledge dependence in BI solutions?
RQ2. How to elicitate the process flow of an undocumented BI workflow process?
RQ3. How to identify assumptions made during data transformations and analytics?
RQ4. Which types of process parameterizations can be identified, and how are these modelled?
RQ5. How to transform data assumptions into parameterized process activities?
RQ6. How to successfully implement the metadata driven workflow model?

1.3 Scope

Our research problem is scoped by the following assumptions and scope definitions.

#	Assumption / scope definition
A1.	The process relies on the data flow architecture components of data warehousing (Wilbik & Kaymak, 2013): data sources, ETL, data loading, comprehensive database.
A2.	The process has a single begin and end state, defined as data objects.
A3.	The maturity of the as-is process is located at EBIMM level 1 (Chuah, 2010) and the organization has the ambition to move to level 2 (or 3).
A4.	There is a lack of documentation on the process, activities and the data flow.
A5.	The execution of a case involves a single or only a few direct users: case assignment heuristic (Dumas, La Rosa, Mendling, & Reijers, 2013).
A6.	An enterprise BI solutions is a value proposition to a particular data analytics process, which extracts knowledge from raw data.

Table 1: Definition of assumptions to the research context

1.4 Industrial context

In this section, we explain the industrial background of the master project.

1.4.1 Company profile

The company involved in the case study of this thesis is KPMG Advisory N.V, which is part of the KPMG network; one of the largest professional services companies in the world[1]. As of 2015, more than 160.000 professionals are working in 155 countries[2].

The organization is organized around the following three main practices;

- *Audit and Assurance*: assures reliability of information towards stakeholders.
- *Tax*: provides advisory and compliance alignment within the tax domain.
- *Advisory*: improves performance of organizations through analytics.

The Tax practice of the Dutch KPMG firms is executed by Meijburg & co, which joined the KPMG network as of 1992.

The case study takes place in the Advisory practice, within the Risk Consulting IT Advisory department at the office in Eindhoven. IT Advisory is delivering services for two types of customers;

- *Internal*: to enable IT-supported operations within *Audit and Assurance*, and *Tax* practices (e.g. data preparation or tool support).
- *External*: to provide insights and IT-assurance to external customers (e.g. data-driven operational excellence and IT quality assessment).

1.4.2 Tax Technology

Originated in 2006, KPMG IT Advisory is developing KPMG's Tax Intelligence Solution (TIS), in close collaboration with Meijburg & Co. This solution is delivered as IT platform for tax advisors to monitor risks and opportunities related to tax involved in transactions. Risks are for example transactions which are registered with a lower Value Added Tax (VAT) percentage than required (according to law regulations). Opportunities are situations where the customer could save money which otherwise would be paid as tax. A trivial example would be a transaction which is allowed to be charged for a low VAT percentage (under certain circumstances) but is registered under the regular (high) VAT percentage. Core activities of tax advisory are performed by tax experts of Meijburg & Co.

[1] http://en.wikipedia.org/wiki/KPMG

[2] http://www.kpmg.com/Global/en/about/Overview/Pages/History.aspx

1.5 Thesis structure

In Chapter 2 we present related work to the research problem and a blueprint of the developed methodology.

The subsequent seven chapters provide outcome to the research questions (section 1.2). Chapter 3 answers RQ1 by discussing which stakeholders are involved in which activities. Chapter 4 discusses related work on process flow elicitation. Chapter 5 proposes the process flow elicitation approach (RQ2). Chapter 6 guides the process of identifying data processing assumptions (RQ3). Concepts for modelling types of BI parameters are elaborated in Chapter 7 (RQ4). The transformation of assumptions to a parameterized workflow model (RQ5) is realized in Chapter 8. Finally, Chapter 9 covers the implementation requirements (RQ6).

Using a case study in an industry setting we evaluate the applicability of BIPM (Chapter 10). We conclude our thesis in Chapter 11 with a summary of conclusions and suggestions for further research.

2. The BIPM methodology

In our study we designed a methodology to transform expert knowledge dependent BI solutions into metadata driven BI products. This chapter presents our proposed methodology. In section 2.2 we give a high-level overview of steps and their interrelations. In the following sections, the motivation and key activities are discussed.

2.1 Related work

Business Intelligence (BI) is an umbrella term that combines architectures, tools, databases, applications and methodologies (Wilbik & Kaymak, 2013). A data warehouse forms the cornerstone for BI and is a subject-oriented and integrated collection of data (Chaudhuri & Dayal, 1997). An essential component of data warehousing is the Extract, Transform and Load (ETL) process (El Akkaoui & Zimanyi, 2009). Data cleaning and modifications of the data are required to store data in the data warehouse.

A few papers deal with conceptual modelling of ETL processes (El Akkaoui & Zimanyi, 2009). These papers abstract from the challenges of discovering process steps. Vassiliadis, Simitsis & Skiadopoulos (2002) developed a metamodel to define data processing activities among data artifacts. Operations are atomic and based on concepts from relational algebra. Simitsis (2005) extends this line of research with the mapping to a logical (process) design. Furthermore, El Akkaoui & Zimanyi (2009) propose a platform-independent approach to ETL design based on BPMN. The approach focusses on the conceptual modelling aspect. Modelling is focused towards elementary data processing operations, such as loading raw data into tables and filling additional columns in the dataset. Xu, Liao, Zhao & Wu (2011) propose a metadata-driven service model as basis for reusable ETL processes. This model forms the bases for an ETL service framework. The framework includes *Process Customization Services* to configure parameter metadata.

2.2 Methodology blueprint

A high-level overview of our proposed methodology BIPM is shown in Figure 1. Each fragment of the diagram is mapped to one of the six research questions (section 1.2). This is shown using a color coding. We also indicated the chapter in which each research question is answered. These chapters form the basis of our methodology design. Our methodology is based on two design success principles; 1) making expert knowledge explicit and 2) delivering flexibility through configurability.

In the following sections, we discuss the activities involved per diagram fragment. We refer to them as *BIPM activities*. The first step is *stakeholder analyses* and corresponds to RQ1. This relates to involvement of the right stakeholders and is discussed in section 2.3. Then, both *process flow elicitation* (RQ2) and *identification of assumptions* (RQ3) start. These activities are discussed in

sections 2.4 and 2.5 respectively. After both activities finished, the transformation *from assumptions to process activities* (RQ4+RQ5) takes place. Here, the identified assumptions are modelled as process parameters and integrated into the workflow model. This activity is discussed in section 2.6. The result is the *metadata driven workflow model*. Finally, the *implementation* takes place (RQ6). Implementation guidelines are depicted in section 2.7. Optionally, beta users of the prototype provide additional input for *process flow elicitation* and *identification of assumptions*.

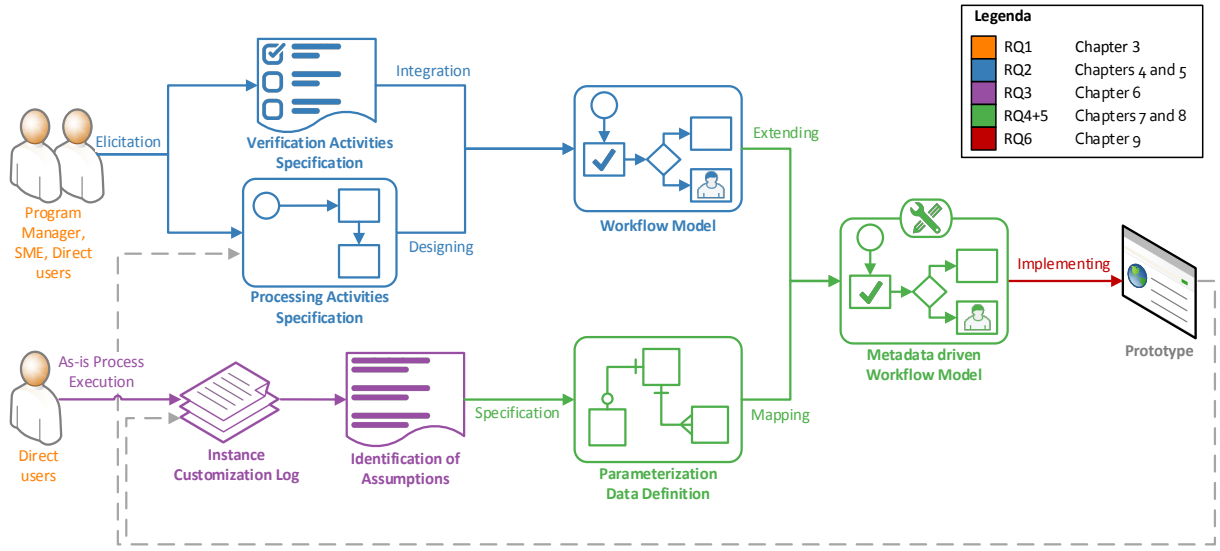

Figure 1: High-level overview of BIPM

2.3 Stakeholder analyses

The first step is to involve the right stakeholders that are required for the rest of the activities. Background on relevant stakeholders is discussed in Chapter 3. In section 3.2 we identified a selection of potentially relevant stakeholders throughout the process. For reference, their interests are elaborated in section 3.3. The assignment of required stakeholder roles to BIPM activities is shown in section 3.4. In this BIPM activity, we assign concrete names or organization units to these (generic) stakeholder names. Planning and involvement of proceeding BIPM activities is based on this role matrix.

2.4 Process flow elicitation

Essential for the design of a workflow model is to discover the process flow of the undocumented BI process. We propose to use a combination of interviewing, observation and document analysis as methods for elicitation (section 5.1.7). Input is required from the Program Manager, Subject Matter Expert and Direct users and involves the following phases:

1. Identify the process boundaries;
2. Identify the activities and events;

3. Identify resources and their handovers;
4. Identify the control flow;
5. Identify additional elements.

A fine-grained overview of required tasks and deliverables is depicted in section 5.1.5. Optionally, additional input for refinement is possible through feedback from beta users of the to-be implemented prototype. The output of this BIPM activity is a *workflow model*.

2.5 Identification of assumptions

This activity provides input which is required for creating a metadata driven extension of the process model. The input comes from *assumptions* in data processing activities. Background on identification of these assumptions is depicted in chapter 6. Involvement is required from the business analyst and direct users and consists of the following phases:

1. Introduction;
2. Collecting evidence;
3. Analysis and elaborating assumptions.

A fine-grained overview of required tasks is depicted in section 6.2.1. Optionally, additional input for refinement is possible through feedback from beta users of the to-be implemented prototype. This is depicted in section 6.2.2. The output of this BIPM activity is an overview of assumptions and their relations to the content of BI activities.

2.6 From assumptions to activity parameters

In this activity we extend the elicitated workflow model (section 2.4) with parameterization. The identified *assumptions* of data processing activities (section 2.5) are transformed into configurable activity parameters. For reference, background on types of process parameters is depicted in Chapter 7.

In Chapter 8 we discuss the two-step approach for designing and integrating activity parameters using assumptions:

1. Specification of parameterization data definition;
2. Integration into workflow model.

The data modeler and process analyst are involved in this design process. A fine-grained overview of required tasks is depicted in sections 8.3 and 8.4. The output of this BIPM activity is a *metadata driven workflow model*.

2.7 Implementation

Finally, the *metadata driven workflow model* is implemented as prototype using workflow management technology. In Chapter 9 we provide guidance in selecting appropriate workflow

technology. First, we need to determine which prototyping approach to follow and which class of workflow systems to use. This is discussed in section 9.2 and section 9.3.1 respectively. After this, a workflow engine is selected based on technical appropriateness (section 9.3.2) and contextual requirements (section 9.3.3).

Optionally, feedback from beta users of the prototype becomes new input for the BIPM activities *process flow elicitation* and *identification of assumptions*.

3. Stakeholder analyses

In this chapter we discuss the relevance and interests of product stakeholders in the context of expert knowledge dependent BI solutions (RQ1, section 1.2). For BIPM activities (e.g. *process flow elicitation*) we need to know from which stakeholder input is required (Figure 2). This chapter provides guidance using a detailed overview of stakeholder roles for each of the activities in BIPM.

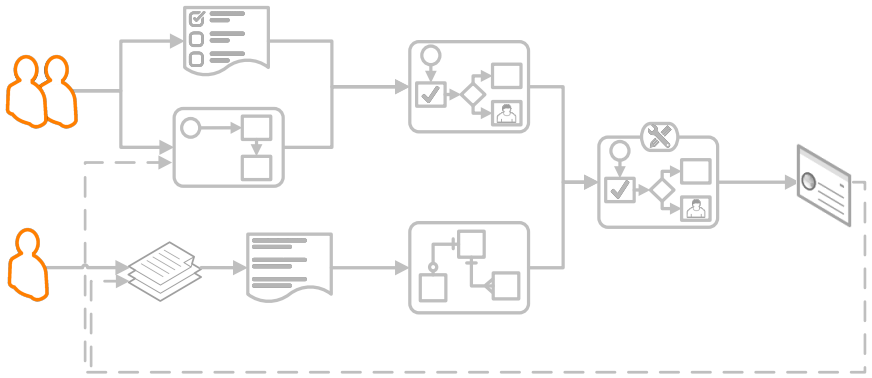

Figure 2: Stakeholder analyses in the context of BIPM

In section 3.1 we motivate the importance of stakeholder management and discuss potential stakeholders in corporate (software) projects using related work. A selection of stakeholders for expert knowledge dependent BI solutions is made in section 3.2. Their interests are discussed in section 3.3. Finally, in section 3.4 we define the stakeholder roles for *process flow elicitation* (chapter 5), *identification of assumptions* (chapter 6) and *transforming assumptions into activity parameters* (chapter 8).

3.1 Related work

3.1.1 Importance of stakeholder management

Each software project requires users that interact with the system and these users work in a certain context (e.g. IT specialists or domain experts). Besides users, other people like product owners and support staff are also involved in the project. If user classes are not identified early, some user needs will not be satisfied due to missing requirements (Wiegers & Beatty, 2013). Persons, groups or organizations that are involved in the project, process or outcome are defined as *stakeholders* (Wiegers & Beatty, 2013). Within the domain of requirements engineering, stakeholder analysis is considered crucial for project success; to prevent rework of requirements specification (IIBA, 2009) and to increase the chance of adaption (Wiegers & Beatty, 2013).

3.1.2 Potential stakeholders

Wiegers et al. (2013) identified a spectrum of potential stakeholders in Figure 3. Three organizational levels are taken into account: outside the development organization, the development organization and the project team. Note that often only a subset of stakeholders apply to a particular project.

Direct users will interact with the system hands-on. Indirect users may dependent on the output of the system, but do not interact with the system itself. Beta users are the direct users that are involved in tests of early implementations, such as prototypes.

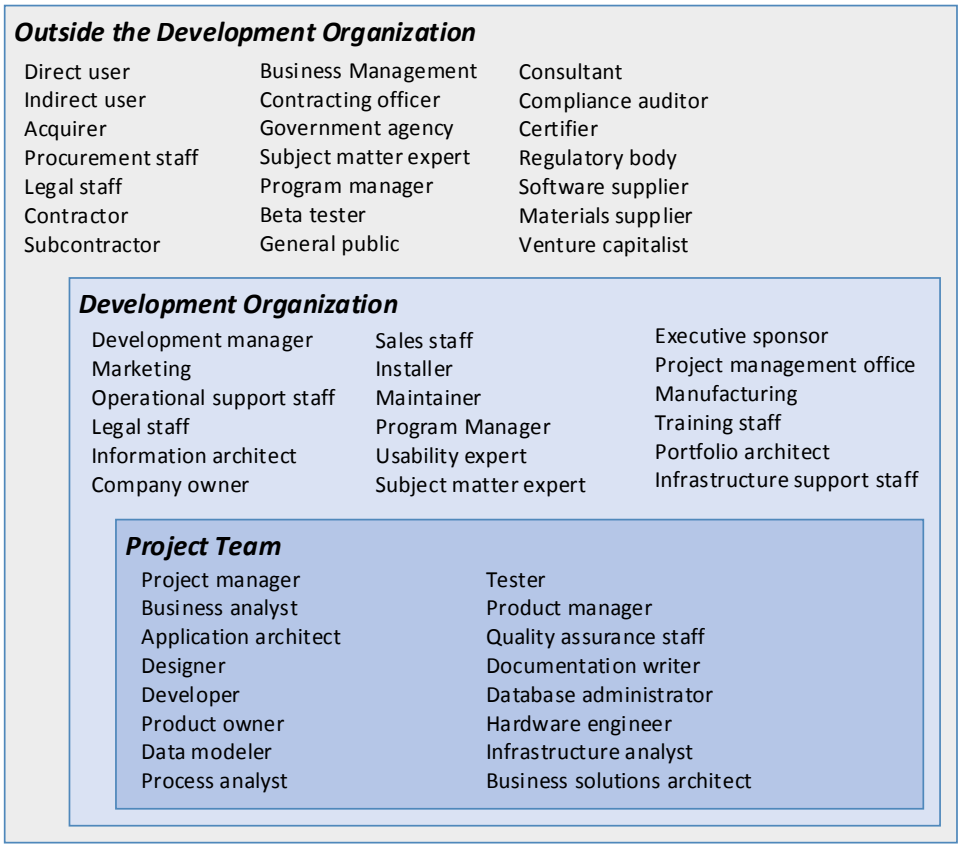

Figure 3: Potential stakeholders spectrum (Wiegers & Beatty, 2013)

3.2 Identification

In BIPM, we use the proposed potential stakeholder spectrum of Wiegers et al. (2013) to select a subset of stakeholder candidates that are of high relevance in transformation to workflow-supported BI products. The outcome of this projection is shown in Figure 4.

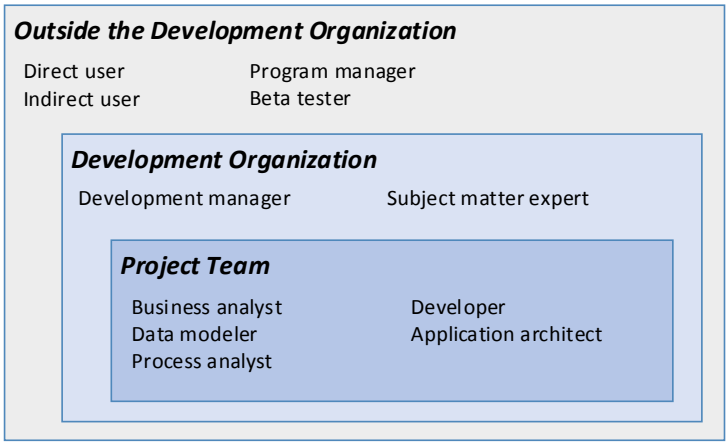

Figure 4: Selection of relevant stakeholder roles

3.3 Interests

Each stakeholder has its own perspective on an enterprise project and therefore interests differ among stakeholders. Additionally, each stakeholder has different responsibilities according to the nature of the stakeholder's role. Smith (2000) proposes a matrix for elaborating stakeholders' interests, impact and priority. We abstract from the impact and priority dimensions, since they are assumed to be project specific. Assumptions regarding the interests in the project are described in Table 2.

Context	Stakeholder	Interests
Outside Development Organization	**Direct user**	Ease of execution (usability) Robustness Minimizing of SME dependence
	Indirect user	Efficient delivery Data analytics tailored towards case specific needs
	Program manager	Efficient delivery Scalability of indirect user base
	Beta tester	Ease of execution (usability), robustness Minimizing of SME dependence
Development Organization	**Development manager**	Scalability of direct user base
	Subject matter expert	Promote to 2nd line SME Deploy SME knowledge efficiently
Project Team	**Business analyst**	Supporting business driven using technology
	Data modeler	Documenting and standardizing data flow
	Process analyst	Documenting and standardizing process flow
	Developer	Supporting business processes using technology
	Application architect	Preserving consistency and quality among technology landscape

Table 2: Stakeholder interests for reducing expert knowledge dependent BI solutions

3.4 Relevance

In Chapters 5, 6 and 8 we design BIPM activities that an organization needs to fulfill in order to reduce expert knowledge dependence in BI solutions: process flow elicitation, identification of assumptions and transforming assumptions into process activity parameters. These activities require inputs from stakeholders and require specific skillsets from people inside the project team. We defined the activity roles using the RACI matrix definition (IIBA, 2009):

- [R]esponsible – does the work;
- [A]ccountable – is the decision maker;
- [C]onsulted – must be consulted prior to the work and gives input;
- [I]nformed – means that they must be notified of the outcome.

An overview of required stakeholder and their role on the corresponding BIPM activities is shown in Table 3. This schema provides input for BIPM project organization, we propose to assign concrete names or organization units to these (generic) stakeholder names. This enables to plan involvement of the right people for the rest of the BIPM activities.

Context	Stakeholder	Process flow elicitation	Identification of assumptions	From assumptions to activity parameters	Implementation
Outside Development Organization	**Direct user**	C	R		
	Indirect user	I			
	Program manager	A, C			
	Beta tester				C
Development Organization	**Development manager**	C		I	C
	Subject matter expert	C	C		
Project Team	**Business analyst**	R	R		
	Data modeler	R		R	
	Process analyst	R		R	
	Developer	R		I	R
	Application architect			I	R

Table 3: Assignment of required stakeholder roles to BIPM activities

Description on the roles within *process flow elicitation* are provided in section 5.1.5. For *identification of assumptions*, these are elaborated in section 6.2.1. Role details on transforming *from assumptions to process activities* and *implementation* are depicted in sections 8.2 and 9.3.3 respectively.

4. Process flow elicitation in BPM

Elicitating a process model is required to functionally design an executable BI workflow (RQ2, section 1.2). In this chapter we discuss best practices of as-is process discovering in the Business Process Management (BPM) domain. BPM is a crossroad of multiple viewpoints (Dumas et al., 2013). For business managers, BPM demonstrated the ability to improve performance and quality of operational processes. From the IT-perspective, BPM provides a shared language to communicate with the business (Dumas et al., 2013). Total Quality Management and Operations Management are other well-known disciplines which investigate operational processes. These two disciplines are primary found in the manufacturing domain (Dumas et al., 2013). BPM fits closer to our research goal since it is oriented to service processes.

In sections 4.1 we indicate the context within the BPM lifecycle. Section 4.2 elaborates best practices in process discovery. Chapter 5 maps these findings to the field of expert knowledge dependent BI solutions.

4.1 Relation to the BPM lifecycle

In most BPM initiatives, multiple processes are considered. This requires the construction of a process map using *process identification*. Within the context of our research problem, we focus solely on one BI process definition. *Process discovery* is the activity to gather and model information of an as-is process model (Dumas et al., 2013). Process discovery results in a model based representation of the existing process instance. Once the process is modelled, *process analysis* is enabled (Figure 5).

Process discovery consists of the following four phases (Dumas et al., 2013):

1. Defining the setting: organization of the process discovery and modelling act;
2. Gathering information: extracting, collecting and conflict resolving to understand the process;
3. Conducting the modeling task: construction of a model that represents the as-is process;
4. Assuring process model quality: evaluating the quality of the model on semantic, syntactic and pragmatic aspects.

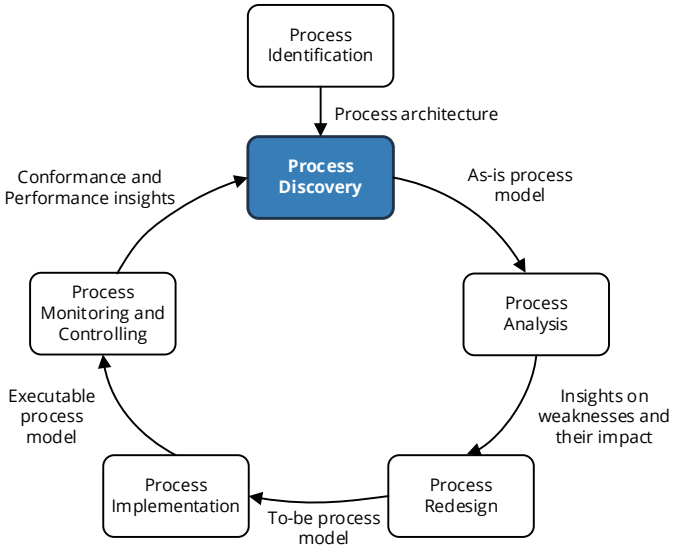

Figure 5: Context within the BPM lifecycle (Dumas et al., 2013)

4.2 Product discovery

We distinguish two main development disciplines from the four phases of Process Discovery (4.1), namely *modeling* and *discovery*. Dumas et al. (2013) recognizes these disciplines by showing the role differences between the process analyst and domain expert (Table 4).

Aspect	Process analyst	Domain expert
Modeling Skills	Strong	Limited
Process Knowledge	Limited	Strong

Table 4: Typical profile of a process analyst and domain expert (Dumas et al., 2013)

4.2.1 *Product Discovery Application Framework*

Dumas et al. (2013) elaborates the components and methods involved within process discovery. We constructed a framework that incorporates these elements. This framework is represented as a diagram in Figure 6.

The leafs of the graph represent choices, solutions or facets of their corresponding parent aspect. For example, quality assurance is measurable on the syntactic, semantic and pragmatic quality dimensions. In the following sections we depict the best practices of each element, as reported in literature (Dumas et. al, 2013).

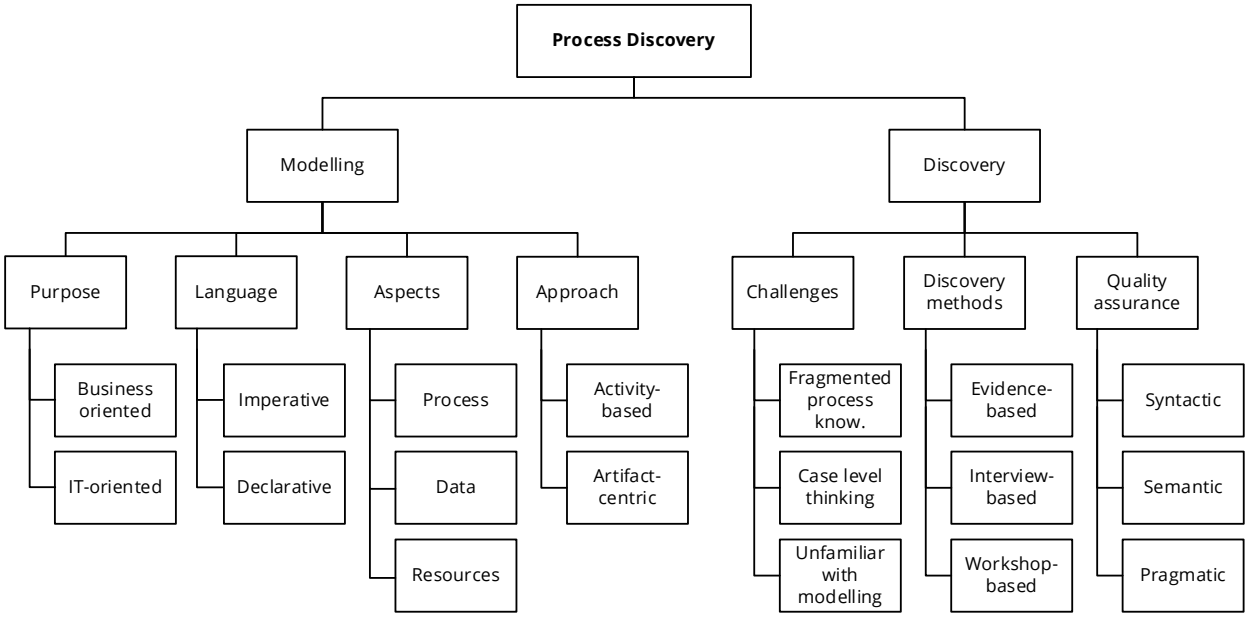

Figure 6: Process Discovery Application Framework (PDAF)

4.2.2 *Modelling – Purpose*

According to Stachowiak's definition of a model (Stachowiak, 1973), a model fulfills three characteristic properties:

- *Mapping* criterion; a model is based on some original.
- *Reduction* criterion; a model abstracts from certain details that are irrelevant.
- *Pragmatic* criterion; a model is usable for a particular purpose.

In process modelling, the model is mapped to a particular real-world process and abstracts from aspects like physical environment or execution details. Process models are classified into two main purposes (Dumas et al., 2013): organizational design (*business oriented*) and application system design (*IT-oriented*). The level of abstraction depends on the purpose; an organizational design is mainly used for understanding and communication (Dumas et al., 2013), where an application system design describes details on the execution and automation of the process.

4.2.3 *Modelling – Language*

Syntax, semantics and notation are the main elements of a modelling language (Dumas et al., 2013). The rules of the process modelling language are specified by the syntax. Semantics determine how an instance should be interpreted by specifying the meaning of the model elements. In addition, a modelling language often specifies the graphical representation of these modelling elements. In process modelling, this may help to understand ordinal relations among activities.

In the context of process modelling, the language either explicitly describes which behavior is enabled (closed world, imperative), or describes which behavior is disabled (open world,

declarative). Pichler et al. (2012) observe that imperative process models are more comprehensible than then the declarative variants, often due to the dominance imperative process modelling training in practice.

Well-established Business Process Modelling Languages (BPMLs) have their roots from the system engineering, software engineering or the process engineering domain (List & Korherr, 2006). This corresponds to the level of abstraction of the language constructs; classical Petri Nets (system engineering) are timeless and have no hierarchy, but BPMN activities (process engineering) involve the time aspect and allow hierarchy using sub processes. List et al. (2006) compared the expressiveness of seven BPMLs which have future potential or receive popularity in practice. The evaluation is based on five modelling perspective. We summarized the characteristics and evaluations in Table 5.

BPML	Purpose	Source Domain	Business Process Context Perspective	Functional Perspective	Informational Perspective	Organization Perspective	Behavioral Perspective
AD	Description, Enactment	Software Engineering	-/+	-/+	-/+	+/+	-/+
BPDM	Enactment	Process Engineering	-/-	-/+	-/+	+/+	-/+
BPMN	Description, Enactment	Process Engineering	-/-	-/+	+/+	+/+	+/+
EPC	Description, Analysis	Software Engineering	-/+	+/+	+/+	-/-	+/+
IDEF3	Description	Software Engineering	-/-	+/+	-/-	-/-	+/+
Petri Nets	Enactment	System Engineering	-/-	-/-	-/-	-/-	+/+
RAD	Description	Software Engineering	-/+	+/+	+/+	+/+	-/+

Table 5: Summary of BPML modelling support (List et al., 2006)

4.2.4 *Modelling – Aspects*

Process Modelling is a complex task. Dumas et. al (2013) propose a systematic aspect-based approach; the resource perspective is separated from the identification of the activities and the control flow. Jablonski & Bussler (1996) identify the process (control-flow), data and resource perspectives on (workflow) process modelling. Modelling the process aspect incorporates 1) the identification of activities and 2) the control flow which connects these activities. The data perspective models the context where cases operate in (i.e. process and task variables). Business processes are executed by actors (people or systems). The resource perspective focusses on the specification of these actors on the activity level.

4.2.5 *Modelling – Approach*

A common approach is *activity-based modelling*, were the focus is on the activities and the control-flow relations in the process. This approach is used by well-established industry modelling

standards such as UML Activity Diagrams and Event Driven Process Chains (EPC) (List & Korherr, 2006). Dumas et. al (2013) propose a systematic concretization of this approach ("Process Modeling method"), involving five identification phases:

1. Identify the process boundaries
2. Identify activities and events
3. Identify resources and their handovers
4. Identify the control flow[3]
5. Identify additional elements

In principle, each phase relates to 1 or more of the process modelling aspects: process, data and resources. Phase 1 considers all three aspects, phase 2 and phase 4 relate to the process. Phase 3 relates to the resource aspect and phase 5 relates to both the data (data artifacts and their relations to activities) and the process (exception handling) aspect.

An alternative to the *activity-based* type is *artifact-centric modeling*. Instead of focusing on activities, here the modeler identifies objects (artifacts) which are created or modified throughout the process. Artifact-centric modeling puts emphasis on the lifecycle of an object, based on the possible states a particular object has (e.g. created, approved, archived). This type of approach is useful for variable process execution environment, such as processes with unplanned activities (Redding, Dumas, Hofstede, & Iordachescu, 2010) and variability between the execution context such as different business units or types of customers (Dumas et al., 2013).

4.2.6 *Discovery – Challenges*

Dumas et. al (2013) identify three main challenges during process discovery;

Fragmented process knowledge: business processes often involve multiple participants which are responsible for parts of the process. In order to retrieve detailed knowledge about the process, discovery sessions with multiple domain experts need to be organized. This requires the process analyst to resolve inconsistencies due to diverging assumptions among stakeholders. Therefore, multiple process discovery cycles are required (Dumas et al., 2013).

Thinking of processes on a case level: domain experts that work on the execution of cases often describe activities according to a particular case. This requires a process analyst to ask what-if questions and be able to generalize use cases.

Unfamiliarity with business process modeling languages: domain experts are often not trained in modelling processes. Therefore, understanding the control flow part of a process model is difficult. In order to validate whether the process model aligns with reality, the process analyst needs to explain the behavior in natural language.

[3] The control flow identification phase is focused towards modelling the normal flow of the (business) process.

4.2.7 *Discovery – Discovery methods*

Dumas et. al (2013) identify three classes of techniques to retrieve input for process modelling. This relates to *requirements elicitation* for processes in the field of software requirements management.

Evidence-based discovery: contains three methods: document analysis, observation and automatic process discovery. *Document analysis* relies on existing documentation material of the system or business process under investigation. Documentation may be a "cheap" source for discovery, however material may not be clearly organized in a process-oriented way, is of a different granularity or is outdated (Dumas et al., 2013). Secondly with *observation,* the process analyst follows particular cases. Either in the active customer role (triggering a case in the system) or as passive observer (following the entire process of a real-life case). The active role emphasis the interaction with the external environment on a sub process were the passive role focusses on the entire process and how people are working on it. Thirdly, *automated process discovery* relates to the discovery field of process mining; using event logs to obtain process insights from many (historical) process instances.

Interview-based discovery: methods for retrieving insights using interviews with domain experts. The process analyst needs to generalize input from domain experts and ensure completeness of the control-flow. This relates to the discovery challenge *thinking of processes on a case level*: domain experts provide feedback often from the perspective a particular case instance. Interviews provide rich details on the process and the resource perspective, however are labor-intensive since one iteration is often not sufficient.

Workshop-based discovery: organized team meeting, sometimes known as "brown paper sessions". In contrast with interviews, this discovery method class is more interactive and involves activity modeling (e.g. using sticky notes). This enables rapid feedback to resolve *fragmented process knowledge* issues, but requires an active coordinative role of the facilitator.

4.2.8 *Discovery – Quality assurance*

Quality of a process model is assessed on three aspects:

Syntactic quality and verification: the degree in which the model conforms to the syntax of the modeling language, such as BPMN, and the structural and behavioral correctness.

Semantic quality and validation: the degree in which the model conforms to the real-world; validity and completeness.

Pragmatic quality and certification: the degree of understandability and maintainability of the model.

5. Process flow elicitation in BIPM

In this chapter we discuss how to discover an as-is process model from an undocumented BI process (RQ2, section 1.2). Discovering a process model is required to functionally design an executable BI workflow. Several stakeholders are involved in this effort, both in- and outside development organization (section 3.4). The corresponding BIPM fragment is highlighted in Figure 7. In chapter 8 we use the discovered process model to make it metadata driven. In chapter 4 we perform a literature survey to retrieve best practices in process discovery. In this chapter we map these practices to our research problem. The outcome is a guided step-based approach for elicitation and modelling the process flow.

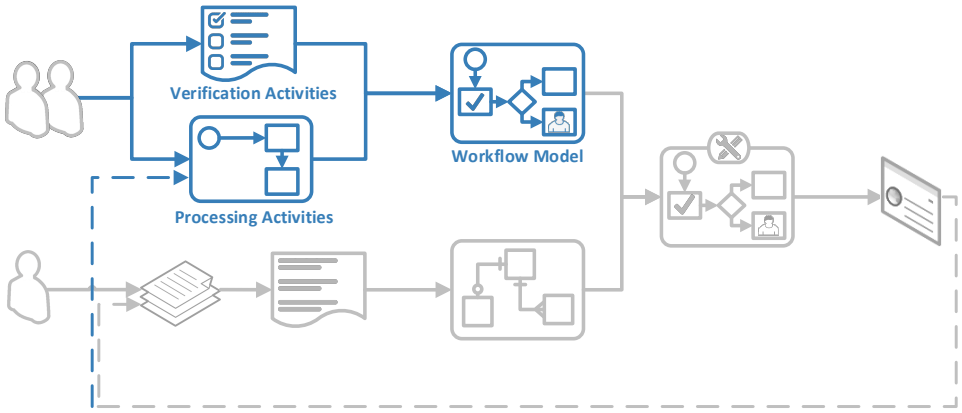

Figure 7: Process flow elicitation in the context of BIPM

5.1 Application of PDAF framework to BIPM

We project the Process Discovery Application Framework (section 4.2.1) to the field of expert knowledge dependent BI solutions. From assumptions on the research problem (section 1.3) we derive design choices. The design choices form application guidelines to elicitate a representative process model.

5.1.1 *Overview*

The projection on PDAF is shown in Figure 8. Note that the classification *context dependent* indicates that the choice/scope dependents on the industry use case where BIPM is applied to.

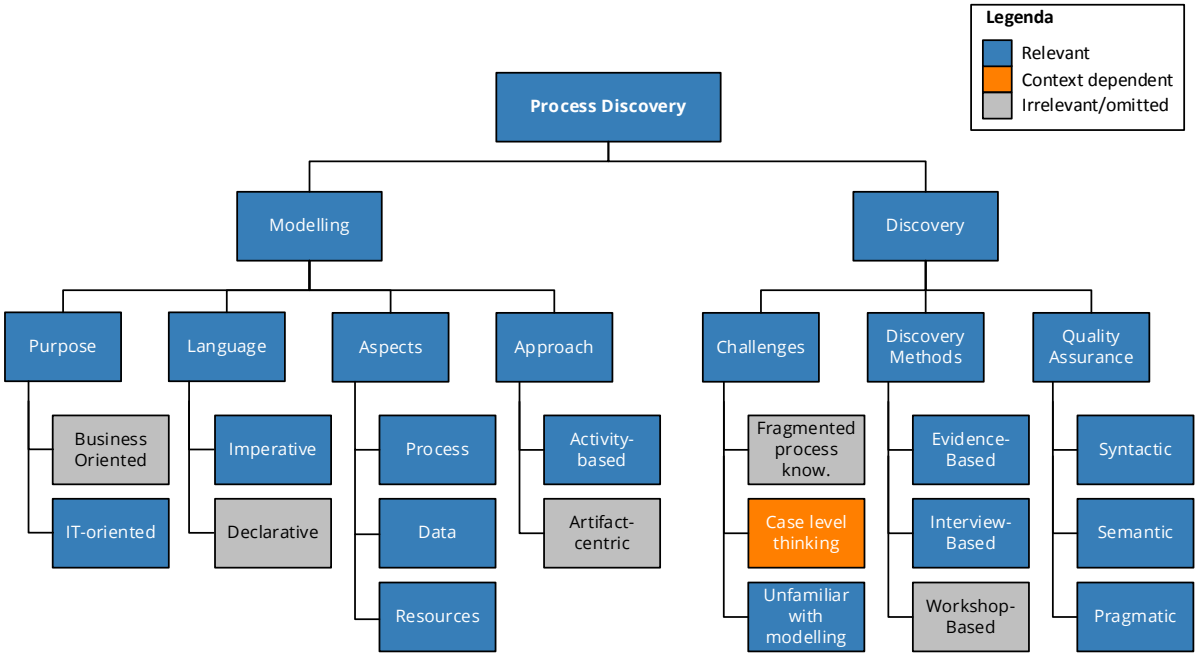

Figure 8: Application of PDAF to the field of expert-knowledge dependent BI solutions

5.1.2 *Modelling – Purpose*

The process under consideration is clearly technology-driven; data is transformed and analyzed using IT applications. Once the process is modelled, it will be used for automation. This implies we need to model on the granularity level of software specifications (Dumas et al., 2013).

5.1.3 *Modelling – Language*

The to-be modelled process has two purposes, description (as input for further refinement) and input for enactment (workflow coordination). The language should be able to model the process, data and resource aspects. List et al. (2006) concludes that Activity Diagrams and BPMN fulfil both modelling purposes. BPMN supports the data aspect (information perspective) slightly better, therefore we choose to use BPMN as modelling language.

5.1.4 *Modelling – Aspects*

For eliciting the control flow, we take the process aspect, resource and data aspect into account. The data aspect provides additional information regarding the data flow of the BI process.

5.1.5 *Modelling – Approach*

Based on the stages of the Process Modeling Method as proposed by Dumas et. al (2013), we defined a fine-grained approach to model expert dependent BI solutions (Table 6). For each phase, we defined which information needs to be collected and which of the stakeholders (Chapter 3) is required to consult. Furthermore, based on the high-level stakeholder role matrix of section 3.4 we indicate which stakeholder is responsible and therefore takes ownership of the task.

Process elicitation phase	Deliverables	Input	Owner
1. Identify the process boundaries ▪ Define the BI value propositions in scope and their business value; ▪ Define the data structure and location of the raw data sources that are fed as input for start the process; ▪ Define the data structure and location of the analytical objects produced as outcome.	▪ Goal of the BI value proposition; ▪ Representative (example) raw data set; ▪ To-be data mart definition;	Program manager, Development Manager	Business analyst
2. Identify the activities and events ▪ Identify and specify ETL / data processing activities and an initial sequential ordering; ▪ Identify and specify activities for verification of data correctness and completeness;	▪ Ordered processing activities specification; ▪ Functional specification of verification activities.	Direct user, Subject matter expert	Process analyst
3. Identify resources and their handovers ▪ Identify which (verification) activities can be automated or require expert judgements; ▪ Identify which human resource classes and system resource classes are involved and how activities are mapped to these resources;	▪ Activity-resource mapping.	Direct user, Subject matter expert	Business analyst
4. Identify the control flow ▪ Identify object-dependencies between ETL activities; ▪ Review sequential ordering of activities and design control flow; o Introduce data objects for key BI data artifacts; o Introduce choices for decision points; o Introduce parallelism to independent ETL activities; ▪ Determine point-in-time for execution of verification activities (knock-out heuristic; (Dumas et al., 2013))	▪ BI data flow dependency diagram. ▪ Control flow design.	Subject matter expert	Data modeler, Process analyst
5. Identify additional elements ▪ Define exception-flow for malformed raw data and negative verification results.	▪ Extended control flow design: *workflow model*.	Subject matter expert	Developer

Table 6: Process elicitation approach

Compared to the generic Process Modeling Method of Dumas et. al (2013), one can notice the following key differences:

- Identification of (intermediate) data objects is embedded in the *process boundary identification* and *control flow identification* phases, since the order of activities in the BI process is dependent on the data flow (assumption A1).
- The *resource identification* phase involves determining which activities can be executed autonomous and which are manual steps, since the process is IT-oriented (section 5.1.2).

5.1.6 *Discovery – Challenges*

Fragmented process knowledge is less relevant, since we assume that only one or a few direct users (data scientists) are involved in the executing of a particular case (A5; *case assignment heuristic*, (Dumas et al., 2013)).

Thinking of processes on a case level depends on the variability of the control flow and the required level of process flexibility and therefore is application context dependent. When the BI data flow follows a step based approach (without choices), the control flow is already sufficient generalized based on input from only one or a few cases.

Unfamiliarity with business process modelling languages is considered relevant in our problem context. Since no documentation is available of the process (assumption A4), direct users are possibly trained using a hands-on approach, instead of a structured model based approach.

5.1.7 *Discovery – Methods*

For each phase of our approach (section 5.1.5), we propose an organizational setup that incorporates one or more discovery methods (Table 7). Process boundary identification is organized using a kick-off session to determine the scope of the process, followed by modelling the start and final state definitions (*evidence-based*). The rest of the phases involves *interview-based* and *evidence-based* methods to retrieve requirements. As discussed in section 5.1.6, the *fragmented process knowledge* challenge is less relevant for the problem context, therefore we propose interview-based instead of workshop-based.

Process elicitation phase	Discovery methods	Application
1. Identify the process boundaries	Interview-based	Meeting with program manager and development manager to determine scope and example raw data set. Program manager informs indirect users on the scope.
	Evidence-based	Examining structure of example raw data and outcome data marts (*Document Analysis*).
2. Identify the activities and events	Evidence-based	Meetings with direct users for end-to-end observation on representative contemporary cases (*Passive Observation*). *Automated Process Discovery* based on historical process case logs (when available).
	Interview-based	Meeting with subject matter expert for functional specification of verification activities.
3. Identify resources and their handovers	Interview-based	Meeting with subject matter expert to identify resource classes and map activities.
4. Identify the control flow	Evidence-based	Examining dependency graph of produced data objects (*Document Analysis*).
	Interview-based	Meeting with subject matter expert to refine control flow design.
5. Identify additional elements	Interview-based	Meeting with subject matter expert to refine exception-flow design.

Table 7: Application of discovery methods in modelling approach

5.1.8 *Discovery – Quality assurance*

As discussed in section 5.1.3, we choose BPMN as modelling language. The process analyst should be trained in process modelling. The process analyst is responsible for verification of the *syntactic quality* directly after the control flow identification and additional elements identification phases.

The subject matter expert is responsible for input and validation whether the model conforms to the real world (*semantic quality*). This is part to all phases after process boundary identification.

The *pragmatic quality* is the precondition for enabling validation by the subject matter expert; the process analyst should use the business language while modelling and explaining the process.

6. Identification of assumptions

In chapter 5 we discussed how to elicitate a process model from an as-is BI process on the granularity level of BI activities (e.g. data transformations and analytics). In this section we dive into the content level of the BI activities. Due to evolutionary BI activity development, the process may have certain *assumptions* regarding the format, context and semantics of the raw data. Assumptions could limit the application scope of the process or lead to incorrect results. In this chapter we discuss how to identify these assumptions (RQ3, section 1.2). The outcome is a documented overview of assumptions and their relations to the content of BI activities. This is visualized in Figure 9. In chapter 8 this information is used to remove assumptions by transforming them into activity parameters.

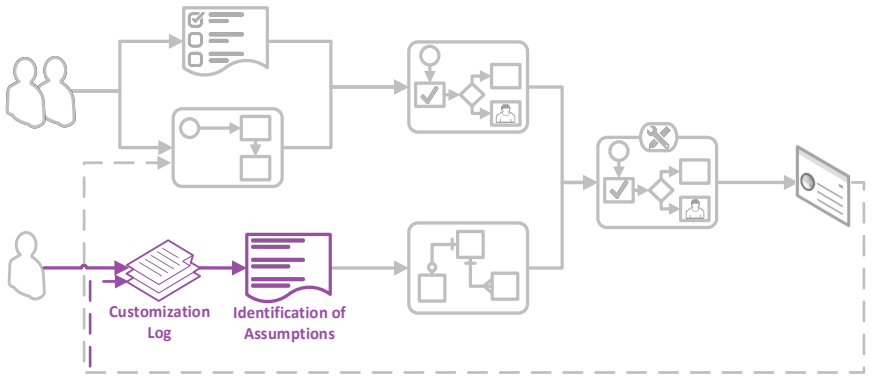

Figure 9: Identification of assumptions in the context of BIPM

In section 6.1 we discuss types of data variables using related work and give examples of assumptions according to these variable types. Then, in section 6.2 we propose a method to systematically identify and refine assumptions. Finally, in section 6.3 we design a metadata model to structure the data input that is required to analyze the assumptions.

6.1 Related work

BI processes load raw data, and then transform and possibly interpret the structure and their values. Data variables are classified into four data type categories: nominal, ordinal, discrete and continuous (Giudici, 2003).

Continuous variables are quantitative and come from measures. An example is the value of an invoice. Suppose that a data analyst is interested in exceptional invoices, and therefore automatically classifies invoices above a certain threshold (e.g. > $ 100.000) as exceptional. This criterion is potentially based on an assumption on the context of the raw data. For a small retailer this threshold is suitable, but for a large bank such large transactions are part of daily business.

Nominal variables are qualitative and are often used for categories, e.g. the credit rate of a company (Giudici, 2003). Data analysis in the risk domain could use the credit rate category

together with the sum of payments to be received for classifying the financial risk of a particular client. Analytics could assume the use of certain credit rate scheme type. However, each legal entity may use different schemes, varying from binary to very fine grained ratings (e.g. Moody's).

6.2 Evidence-based discovery approach

In our research problem, we consider BI solutions which have no or very little documentation available on the process or activity level. Together with direct users and subject matter expert it's possible to outline the process footprint (section 5.1). Dumas et al. (2013) depict *thinking of processes on a case level* as one of the challenges during *process discovery*. People are used to talking about processes according to a particular case or don't mention all exceptions and deviations. Our hypothesis is that assumptions are reflected to runtime activity customizations for individual cases. Ensuring completeness in identification of assumptions is therefore a challenging task. As a result, we propose to use evidence-based discovery to evolutionary identify ad-hoc customizations. These customizations are considered the result of hard-coded data assumptions.

6.2.1 *Documenting customizations on as-is process instances*

Contemporary instances of the BI process are executed by users in an ad-hoc fashion; in some cases, customizations are made to data transformations and analytics by editing queries or analytical procedures. We propose to train direct users to actively log the changes made, such that a business analyst uses this evidence as input for analysis. The blueprint for such a changelog is defined in section 6.3. A subset of the customizations is the result of hard-coded assumptions. The rest is not related to generalization of the process itself, but to case level flexibility (deviations that are instance-specific). This discovery process follows a three-phase approach, which is elaborated in Table 8. Note that it is possible to execute phase 2 and 3 iteratively to evolutionary discover more data assumptions.

Phase	Activities	Responsible
1. Introduction	▪ Introduce direct users to the goal of assumption identification. ▪ Implement a schema (spreadsheet or tool) based on the customization logging metamodel (section 6.3). ▪ Explain direct users in using this schema in contemporary process instances.	Business analyst
2. Collecting evidence	▪ Document customizations made during contemporary process instances.	Direct user
3. Analysis and elaborating assumptions	▪ Analyze and discuss the customization log with the subject matter expert to clarify the functional background of the change. ▪ Concretize the assumption and the data processing fragment that caused the need for change.	Business analyst

Table 8: Primary approach to identify runtime customizations caused by data assumptions

6.2.2 *Identifying additional customizations using event logs*

Optionally, it is possible to improve completeness of our identification effort supported by automated logging techniques. Once the process is implemented as an executable workflow, it is possible to perform phase 2 (section 6.2.1) by the workflow engine. The workflow engine facilitates the process execution and determines the changes made to the content of the BI activities. This happens directly after task completing. For interoperability purposes, our customization logging metamodel (section 6.3) is applicable as extension of the XES standard[4] for event logs. A model transformation to a XES extension is presented in Appendix A.

6.3 Metamodel for logging customizations

The metamodel of Figure 10 facilitates runtime customizations on data transformations and analytics. The *Customization Type* property specifies whether the need is triggered due to configurability (*Configuration)* or whether the change type is non-generalizable (instance-specific, *Exception*). *Changeset* consists of a reference to the data processing fragment and the adapted version.

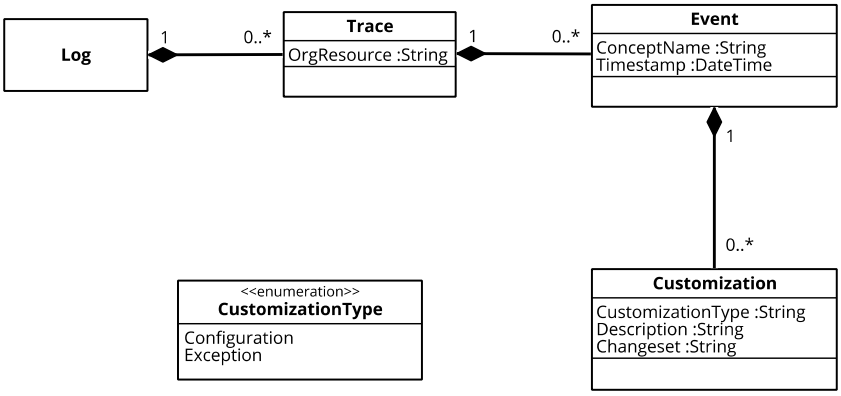

Figure 10: Metamodel for logging (query) customizations within activities

[4] http://www.xes-standard.org/

7. Identification of process parameter types

In this chapter we discuss metadata concepts to capture parameterization of data on process activities (RQ4, section 1.2). BI activities that perform data transformation or interpretations on input data sources require customizability on the level of a particular case. In our research problem, we enable the user to refine the process instance by parameterizing activities using metadata (chapter 8). This chapter covers the conceptual definition of a metamodel which is flexible enough to capture frequent parameterization types (Figure 11).

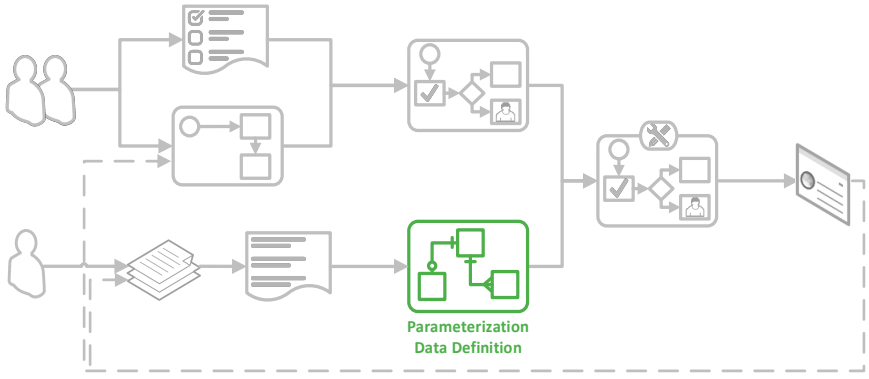

Figure 11: Identification of process parameter types in the context of BIPM

In section 7.1, we discuss concepts of flexibility in workflow design and how BPMN facilitates the handling of data using related work. Then, in section 7.2 we identify the scope of parameterization types we support. After briefly discuss existing modelling support (section 7.3) we formalize our parameterization approach into a data definition metamodel (section 7.4).

7.1 Related work

7.1.1 *Flexibility in workflow design*

Schonenberg et al. (2008) identify four ways of flexibility in workflow design: flexibility by design, deviation, underspecification and change. Research on configurable process models focusses on flexibility by design and underspecification of the process aspect. Flexibility by design relates to the ability to configure a fairly generalized process flow (van der Aalst, Dreiling, Gottschalk, Rosemann, & Jansen-Vullers, 2006). An example of flexibility by underspecification is filling in placeholder model fragments (Ramezani, Fahland, & van der Aalst, 2014).

7.1.2 *Data modelling in BPMN*

Activities represent operations on data (e.g. transformations and analytical queries). BPMN facilitates two notions of handling data; *data objects* and *properties* (Ter Hofstede, Van der Aalst, Adams, & Russell, 2010). Data objects often represent (electronic) documents and provide

additional information about the process, but according to the BPMN standard, these objects do not directly affect the execution flow. Properties are defined on the process, activity or document object level (Ter Hofstede et al., 2010). They consist of a name, type descriptor and possible subproperties, however the BPMN standard does not provide a taxonomy for data types, leaving freedom to the modeler (Ter Hofstede et al., 2010). Modelling languages for execution environments (such as BPEL) resolve these data modelling ambiguities, however these focus towards execution instead of conceptual modelling.

7.2 Types of process parameters

The simplest notion of feeding a process instance with (data-based) customizations is using a *single value*, e.g. the preferred currency or whether the source data contains a header column. In some cases, a single value is insufficient, and a *list of values* is used, e.g. a list of preferred supplier IDs. When the list of suppliers is country specific, we are talking about a *mapping* from a country to a list of suppliers. Classification of data throughout the process could lead to false positives or true negatives. In such cases, the user needs to refine the predefined conditions of the analytics. In this case, we use the notion of a *proposition*.

7.3 Modelling process parameters types

In our solution to the research problem, we use a metadata driven approach to parameterize process activities. That is, we propose extensions of a process model on the data aspect to feed metadata to the process activities. This relates to *flexibility by design* on the data aspect; the data model itself does not change once the process model is implemented.

In chapter 4 we proposed BPMN as modelling language for modelling the as-is BI process. Dumas et al. (2013) discussed that BPMN leaves freedom to the modeler to model data structures. To overcome the lack of specification, we have chosen to use UML Class Diagrams as specification language.

7.4 Conceptual metamodel for process parameters

Our initial set of data requirements (section 7.1) is used as input to construct a data definition metamodel (Figure 12). This metamodel provides a high-level taxonomy of datatypes and enables the modeler to define process activity parameters on the BI process design level.

- *Single values*: parameters with this type are represented as *ScalarValue* type. Based on the value type (bool, integer, decimal, double, timestamp and string), the modeler initializes one of the six specialization classes to capture a particular parameter.
- *List of values*: parameters containing a list (or set) of scalar values is modelled by initializing an instance of *ValueList*. Note that a type (V) is required as argument. This type represents the datatype of the list members.

- *Mapping*: represented as *ValueMapping* instance. Maps a single value of type X to the domain of value objects of type Y, i.e. $f : X \rightarrow Y$. The mapping output is *NULL* for inputs which have no corresponding *Pair* instance specified.
- *Proposition:* represented as *Proposition* instance. A *Proposition* is modelled as a recursive tree of *PropositionComponents*. The leafs of the tree are instances of *Conditions*; they form the atomic parts of a proposition, consisting of a *ReferenceObject* (the subject), comparison operator and *ReferenceValue*.

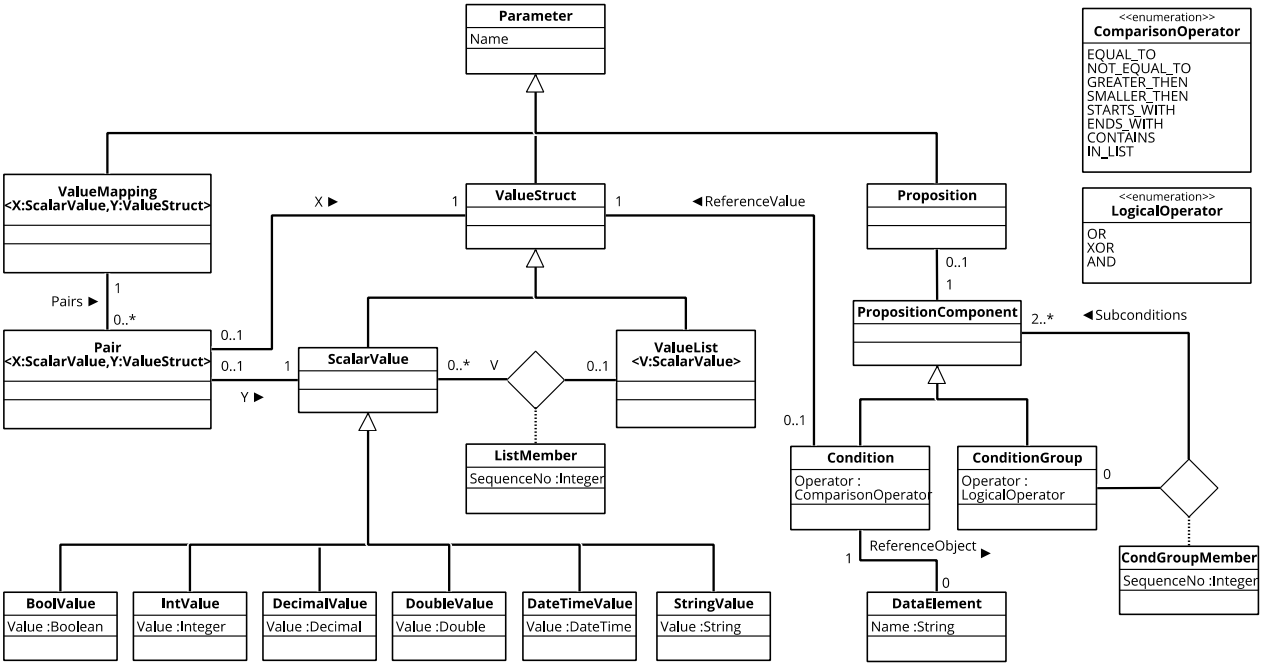

Figure 12: Metamodel for modelling BI activity parameters

8. From assumptions to activity parameters

In chapter 6 we introduced the concept of *assumptions* within the context of BI processes. We also discussed how these are identified and which part of the activity (data processing fragment) corresponds to this assumption. In chapter 4 we discussed how to discover a workflow model from an as-is BI process. Chapter 7 defines a conceptual metamodel that facilitates configurability of the process through metadata. In this chapter we discuss how to remove assumptions by transforming them into configurable activity parameters (RQ5, section 1.2). The output is a *metadata driven workflow model,* as shown in Figure 13.

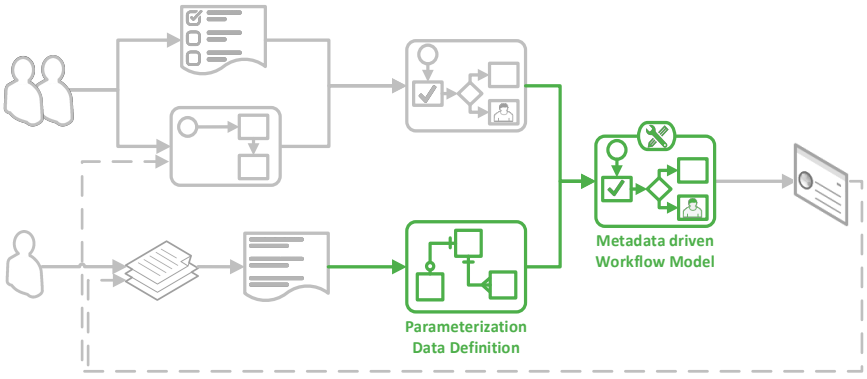

Figure 13: From assumptions to activity parameters in the context of BIPM

We follow a two-step approach, as shown in section 8.2. Section 8.4 covers the design of assumption into a parameterization definition. Then, in section 8.4 we take this specification and extend our *workflow model* (chapter 5) with parameterization of data processes activities.

8.1 Related work

In section 7.1.2 we discussed the concepts within BPMN for handling data: *data objects* and *properties* (Ter Hofstede et al., 2010). Data objects are graphically shown as documents. We noticed that the BPMN language does not provide a taxonomy of data types. Properties are more precise then data objects and are mapped to data variables in process execution languages, but they are not graphically visible (Ter Hofstede et al., 2010). Ter Hofstede et al. propose a hybrid method to model data: use properties to specify (XML schema) data types and data objects for documentation. Alternatively, annotations on BPMN activities provide a way to show production and consumption of ETL data artifacts (El Akkaoui & Zimanyi, 2009).

BPEL is a language for defining executable business processes and supports notions of data natively (Dumas et al., 2013; Ter Hofstede et al., 2010). However, due to the missing graphical aspect, BPMN may be more suitable for communication purposes (El Akkaoui & Zimanyi, 2009).

8.2 Approach

We follow a two-step approach to remove assumptions by transforming them into configurable activity parameters. First, the data modeler defines the to-be modelled parameter as data structure (section 8.3). Then, the process analyst extends the elicitated *workflow model* with notions of configuring parameter values and feeding them to process activities (section 8.4). The resulting *metadata driven workflow model* provides input for implementation to a developer and application architect.

8.3 Specification

In section 6.2 we discussed the identification of assumptions using an evidence-based approach. An identified assumption involves 1) a data processing fragment (code) and 2) background information on the (domain-specific) context. Using these two information components, a data modeler is able to model a parameter definition:

1. Determine a name and textual definition of the to-be modelled parameter.
2. Using the metamodel of section 7.4, model a data structure that fulfills this definition.
    a. Pick a suitable generic data structure (*ScalarValue*, *ValueList*, *ValueMapping* or *Proposition*).
    b. Pick suitable types for UML template variables.
    c. Realize the data type by using object initialization and assign the value of the *Name* attribute.
    d. Optional: pre-populate the data structure with default values upon instance initialization.

When all parameters are specified, the *parameterization data definition* is complete. In the following sections, we demonstrate our specification approach using two concrete examples.

8.3.1 *Example: threshold for exceptional invoices*

Suppose we identified the following assumption: *sales invoices with a value greater than $ 100.000 are assumed to be exceptional*. The related data processing fragment (SQL) is shown in Figure 14.

```
SELECT i.invoice_id, i.invoice_value
INTO results_exceptional_invoices
FROM invoices AS i
WHERE i.invoice_type = 'SALES' AND i.invoice_value > 100000
AND ……
```

Figure 14: Example data processing fragment of single value parameter

We design a parameter called *Sales Invoice Threshold Value* with definition: *the maximum expected invoice value of regular invoices*. We choose *ScalarValue* as generic data structure since we are modelling a single value. We create an object from the specialized class *DecimalValue* and initialize the name property. We pre-populate the data structure with Value = 100000.

8.3.2 *Example: classification of companies to business regions*

Suppose we identified the following assumption: *all business entities are located in the Americas region, except entities with 'Europe' in the company name.* The related data processing fragment (SQL) is shown in Figure 15.

```
SELECT b.entity_id, b.entity_name, (CASE b.company_name LIKE '% Europe' THEN
'EMEA' ELSE 'Americas') AS business_region, s.turnover
INTO internal_company_details
FROM business_entities AS b, business_entity_stats s
ON b.entity_id = s.entity_id
WHERE b.company_type = 'INTRACOMPANY'
AND ……
```

Figure 15: Example data processing fragment of mapping parameter

We design a parameter called *Business Region by Company ID* with definition: *the mapping of companies to their business region*. We choose *ValueMapping* as generic data structure since we are modelling a single value. We pick X = IntValue, Y = StringValue for the UML template variables. We create an object of *ValueMapping<IntValue,StringValue>* and initialize the name property.

8.4 Integration into workflow model

As discussed in 8.1, a way of documenting an formalizing data in BPMN is to use a combination of data objects and properties. Alternatively to data objects, activity annotations can be used instead. In this section we propose a method to extend the workflow model of Chapter 5 to a metadata driven workflow model. Additional information is added to indicate at which control point the user performs parameterization.

Based on the information of section 8.3, a modeler is able to integrate each parameter using a structured approach. For the graphical aspect, either *data objects* or *data annotations* are used. *Data objects* show directly the *production* and *consumption* relation, but make the model unreadable when many *data objects* are involved. In the latter case, we propose to use *data annotations* instead.

Integration with Data Objects

1. Add a *Input Data Object* and assign the name property;
2. Draw a *Data Output Association* from the *Data Object* to the activity that contains the corresponding code fragment. Draw also associations to other activities if they also include the same assumption.
3. If the user requires output from an activity to determine the parameter value: 1) insert a preceding activity that provides this guidance (if it does not exist yet) and 2) draw a *Data Input Association* from the activity to the *Data Object*.
   Otherwise: do not link any *Data Input Associations* to the *Data Object*.
4. Add a *Property* for each *Activity* that has a *Data Object* attached. Initialize the name and type using the name attribute of the *Data Object*.

5. In a similar way, add process parameters for the *Data Object.*
6. Attach a type definition for the parameter. For example, a XSD definition that defines the concretized type definition of the parameter or a realization using an UML Object Diagram.
7. Update the content of the activities to refer to the parameterization name instead of the hard-coded data assumption.

Alternative: integration with Data Annotations

1. Add an annotation to the activity that contains the corresponding code fragment. Use "Input:" followed by the name as description. Repeat the same for other activities if they also include the same assumption.
2. If the user requires output from an activity to determine the parameter value: 1) insert a preceding activity that provides this guidance (if it does not exist yet) and 2) add an annotation to the activity. Use "Output:" followed by the name as description.
3. Add a *Property* for each *Activity* that references the parameter. Initialize the name and type using the name attribute of the parameter.
4. In a similar way, add process parameters for the parameter.
5. Attach a type definition for the parameter. For example, a XSD definition that defines the concretized type definition of the parameter or a realization using an UML Object Diagram.
6. Update the content of the activities to refer to the parameterization name instead of the hard-coded data assumption.

We show *integration with Data Objects* in details using a representative example.

8.4.1 *Example: classification of companies to business regions*

In this example we will take the specification of section 8.3.2 and a simplified process model (Figure 16).

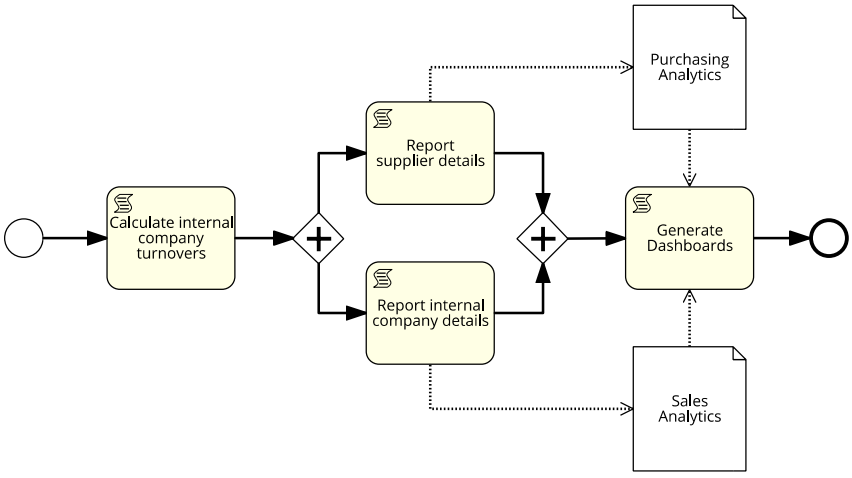

Figure 16: Example of as-is process model for parameterization

The data processing fragment resides in activity *Report internal company details*. The activity *Retrieve internal companies* will show the user a list of member firms. We add an *Input Data Object* called *Business Region by Company ID* (step 1), draw an output arc to *Report internal company details* (step 2). The user bases the mapping on the list of internal companies. Therefore we insert the activity *Retrieve internal companies* and draw an input arc from this activity (step 3). We add a property to the activities involved (step 4) and to the level of the process (step 5). The updated process model is shown in Figure 17.

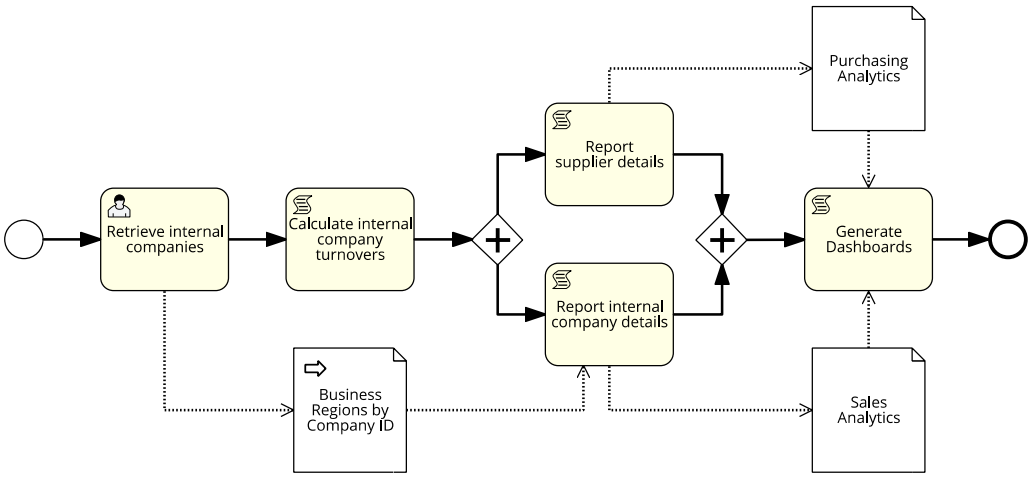

Figure 17: Example of parameterized process model

We then define the type specification of the parameter as an UML Object Diagram to (step 6, Figure 18). The updated content of the processing fragment (section 8.3.2) is shown in Figure 19.

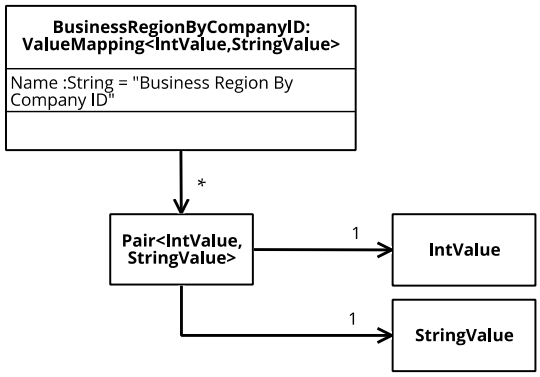

Figure 18: UML Object Diagram of example parameter

```
SELECT b.entity_id, b.entity_name, invoke_mapping('BusinessRegionByCompanyID',
b.entity_id) AS business_region, s.turnover
INTO internal_company_details
FROM business_entities AS b, business_entity_stats s
ON b.entity_id = s.entity_id
WHERE b.company_type = 'INTRACOMPANY'
AND ……
```

Figure 19: Update SQL code fragment for example parameter

9. Implementation

In this chapter we discuss the requirements for implementing the workflow model (RQ6, section 2.7). In chapter 8 we constructed an *metadata driven workflow model* using the notions of BPMN. In this chapter we provide an approach to select an appropriate workflow engine for this model. This workflow engine forms the basis for implementation (Figure 20).

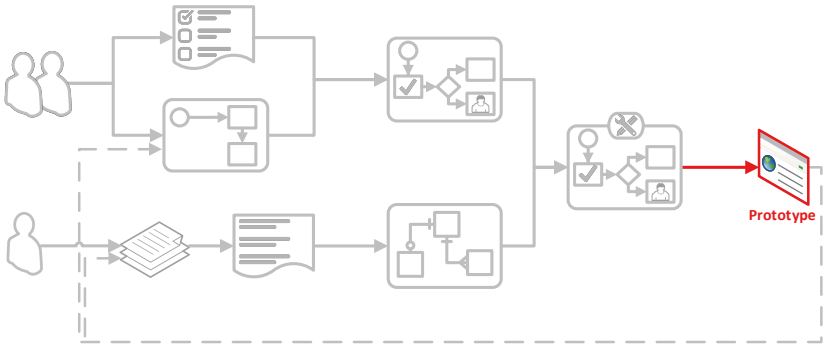

Figure 20: Implementation in the context of BIPM

In section 9.1 we discuss related work in implementation of prototypes and the architectural characteristics of workflow management systems. Then, in sections 9.2 and 9.3 we discuss the implementation approach for the resulting workflow model of BIPM.

9.1 Related work

In the past decade, various workflow technology became available. In this section we consider the characteristics of workflow system concepts and prototype implementation approaches. We describe the notions of Business Process Management Systems, Scientific Workflow Management Systems and related work on pattern-based evaluation of workflow systems.

9.1.1 *Prototyping*

A software prototype is defined as the partial or preliminary implementation of a new software product (Wiegers & Beatty, 2013). Davis Bersoff, Edward & Comer (1988) identified three prototyping approaches:

- *Rapid throwaway prototyping:* aims to ensure that user requirements are met in the software product. Potential users use the prototype and provide feedback back to the project team. In parallel, the team will build the actual system;
- *Incremental development:* construction of a partial implementation of the system (e.g. a particular module), slowly extending the system with more modules;
- *Evolutionary prototyping:* construct a solid foundation based on known requirements. Requirements become fine-grained while users use the system. Agile development is an example of this approach (Wiegers & Beatty, 2013).

9.1.2 *Workflow Management Systems*

Organizational processes are categorized into material processes, information processes and business processes (Georgakopoulos et al., 1995). Business processes are describing high-level activities of organizations. These are realized using information processes and/or material processes. Information technology enables organizations to automate standardized information processes. Workflow Management concepts are often used in such environments. These systems are also known as *Business Process Management Systems*. Another application is the support from workflow technology in scientific work: *Scientific Workflow Management Systems*. In contrast to common business processes, this class of workflow systems is focused to the handling of scientific tasks using dataflow constructs.

9.1.3 *Business Process Management Systems*

The Workflow Management Coalition standardized the concepts of Workflow Management Systems for business purposes into a Workflow Reference Model (Hollingsworth, 1995). Notable parts of this reference model are the concepts of *process definition*, *enactment*, *application interoperability* and *workflow clients*. Grefen & De Vries (1998) introduced a system reference architecture based on this reference model. Input for the architecture are four key design principles:

- *D1: Top-down design strategy*: ability to modularize the workflow system in components;
- *D2: Separation of enactment and design perspectives*: distinguish design time from runtime operations;
- *D3: Separation between kernel functionalities and additional functionalities*: system extensibility to support incremental system design and installation;
- *D4: Explicit interfaces between the WFMS and the software platforms*: separation of concerns and platform independence.

The *global architecture* consists of three main components: *WF Design Module* as service for defining the processes, *WF Server Module* for centralizing workflow enactment and *WF Clients Module* to facilitate end user communication with the workflow instance.

9.1.4 *Scientific Workflow Management Systems*

Scientific Workflow Management Systems (SWFMS) define, manage and enact scientific workflows (Chebotko & Fotouhi, 2009). The goal and environment of scientific workflows differs from that of business workflows. Business workflow aim to reduce human resources and costs and scientific workflows aim to accelerate the process of discovering new findings from large amounts of data (Chebotko & Fotouhi, 2009). Furthermore Chebotko et al. (2009) indicate that business workflows are typically control flow oriented and scientific workflows more dataflow oriented. Another key difference is the flexibility aspect. Scientific workflows are often not completed before they start (Wainer, 1997). This implies that the scientist has the role of designer and the end user simultaneously, due to their exploratory nature.

Chebotko et. al (2009) identified seven key architectural requirements for SWFMs:

- *R1: User interface customizability and user interaction support:* the ability to customize the user interface to the scientific context.
- *R2: Reproducibility support:* management of provenance metadata to reproduce the outcome of the workflow.
- *R3: Heterogeneous and distributed services and software tool integration*: integration of various analytical software as workflow tasks to solve complex scientific problems.
- *R4: Heterogeneous and distributed data product management*: support efficient management of data products (workflow source data, workflow parameters and workflow results).
- *R5: High-end computing support*: separate science-focused problem solving environment (PSE) from high-end computing infrastructure (grid/cloud computing).
- *R6: Workflow monitoring and failure handling:* provide support for failure monitoring in ad hoc workflow design.
- *R7: Interoperability:* ability to collaborate with other SFWFMs in collaborative research projects.

9.1.5 *Workflow Patterns*

A systematic approach in evaluating capabilities of workflow engines is to indicate requirements as *workflow patterns*. Riehle & Züllighoven (1996) define patterns as: *the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts*. Russell, Hofstede & Edmond (2003) identified 20 control-flow oriented patterns and evaluate their support in commercially available workflow management systems. Russell (2007) extends the taxonomy of *workflow patterns* on the data, resource and exception handling perspectives and evaluates their support in commercial products. A detailed evaluation of workflow pattern support in commercial and open source products, as well as standards (e.g. BPEL) is publicly available[5].

9.2 Prototyping approach

In BIPM, we propose to use prototyping to validate completeness of the workflow. Depending on the application context, this corresponds to either *rapid throwaway prototyping* or *evolutionary prototyping*. Implementation of the workflow is a joint effort of the application architect and developer.

9.3 Workflow Management System support for BIPM

Choosing a suitable workflow engine to implement a metadriven workflow model is a complex task, there is a wide variety of open source and proprietary workflow engines available, especially in the BPMS domain (Russell, 2007) but also in the area of SWFMS (Curcin & Ghanem, 2008).

[5] http://www.workflowpatterns.com/evaluations/

First, we elaborate the applicability of BPMS and SWFMS for metadata driven BI workflows. After that, we propose a systematic approach to select a suitable workflow engine.

9.3.1 *BPMS vs. SWFMS*

We need to determine which class of Workflow Management Systems is suitable. As discussed in section 9.1.4, SWFMS are more data focused and have a more exploratory nature. For BIPM, only parts of the characteristics of SWFMS are applicable. R1 and R7 (section 9.1.4) are not relevant since the methodology is applied in an industrial BI setting. R2, R4 and R6 correspond to core data handling requirements and are applicable to BI workflows. R3 and R5 relate to *analytical maturity* and *data volume* topics. The relevance of these requirements dependents on the application context. In section 9.1.3 we referred to four design principles that were used for designing a BPMS reference architecture. Principle D2 describes the ability to separate the design of the workflow with the execution of the process. However, in SWFMS the scientist simultaneously acts as the process designer and the user. In BIPM, the latter use case is not applicable; when we start with the implementation, a predefined workflow model is already available.

To conclude, we identified several key differences between the BPMS and SWFMS concepts. SWFMS are more data-focused but have no strict separation between design time and runtime. In the end, the choice for the workflow management system type depends on the application setting.

9.3.2 *Selection of an appropriate workflow engine*

As mentioned in section 9.1.5, an approach to evaluate capabilities of a workflow engine is the use of *workflow patterns*. The set of required workflow patterns dependents on the designed *metadata driven workflow model* (section 8.4), therefore the evaluation is context-dependent.

We propose the following evaluation approach to systematically identify the requirements of a Workflow Management System.

1. Identify the control-flow, data, resource and exception handling patterns from the *metadata driven workflow model*.
   An overview of patterns in BPMN is provided in (Wohed, Aalst, & Dumas, 2005).
2. For each candidate Workflow Management System check whether all identified patterns are featured in the product.
   An detailed evaluation of well-known Workflow Management Systems is available at http://www.workflowpatterns.com/evaluations/

9.3.3 *Contextual requirements*

Next to the functional requirements of the workflow engine, other context-specific requirement might apply. Additional requirements come from the application architect of the project team. Some enterprises defined a *list of preferred technology partners* or have *compliance regulations* in place.

10. Case study

CONFIDENTIAL

11. Conclusions and recommendations

The research goal of this thesis was to develop a methodology to reduce expert knowledge dependence in BI solutions. This thesis proposes Business Intelligence Process Management (BIPM), a systematic methodology to discover, model and parameterize expert knowledge dependent BI processes. We present our findings and conclusions (section 11.1), contributions (section 11.2) and recommendations for KPMG (section 11.3). Finally, we provide suggestions for future work (section 11.4).

11.1 Summary of findings and conclusions

The main research problem of this thesis is: *how to reduce dependency on expert knowledge in order to scale enterprise BI solutions.* Our goal is to *develop a methodology to reduce expert knowledge dependence in BI solutions*. Six research questions form the basis for our methodology research and design. We present our core findings according to these research questions:

- *RQ1: Which main stakeholders and interests can be identified in the process of reducing expert knowledge dependence in BI solutions?*

  Stakeholder management is required to prevent rework of requirements specification and to increase the chance of adoption. We identified stakeholders on three levels: 1) outside the development organization; direct users, indirect users, program managers and beta testers, 2) the development organization; development manager and subject matter expert, and 3) the project team; process analyst, developer and application architect. We provided a role matrix that maps these stakeholders to BIPM activities. Many stakeholders are involved in the *process flow elicitation process.*

- *RQ2: How to elicitate the process flow of an undocumented BI workflow process?*

  We constructed the Process Discovery Application Framework which consists of seven important elements of process discovery in the field of BPM: purpose, language, aspects, approach, challenges, discovery methods and quality assurance. For process flow elicitation, a combination of interviewing, observation and document analysis methods is proposed. Input is required from the program manager, subject matter expert and direct users and involves fives phases: 1) identify the process boundaries, 2) identify the activities and events, 3) identify resources and their handovers, 4) identify the control flow, and 5) identify additional elements. The result is a BPMN workflow model were the control flow is based on the data flow dependencies. Optionally, additional input is possible through feedback from beta users of the to-be implemented prototype.

- *RQ3: How to identify assumptions made during data transformations and analytics?*

  BI processes may have assumptions regarding the data format, context and semantics of the raw data. An evidence-based discovery technique is provided, consisting of: 1) introducing direct users to the goal and recording scheme, 2) documentation of customizations by users and 3) analyzing the assumptions that caused these changes, together with the subject matter expert. Additionally, a XES extension is provided for automated logging.

- *RQ4: Which types of process parameterizations can be identified, and how are these modelled?*

  Four parameter types are identified: single value, list of values, mapping and proposition types. Modelling of parameters relates to flexibility by design. BPMN leaves freedom to model data structures, therefore we used the formalism of UML Class Diagrams. A conceptual metamodel is designed for modelling BI activity parameters. This metamodel is flexible enough to capture all four frequent parameterization types.

- *RQ5: How to transform data assumptions into parameterized process activities?*

  First, a modeler defines the to-be modelled parameter as data structure, based on the BI activity parameters metamodel. Then the process analyst extends the workflow model with data artifacts, to 1) configure parameter values and 2) feed activities with parameter values (output). Either *Data Objects* or *Data Annotations* can be used as data artifacts. The result is a metadata driven BI workflow model that provides flexibility by design.

- *RQ6: How to successfully implement the metadata driven workflow model?*

  Before starting the implementation, the prototyping approach is defined: rapid throwaway, incremental or evolutionary. Depending on the modelling context, business process management systems (BPMS) or scientific workflow systems (SWFS) can be used. SWFMS are more data-focused but have no strict separation between design time and runtime. Workflow patterns support the evaluation of selecting an appropriate workflow engine. Next to the functional fit, additional contextual requirements (such as technology partnerships) might apply.

In this study we developed the BIPM methodology to transform knowledge dependent BI processes into metadata driven BI workflows. The methodology consists of a blueprint and a structured descriptive approach on five phases: 1) Stakeholder analyses, 2) Process flow elicitation, 3) Identification of assumptions, 4) From assumptions to activity parameters, 5) Implementation. We based design decisions on related work from the BPM and Workflow Management domains.

Using a case study in a representative industry scenario, we validated the correctness and completeness of the proposed methodology. Our case study was performed at KPMG Advisory, in the domain of tax analytics. We aligned the global business goal of KPMG's Tax Intelligence Solution with our research goal. All steps of BIPM are performed in the case study, including the

implementation of a prototype. The implemented workflow facilitates orchestration and execution of the end-to-end BI process of VAT analytics. We quantitatively validated correctness and completeness successfully using two client datasets from different industry domains.

11.2 Contributions

In our design study, we contributed in two areas. From a *research* perspective, we propose the BIPM methodology to reduce expert knowledge using a systematic approach. From an *industry* perspective, we successfully developed a proof-of-concept for KPMG based on the BIPM approach.

11.2.1 *Research*

Review of related work showed a few papers on the conceptual modelling BI processes. El Akkaoui & Zimanyi (2009) proposes a platform-independent conceptual model of ETL processes based on BPMN. Another approach is artifact-centric based modelling and is built on the relational algebra concepts (Simitsis, 2005; Vassiliadis et al., 2002). These approaches are focused on the modelling phase and are designed at the abstraction level of elementary data processing operations, such as filling in an additional column in the dataset.

In our study, we researched the subject in the context of a *specific goal*, and from a *broader project perspective*: reducing expert knowledge dependence by discovering, modelling and parameterizing BI workflows using metadata. The *specific goal* relates to the problem of reducing expert knowledge dependence to scale up enterprise BI solutions. Instead of solely focusing on the modelling task, we approach the problem activity-based and in a *broader project perspective*: guiding the process from stakeholder analyses to implementation. BIPM is evidence-based, focusses on data-driven processes and enables configurability by design by modelling the workflow metadata driven. Our methodology is built on theory from the BPM and Workflow Management Systems research areas. Furthermore, our methodology is platform-independent to prevent thinking on implementation issues early in the process.

11.2.2 *Industry*

CONFIDENTIAL

11.3 Recommendations for KPMG

CONFIDENTIAL

11.4 Future work

During our study, we identified some aspects to further extend the line of research. This section provides the suggestions for further research.

The first aspect considers the application scope of the proposed methodology. The design of our methodology is focused to BI processes with a single begin and end state, which follow the classical data warehousing paradigm. Furthermore, our goal is to increase maturity from EBIMM level 1 (Chuah, 2010) to level 2 (or 3). Van der Lans (2012) identifies two significant BI trends: operational business intelligence and the advent of big data. These trends relate to the capabilities (e.g. predictive analyses) of higher maturity levels, in which the ETL processing pattern is replaced by *data virtualization* (Van der Lans, 2012). Explorative analysis and flexible BI process flows are also capabilities to consider in this research field. We suggest further research in extensions of BIPM to support such architectural BI paradigms.

Another aspect relates to minimizing the implementation effort of the metadata driven workflow model. BPMN is used to model workflow models as input for enactment. Notions of data artifacts are used to model parameterization. Related work shows that these concepts can be transformed into executable BPEL processes with relatively little effort (El Akkaoui & Zimanyi, 2009; Ter Hofstede et al., 2010). We suggest further work in formal model transformation to specific workflow engines, for example Bizagi (BPMS) or Rapid Miner (SWFMS).

Furthermore, we propose further research in acceptance testing. Our implementation guidelines consider the contextual and functional requirements of workflow engine technology. BIPM facilitates a feedback loop for extending the workflow model driven by functional needs. We did not incorporate a systematic approach of full end-user acceptance testing in BIPM. Although beta users of the case study were enthusiastic on the capabilities of the developed workflow client, we propose an extension of the methodology that provides a structured end-user acceptance testing method.

Xu, Liao, Zhao & Wu (2011) propose a metadata-driven service model as basis for reusable ETL processes. Their proposed service framework is based on the platform as a service (PaaS) concept, including process definition and parameter configuration services. We suggest further research in applying their architectural design for implementing BIPM workflow models.

12. Bibliography

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, *26*(1), 65–74. http://doi.org/10.1145/248603.248616

Chebotko, A., & Fotouhi, F. (2009). A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution. *IEEE Transactions on Services Computing*, *2*(1), 79–92. http://doi.org/10.1109/TSC.2009.4

Chuah, M.-H. C. M.-H. (2010). An enterprise business intelligence maturity model (EBIMM): Conceptual framework. *Digital Information Management (ICDIM), 2010 Fifth International Conference on*, 303–308. http://doi.org/10.1109/ICDIM.2010.5664244

Curcin, V., & Ghanem, M. (2008). Scientific workflow systems - can one size fit all? *2008 Cairo International Biomedical Engineering Conference*. http://doi.org/10.1109/CIBEC.2008.4786077

Davis, Alan M., Bersoff Edward H., Comer, E. R. (1988). A strategy for comparing alternative software development life cycle models. *Software Engineering, …*, *14*(10), 1453–1461. http://doi.org/10.1109/32.6190

Devlin, B. a., & Murphy, P. T. (1988). An architecture for a business and information system. *IBM Systems Journal*, *27*(1), 60–80. http://doi.org/10.1147/sj.271.0060

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. a. (2013). *Fundamentals of business process management*. Retrieved from http://www.worldcat.org/oclc/828794599

El Akkaoui, Z., & Zimanyi, E. (2009). Defining ETL worfklows using BPMN and BPEL. *Proceeding of the ACM Twelfth International Workshop on Data Warehousing and OLAP DOLAP 09*, 41–48. http://doi.org/10.1145/1651291.1651299

Georgakopoulos, D., Georgakopoulos, D., Hornick, M., Hornick, M., Sheth, A., & Sheth, A. (1995). An Overview of Worklow Management: From Process Modeling to Worklow Automation Infrastructure. *Work*, *153*, 119–152.

Giudici, P. (2003). *Applied Data Mining: Statiscal Methods for Business and Industry*.

Grefen, P., & de Vries, R. R. (1998). A reference architecture for workflow management systems. *Data & Knowledge Engineering*, *27*(1), 31–57. http://doi.org/10.1016/S0169-023X(97)00057-8

Hollingsworth, D. (1995). The Workflow Reference Model. *Management*, *am*(1), 1–55. http://doi.org/citeulike-article-id:1378584

IIBA. (2009). *A Guide to the Business Analysis Body of Knowledge*. *International Institute of Business Analysis*.

Jablonski, S., & Bussler, C. (1996). *Workflow Management: Modeling Concepts, Architecture, and Implementation*.

List, B., & Korherr, B. (2006). An Evaluation of Conceptual Business Process Modelling Languages. *2006 ACM Symposium on Applied Computing*, (Section 3), 1532–1539. http://doi.org/10.1145/1141277.1141633

Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., & Reijers, H. a. (2012). Imperative versus declarative process modeling languages: An empirical investigation. *Lecture Notes in Business Information Processing*, *99 LNBIP*(PART 1), 383–394. http://doi.org/10.1007/978-3-642-28108-2_37

Ramezani, E., Fahland, D., & van der Aalst, W. M. P. (2014). Supporting domain experts to select and configure precise compliance rules. *Lecture Notes in Business Information Processing*, *171 LNBIP*, 498–512. http://doi.org/10.1007/978-3-319-06257-0

Redding, G., Dumas, M., Hofstede, A. H. M., & Iordachescu, A. (2010). A flexible, object-centric approach for business process modelling. *Service Oriented Computing and Applications*, *4*(3), 191–201. http://doi.org/10.1007/s11761-010-0065-4

Riehle, D., & Züllighoven, H. (1996). Understanding and using patterns in software development. *Tapos*, *1*, 14. http://doi.org/10.1002/(SICI)1096-9942(1996)2:1<3::AID-TAPO1>3.0.CO;2-#

Russell, N. (2007). Foundations of process-aware information systems. *Language*, (December), 1–419. Retrieved from http://eprints.qut.edu.au/16592/

Schonenberg, H., Mans, R., Russell, N., Mulyar, N., & van der Alst, W. (2008). Processflexibility: A survey of contemporary approaches. *Lecture Notes in Business Information Processing*, *10*(Part I), 16–30.

Simitsis, A. (2005). Mapping conceptual to logical models for ETL processes. *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP - DOLAP*, 67. http://doi.org/10.1145/1097002.1097014

Smith, L. W. (2000). Project Clarity Through Stakeholder Analysis What Is a Stakeholder ? Importance of Stakeholder Analysis Organizational and Project Spotlight on Stakeholders Stakeholder Analysis Approach.

Stachowiak. (1973). Allgemeine Modelltheorie.

Ter Hofstede, A. H. M., Van der Aalst, W. M. P., Adams, M., & Russell, N. (2010). *Modern Business Process Automation*.

Van der Aalst, W. M. P. Van Der, Dreiling, a, Gottschalk, F., Rosemann, M., & Jansen-Vullers, M. H. (2006). Configurable Process Models as a Basis for Reference Modeling. *Business Process Management Workshops*, 512–518. http://doi.org/10.1007/11678564_47

Van der Aalst, W. M. P., Ter Hofstede, a. H. M., Kiepuszewski, B., & Barros, a. P. (2003). Workflow patterns. *Distributed and Parallel Databases*, *14*(1), 5–51. http://doi.org/10.1023/A:1022883727209

Van der Lans, R. (2012). *Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses*.

Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2002). Conceptual Modeling for ETL processes, 14–21.

Wainer, J. (1997). Scientic workflow systems. *Work*, 1–5.

Wiegers, K., & Beatty, J. (2013). *Software Requirements* (Third Edit).

Wilbik, A., & Kaymak, U. (2013). *Business Intelligence Basics*.

Wohed, P., Aalst, W. M. P. Van Der, & Dumas, M. (2005). Pattern-based Analysis of BPMN. *Technology*, *14*(6), 781–787. http://doi.org/10.1197/jamia.M2389

Xu, L., Liao, J., Zhao, R., & Wu, B. (2011). A PaaS based metadata-driven ETL framework. *CCIS2011 - Proceedings: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 477–481. http://doi.org/10.1109/CCIS.2011.6045113

Appendix A

Metamodel for logging customizations as XES extension

```
<xesextension name="ActivityCustomizations" prefix="ad"
uri="http://www.example.org/actdev.xesext">
	<event>
		<list key="list">
			<container key="customizations">
				<string key="customizationtype" />
				<string key="description" />
				<string key="changeset" />
			</container>
		</list>
	</event>
</xesextension>
```

Graphics resources

- Settings Icon Black - http://untvapp.com/
  http://creativecommons.org/licenses/by-nc-sa/3.0/
- Check Icon - http://lov.okfn.org/dataset/lov/
  http://creativecommons.org/licenses/by/4.0/