ulm university universität
uulm

Institut für Datenbanken und Informationssysteme
Leiter: Prof. Dr. Manfred Reichert

Process-Oriented Information Logistics: Aligning Process Information with Business Processes

DISSERTATION
zur Erlangung des Doktorgrades Dr. rer. nat.
der Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
der Universität Ulm

vorgelegt von
BERND MICHELBERGER
aus Bad Saulgau

2015

Amtierende Dekanin:	Prof. Dr. Tina Seufert
Gutachter:	Prof. Dr. Manfred Reichert Prof. Dr. Franz Schweiggert Prof. Dr. Bela Mutschler
Tag der Promotion:	22. Juli 2015

Preface

The results presented in this thesis are the outcome of my work as a research assistant at the Institute of Databases and Information Systems, Ulm University as well as at the Institute of Applied Research, University of Applied Sciences Ravensburg-Weingarten.

I am truly grateful to my mentor Prof. Dr. Manfred Reichert. Dear *Manfred*, thank you for your continuous support, helpful advice, and valuable feedback during the last few years. You gave me all the needed freedom for my research, but at the same time you were always there for me. I know that the last few years were challenging, especially, because our favorite football club goes through more than one fundamental crisis.

I am also deeply grateful to my mentor Prof. Dr. Bela Mutschler. Dear *Bela*, you are such a great inspiration to me and my work. I have never met a person like you. My thesis would not have been possible without your extensive help and support. I cannot even express how lucky I feel that I had the opportunity to be your PhD student. I have learned more from you than you can imagine. No matter what problem and worry I had, you were always there for me. Bela, thank you for being a part of my thesis.

Besides my mentors, I would like to thank all members of my doctoral committee for reading my thesis and giving me feedback: Juniorprof. Dr. Birte Glimm, Prof. Dr. Enrico Rukzio, Prof. Dr. Uwe Schöning, and Prof. Dr. Franz Schweiggert. Dear *Birte*, thank you for your patience and feedback when Klaus and I wrote the paper last year.

I would also like to thank my colleagues and friends from the niPRO project for their support: Daniel Binder, Jürgen Botsch, Hasan Girit, Dr. Markus Hipp, Marco Kollöffel, Jörn Kuppe, Florian Malsam, Jan Meurer, Armin Reisch, and Christian Werb.

I would also like to thank my co-authors with whom I worked in the last few years: Ralph-Josef Andris, Robert Eberhard, Achim Strauss, and Klaus Ulmschneider.

Furthermore, I want to thank the following people for their support during the work on this thesis: Prof. Dr. Peter Dadam, Frank Eichfeld, Dr. Jens Kolb, Michael Offergeld, Rüdiger Pryss, Dr. Karin Schmidt, Lorenz Widmann, and Dr. Jörg Wurzer.

I would also like to thank my colleagues and friends from Ulm University. Even if I have met you only once a month, it was really a great time to be with you.

Dear *Claudia*, thanks for your help. You are one of the best things that ever happened to me. Thanks for your love and good sense of humor. I am looking forward to our future.

Last but not least, I would like to thank my parents Carmen and Willi Michelberger. Dear *Mom* and *Dad*, thank you for your support, unconditional love, and believing in me. When I was a toddler, you have read my favorite bedtime stories. When I was a child, you cheered me up during my football matches. When I was a teenager, you showed me the right way. When I was a PhD student, you supported me emotionally and financially. For all of these things, and many more, I am deeply thankful.

Abstract

During the last decade, research in the field of *business process management* (BPM) has focused on the design, modeling, execution, monitoring, and optimization of business processes. What has been neglected, however, is the provision of knowledge workers and decision makers with needed information when performing knowledge-intensive business processes such as product engineering, customer support, or strategic management.

Today, knowledge workers and decision makers are confronted with a massive load of data, making it difficult for them to discover the information relevant for performing their tasks. Particularly challenging in this context is the alignment of process-related information (*process information* for short), such as e-mails, office files, forms, checklists, guidelines, and best practices, with business processes and their tasks.

In practice, process information is not only stored in large, distributed and heterogeneous sources, but usually managed separately from business processes. For example, shared drives, databases, enterprise portals, and enterprise information systems are used to store process information. In turn, business processes are managed using advanced process management technology. As a consequence, process information and business processes often need to be manually linked; i.e., process information is hard-wired to business processes, e.g., in enterprise portals associating specific process information with process tasks. This approach often fails due to high maintenance efforts and missing support for the individual demands of knowledge workers and decision makers.

In response to this problem, this thesis introduces *process-oriented information logistics* (POIL) as new paradigm for delivering the right process information, in the right format and quality, at the right place and the right point in time, to the right people. In particular, POIL allows for the process-oriented, context-aware (i.e., personalized) delivery of process information to process participants. The goal is to no longer manually hard-wire process information to business processes, but to automatically identify and deliver relevant process information to knowledge workers and decision makers.

The core component of POIL is a *semantic information network* (SIN), which comprises homogeneous *information objects* (e.g., e-mails, office files, guidelines), *process objects* (e.g., tasks, events, roles), and *relationships* between them. In particular, a SIN allows discovering objects linked with each other in different ways, e.g., objects addressing the same topic or needed when performing a particular process task.

The SIN not only enables an integrated formal representation of process information and business processes, but also allows determining the relevance of process information for a given work context based on novel techniques and algorithms. Note that this becomes crucial in order to achieve the aforementioned overall goal of this thesis.

Background Information

The presented research is performed in the *niPRO* project. Goal of this project is to use innovative semantic technology to integrate relevant process information in intelligent, user-adequate *process information portals*. The project is funded by the *German Federal Ministry of Education and Research* (BMBF) under grant number 17102X10. More information can be found at http://nipro.hs-weingarten.de.

Parts of this thesis have been published in the following referred papers:

[1] B. Michelberger, B. Mutschler, and M. Reichert. *Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter.* In: Proc 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107–120, Hamburg, Germany, 2011.

[2] B. Michelberger, B. Mutschler, and M. Reichert. *On Handling Process Information: Results from Case Studies and a Survey.* In: Proc BPM'11 Workshops, 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333–344, Clermont-Ferrand, France, 2011.

[3] B. Michelberger, B. Mutschler, and M. Reichert. *Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes.* In: Proc 16th Int'l Enterprise Computing Conf (EDOC'12), IEEE Computer Society Press, pp. 21–30, Beijing, China, 2012.

[4] B. Michelberger, B. Mutschler, and M. Reichert. *A Context Framework for Process-oriented Information Logistics.* In: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 260–271, Vilnius, Lithuania, 2012.

[5] H. Girit, R. Eberhard, B. Michelberger, and B. Mutschler. *On the Precision of Search Engines: Results from a Controlled Experiment.* In: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 201–212, Vilnius, Lithuania, 2012.

[6] B. Michelberger, A. Reisch, B. Mutschler, J. Wurzer, M. Hipp, and M. Reichert. *iCare: Intelligent Medical Information Logistics.* In: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), ACM Press, pp. 396–399, Vienna, Austria, 2013.

[7] B. Michelberger, B. Mutschler, M. Hipp, and M. Reichert. *Determining the Link and Rate Popularity of Enterprise Process Information.* In: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS'13), LNCS 8185, pp. 112–129, Graz, Austria, 2013.

[8] B. Michelberger, R. Andris, H. Girit, and B. Mutschler. *A Literature Survey on Information Logistics.* In: Proc 16th Int'l Conf on Business Information Systems (BIS'13), LNBIP 157, pp. 138–150, Poznań, Poland, 2013.

[9] M. Hipp, B. Michelberger, B. Mutschler, and M. Reichert. *A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information.* In: Proc 28th Symposium on Applied Computing (SAC'13), ACM Press, pp. 1383–1390, Coimbra, Portugal, 2013.

[10] B. Michelberger, B. Mutschler, D. Binder, J. Meurer, and M. Hipp. *iGraph: Intelligent Enterprise Information Logistics.* In: Proc 10th Int'l Conf on Semantic Systems (SEMANTiCS'14), Posters & Demonstrations Track, CEUR Workshop Proc 1224, pp. 27–30, Leipzig, Germany, 2014.

[11] M. Hipp, A. Strauss, B. Michelberger, B. Mutschler, and M. Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* In: Proc BPM'14 Workshops, 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395–407, Haifa, Israel, 2014.

[12] M. Hipp, B. Mutschler, B. Michelberger, and M. Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* In: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE Computer Society Press, pp. 1–12, Marrakesh, Morocco, 2014.

[13] B. Michelberger, K. Ulmschneider, B. Glimm, B. Mutschler, and M. Reichert. *Maintaining Semantic Networks: Challenges and Algorithms.* In: Proc 16th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'14), ACM Press, pp. 365–374, Hanoi, Vietnam, 2014.

[14] K. Ulmschneider, B. Michelberger, B. Glimm, B. Mutschler, and M. Reichert. *On Maintaining Semantic Networks: Challenges, Algorithms, Use Cases.* In: Int'l J of Web Information Systems (IJWIS), 11(3), pp. 291–326, 2015.

[15] B. Michelberger, M. Hipp, and B. Mutschler. *Process-oriented Information Logistics: Requirements, Techniques, Application.* In: Advances in Intelligent Process-Aware Information Systems, 2015. Accepted for Publication.

Table of Contents

List of Abbreviations

BMBF German Federal Ministry of Education and Research

BPM Business Process Management

BPMN Business Process Model and Notation

CEP Complex Event Processing

CIS Context Information System

CM Context Model

CPU Central Processing Unit

CRM Customer Relationship Management

DIKW Data-Information-Knowledge-Wisdom

EPC Event-driven Process Chain

GIS Global Information System

GPS Global Positioning System

HIS Hospital Information System

ICT Information and Communication Technology

IL Information Logistics

IM Information Management

IR Information Retrieval

IS Information System

IT Information Technology

KM Knowledge Management

KP Knowledge Problem

LP Link Popularity

OEM Original Equipment Manufacturer

OWL	Web Ontology Language
PAIS	Process-Aware Information System
PDF	Portable Document Format
POIL	Process-Oriented Information Logistics
ProNaVis	Process Navigation and Visualization
RDF	Resource Description Framework
RFID	Radio-frequency Identification
RP	Rate Popularity
SCM	Supply Chain Management
SIN	Semantic Information Network
SIN Facade	Semantic Information Network Facade
SWOT	Strengths-Weaknesses-Opportunities-Threats
URL	Uniform Resource Locator
VBA	Visual Basic for Applications
WP	World Problem
WWW	World Wide Web

Part I

Introduction

1 Motivation

Market globalization has led to massive cost pressure and increased competition for enterprises [16]. Products and services must be developed in ever-shorter cycles and new forms of collaboration within and across enterprises are continuously emerging. As examples consider distributed engineering processes in the automotive domain [17], supply chains [18, 19], or the treatment of patients in integrated healthcare networks [20]. To cope with these challenges, effective and efficient *business process management* (BPM) [21, 22] has become a crucial factor for enterprises and organizations.

For a long time, the support of business processes through *information and communication technology* (ICT) [23] has focused on the design, modeling, execution, monitoring, and optimization of business processes [24]. What has been neglected is the provision of knowledge workers and decision makers with needed information when performing knowledge-intensive processes [25] such as product engineering or strategic management.

A major problem in this context constitutes the increasing amount of information enterprises are confronted with [26, 27]. More precisely, the amount of information created, captured or replicated is growing with a rate of 65% a year [28]. Contemporary enterprises handle more than 63 terabytes of information annually on average [29]. Examples include e-mails, office files, process descriptions, forms, checklists, manuals, and best practices. In turn, this information is usually accessed through shared drives, databases, portals, or enterprise information systems. Thereby, process participants are not only interested in quickly accessing the information, but they also crave for pre-processed, complete and aggregated information when performing the tasks of a process [25, 30].

1.1 Problem Statement

Identifying relevant information in a given work context is by far more time-consuming and complex than just managing the information [31]. Problems frequently encountered include, for example, incomplete, incorrect, delayed, or outdated information [32]. Furthermore, quality requirements, such as accessibility, completeness, correctness, ease of interpretation, or reliability, might lead to uncertainties and problems [1, 33]. Think of financial information that needs to be up-to-date and complete in order to be able to correctly charge certain services by the accounting department.

Considering the present information load [26, 34], another challenge is to align process-related information (*process information* for short) with business processes and their process tasks; i.e., to bridge the gap between enterprise (process) information and business processes (cf. Figure 1.1). In this thesis, process information refers to data that has been pre-processed to support process participants in the execution of business processes. Hence, process information has a meaning and value for the process participants' business processes and process tasks. Usually, process information is recorded, e.g., in text documents, spreadsheets, presentations, drawings, and e-mails [2, 35].

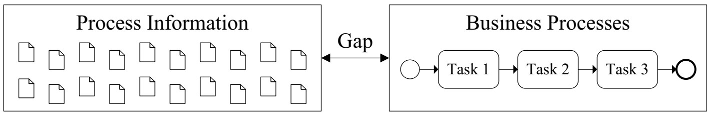

Figure 1.1: Gap between process information and business processes.

In practice, process information is not only stored in large, distributed and heterogeneous data sources, but also managed separately from business processes [2]. For example, shared drives, databases, enterprise portals, content management systems, and enterprise information systems are used to manage process information. In turn, business processes are managed using *process management technology* [36] and *process-aware information systems* (PAIS) [21, 37, 38] respectively. Hence, in practice, process information (cf. Figure 1.2A) and business processes (cf. Figure 1.2B) are often manually and statically linked, e.g., in enterprise process portals connecting specific process information with business processes and associated process tasks [39].

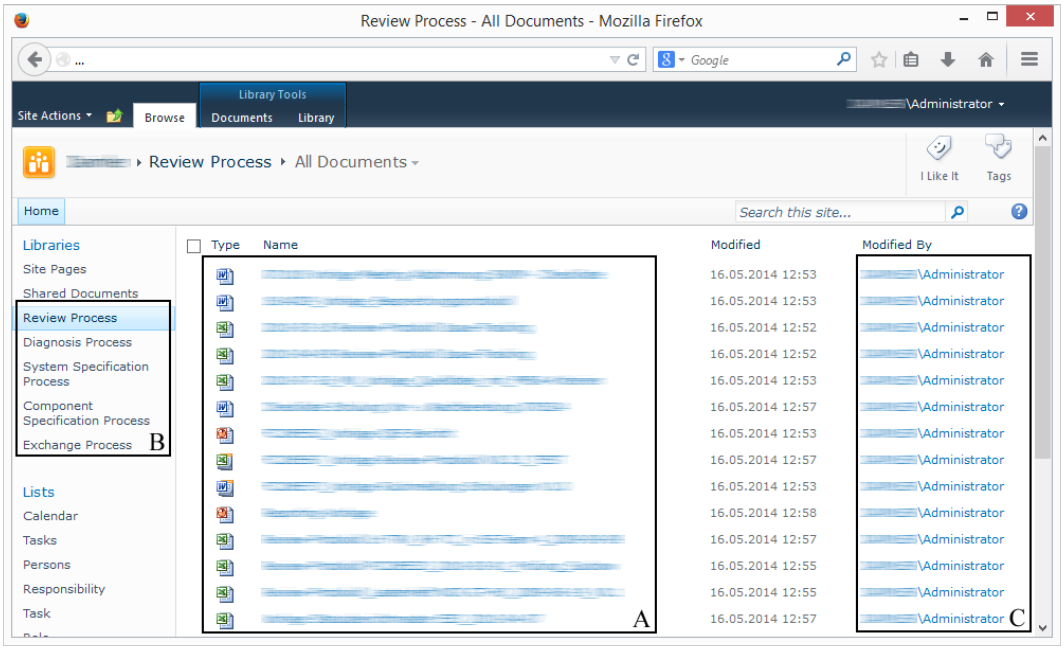

Figure 1.2: Portal manually linking process information with business processes.

In general, such a static approach fails due to high maintenance efforts and costs [2]. In particular, linked process information and business processes must be manually maintained by dedicated administrators (cf. Figure 1.2C). To tackle this challenge, an approach for automatically aligning process information with business processes as well as their tasks is needed; i.e., an approach enabling both *information-* and *process-awareness.*

Aligning process information with business processes and their tasks is far from being trivial. In practice, business processes comprise hundreds or thousands of tasks as well as large amounts of related process information [39, 40]. To cope with this amount, portals have been established as a central point of access to process information as well as business processes and their tasks. However, traditional enterprise portals usually contain complex and static contents, rather distracting than supporting process participants [41].

Providing traditional portals, therefore, is not sufficient for enterprises since the alignment of process information with business processes is strongly influenced by the work context of process participants as well [4]. For example, in a specific work context only selected parts of a process description might be relevant for a particular process participant. Furthermore, less experienced process participants might need more detailed process descriptions than experienced ones. Hence, in order to effectively provide the needed process information (i.e., the process description), the process participant's context must be taken into account as well; i.e., *context-awareness* must be enabled [42].

To tackle the three problem dimensions mentioned (cf. Figure 1.3), this thesis introduces *process-oriented information logistics* (POIL) as a new paradigm for delivering the right process information, in the right format and quality, at the right place and the right point in time, to the right people. In particular, POIL aims at the process-oriented and context-aware delivery of process information to process participants. The overall goal is to no longer manually link process information to business processes, but to automatically identify and deliver relevant process information to process participants.

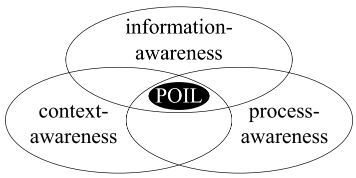

Figure 1.3: Problem dimensions: POIL.

In recent years, various approaches were proposed in the context of the three problem dimensions, including *data warehousing* [43], *business intelligence* [44], *decision support systems* [45], and *enterprise content management* [46]. However, these approaches have not primarily been designed with POIL in mind [3]. Data warehousing, for example, rather focuses on the creation of an integrated database [47]. Opposed to this, POIL deals with the delivery of process information to support the effective and efficient execution of business processes. Traditional business intelligence, in turn, enables data analysis and is usually completely isolated from business process execution [48]. Moreover, information supply is often restricted to decision makers on the management level [49, 50]. Conversely, POIL focuses on the integration and analysis of process information as well as its delivery to both knowledge workers and decision makers. By contrast, decision support systems support decision making; i.e., they serve the management level [51]. Enterprise content management, in turn, deals with the management of information across enterprises referring to strategies, methods and tools [52].

The missing process-awareness in contemporary approaches has guided the development of POIL. The latter aligns process information with business processes, both at the process schema and instance level [53]. Further, it enables a process-oriented and context-aware delivery of process information to knowledge workers. Accordingly, POIL not only combines *information-* and *context-awareness*, but takes *process-awareness* into account as well; i.e., awareness of process schemas and corresponding process instances.

Note that POIL focuses on knowledge-intensive business processes, which are characterized by large amounts of process information, user expertise, user interaction, creativity, and decision making [54]. Examples of knowledge-intensive processes include the engineering of cars in the automotive domain or the treatment of patients in hospitals.

1.2 Research Questions

Developing such an approach, a number of research goals (G1-G5) must be taken into account (cf. Table 1.1). In the following, we explain these goals in detail along an example from the automotive domain; i.e., the review of product requirements.

#	**Research Goals**
G1	Seamless *integration* of process information and business processes.
G2	Intelligent *analysis* of integrated process information and business processes.
G3	Interpretable *representation* of process information and business processes.
G4	Comprehensive *identification* of context information.
G5	Detailed *investigation* how context information can be used.

Table 1.1: Research goals for identifying relevant process information.

First, we have to investigate how process information and business processes can be seamlessly integrated from large, distributed and heterogeneous data sources in order to provide a homogeneous view on them (Goal G1). For example, when considering the process for "reviewing product requirements" in the automotive domain, we must integrate review protocols, review templates, product requirement manuals, and review guidelines. Facing the first goal is a prerequisite to make process information and business processes available to POIL, e.g., to perform the subsequent analysis (see below).

Second, we need to thoroughly analyze integrated process information and business processes to identify syntactic, semantic and conceptual relationships among and between them (Goal G2). This allows us to identify inter-linked process information and business processes. For example, we may have to discover process information having the same author or same tags, dealing with the same topic (e.g., lightweight automotive[1]), or needed for performing a particular process task (e.g., "create a review").

Third, it becomes necessary to investigate how process information, business processes, and their relationships can be represented in a meaningful machine- and user-

[1] *Lightweight automotive* uses techniques that reduce the car's weight to reduce, e.g., fuel consumption.

interpretable form (Goal G3). Note that the results of the integration and analysis phases should be easily usable for the subsequent provision of process information.

Fourth, we need to identify context information influencing the work context of a process participant (Goal G4). Note that we must be able to characterize the specific situation a process participant is involved in. For example, user-related context information (e.g., user name, role, experience level), device-based context information (e.g., display size, bandwidth), or location-based context information (e.g., position, user movement) may be used to characterize a process participant's situation.

Fifth, we need to investigate how context information can be integrated, analyzed and used to enable context-aware process information delivery (Goal G5). This allows us to provide personalized and contextualized process information to process participants.

Finally, we must combine these issues in an integrated and comprehensive approach for enabling the process-oriented and context-aware (i.e., personalized) delivery of process information to knowledge workers and decision makers; i.e., we have to bridge the existing gap between process information and business processes (cf. Figure 1.4).

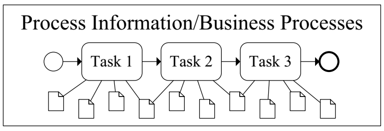

Figure 1.4: Bridging the gap between process information and business processes.

Based on the aforementioned research goals (cf. Table 1.1), we derive a number of research questions (see below), which guide the research described in this thesis. Thereby, we distinguish between *knowledge problems* (KP) and *world problems* (WP); i.e., studying the world and changing it [55, 56]. Knowledge problems address a lack of knowledge about the world. They can be solved by observing others, reading literature, or asking others. Knowledge problems change the state of our knowledge, but not the one of the world [57]. Instead, world problems address the difference between the current and the desired state of the world. They can be solved by changing the world, e.g., changing an organization's structure or implementing a specific concept. Typical engineering problems (e.g., product design) are world problems [56, 58].

In this thesis, five major research questions are addressed:

- **Research Question 1 (KP):** What are existing approaches that may be used to provide relevant process information to process participants?
  - What criteria can be used to compare theses approaches?
  - What are advantages and disadvantages of existing approaches?
- **Research Question 2 (WP):** How can the gap between process information and business processes be bridged?
  - How can process information and business processes be seamlessly integrated?

- How can syntactic, semantic and conceptual relationships between them be determined?
- How can process information and business processes be represented in a meaningful machine- and user-interpretable form?
- How can business processes guide the process-oriented delivery of process information?

- **Research Question 3 (WP):** Which context information is needed to characterize the work context of process participants?
  - How can context information be seamlessly integrated?
  - How can context information be represented in a meaningful machine- and user-interpretable form?
  - How can context information guide the context-aware delivery of process information?

- **Research Question 4 (WP):** How can the relevance of process information for a specific business process and its process tasks be determined?
  - Which criteria influence the relevance of process information?
  - What techniques and methods can be used to determine the relevance of process information?

- **Research Question 5 (WP):** How can process information, business processes, and context information be combined to enable process-oriented and context-aware process information delivery to process participants?

These five research questions guide the research presented in this thesis. Figure 1.5 relates each research question to one or more of the following chapters.

	Part I			Part II				Part III			Part IV	
	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11	Chapter 12
Research Question 1	x	x			x	x	x	x			x	x
Research Question 2			x	x	x			x	x	x	x	x
Research Question 3			x	x		x		x	x	x	x	x
Research Question 4			x	x	x			x	x	x	x	x
Research Question 5				x			x	x	x		x	x

Figure 1.5: Research questions along chapters.

1.3 Contribution

The contributions of this thesis are as follows:

- We identify requirements enabling POIL based on two exploratory case studies in the automotive and clinical domain. Additionally, we present results of an online survey with 219 participants supporting the case study findings.
- We introduce the POIL framework, a comprehensive approach enabling process-oriented and context-aware delivery of process information to process participants.
- We introduce the notion of the *semantic information network* (SIN), a directed, labeled and weighted graph that integrates process information, business processes, and their relationships. In particular, the SIN enables *information-* and *process-awareness* in the POIL framework.
- We introduce techniques and algorithms for maintaining semantic networks (i.e., the SIN). More specifically, we introduce three algorithms for this issue.
- We present a *context model* (CM) for storing and handling context information in a meaningful machine- and user-interpretable form. The context framework introduces *context-awareness* to the POIL framework. Based on the CM, it becomes possible to retrieve process information and business processes related to each other taking the process participant's work context into account.
- We introduce techniques and algorithms for identifying relevant process information. More specifically, we introduce two algorithms for determining the relevance of process information based on their link and rate popularities.
- We present proof-of-concept prototypes implementing the concepts. Further, we report on the results of two validation case studies in the automotive and agricultural domain. Additionally, we present results of performance tests.

1.4 Research Methodology

Our research comprises four major steps: (1) *problem analysis*, (2) *requirements analysis*, (3) *solution design*, and (4) *solution validation* (cf. Figure 1.6). These steps are supported by both empirical (e.g., case studies, online surveys) and non-empirical research activities (e.g., literature surveys, proof-of-concept prototypes, and performance tests).

We start with an analysis of the problem to be investigated (Step 1); i.e., the gap between enterprise (process) information and business processes. We conduct two exploratory case studies and an additional online survey. Based on these activities as well as on practical experiences and a literature survey, we derive the requirements for POIL (Step 2). Addressing the identified requirements, in turn, we create the solution; i.e., the POIL framework (Step 3). Finally, we validate the latter based on several prototypes,

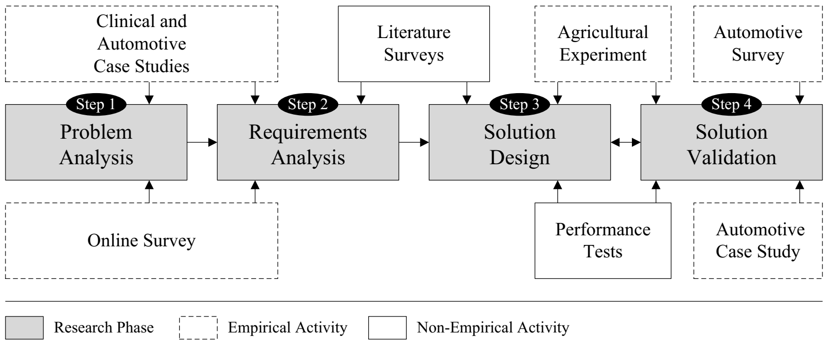

Figure 1.6: Research methodology of the thesis.

a validation case study in the automotive domain, an experiment in the agricultural domain, a survey, and performance tests in the automotive domain (Step 4).

Overall, our research can be characterized by two research paradigms: (1) *behavioral science* and (2) *design science*. Goal of behavioral science is to develop and verify theories that explain or predict human or organizational behavior [59]. An artifact, implemented in an enterprise or organizational context, is often the object of study. The paradigm explains phenomena that occur with respect to the artifact's use, perceived usefulness, and impact on individuals and organizations. Behavioral science consists of two main activities, i.e., *discover* and *justify*. Discovery is the process of generating or proposing claims, whereas justification is the process of testing such claims for validity [60].

In turn, the goal of design science is to create innovative solutions (i.e., artifacts) that serve human purposes and solve real-world problems [60, 61]. Such artifacts are broadly defined by March and Smith as constructs, models, methods (e.g., algorithms), and instantiations (e.g., proof-of-concept prototypes) [60]. Basic activities in design science are *create* and *evaluate* [59]. These activities constitute the counterparts to the discover and justify activities from behavioral science (see above) [60]. Creating is the process of constructing a new artifact for a specific purpose, whereas evaluation is the process of determining how well the artifact suits this purpose. Both paradigms are fundamental to the *information systems* (IS) discipline [59]. Thus, IS research can make significant contributions by engaging their complementary use [59, 60].

In this thesis, we mainly use the research principles of design science, since the goal of this thesis is to provide a solution for a real-world problem by creating innovative artifacts. In order to ensure effective design science research, we follow the research framework suggested by Hevner et al. [59] (cf. Figure 1.7). This framework comprises a set of seven guidelines for conducting and evaluating design science research.

We give a brief description of each of the seven guidelines and discuss how this thesis addresses each guideline of the research framework:

- **Design as an Artifact:** Design science research has to produce an innovative artifact. The latter can be a construct, a model, a method, or an instantiation [60].

The artifact of this thesis is the POIL framework. Note that POIL comprises sub-artifacts such as the SIN, CM and SIN Facade.

- **Problem Relevance:** The goal of design science research is to develop solutions to solve real-world problems. *The problem addressed by this thesis is the gap between enterprise (process) information and business processes.*

- **Design Evaluation:** The utility, quality and efficiency of an artifact must be demonstrated by evaluation methods. *In this thesis, several methods such as case studies, experiments, proof-of-concept prototypes, and use cases are applied in order to evaluate the POIL framework and its sub-artifacts.*

- **Research Contributions:** This guideline requires that effective design science research must provide contributions in the areas of the design artifact, design foundations, and/or design methodologies. *In this thesis, the main contribution is the artifact itself, i.e., the POIL framework. Further contributions are sub-artifacts such as the SIN and CM.*

- **Research Rigor:** Design science research relies upon the application of rigor methods. *In this thesis, we apply previous work from the research fields of information management (IM) and BPM. In addition, rigorous mathematical techniques, such as graph theory or linear algebra, are used. Finally, we apply information retrieval (IR) techniques as the basis for our algorithms.*

- **Design as a Search Process:** To search for an effective artifact requires utilizing available means to reach desired ends. *Implementation and iteration are central issues in this thesis. We study proof-of-concept prototypes that instantiate posed or newly learned design prescriptions from our research.*

- **Communication of Research:** Design science research must be presented both to technology- and management-oriented audiences. *This thesis provides comprehensive information related to technical as well as managerial issues. We introduce technical implementations but also risks and benefits when applying POIL.*

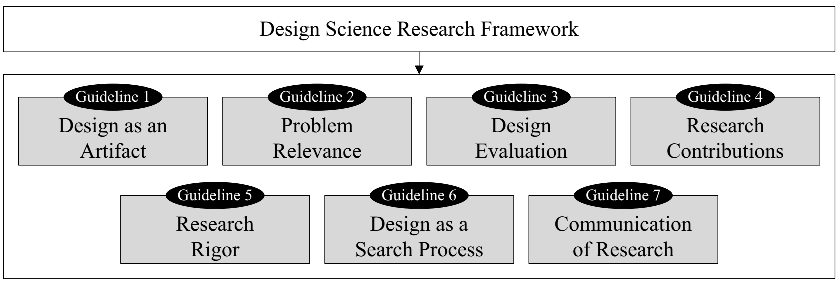

Figure 1.7: Guidelines of the research framework.

In summary, the research framework suggested by Hevner et al. helps us to conduct design science research in an effective and efficient manner [59].

1.5 Outline

This thesis is organized as follows (cf. Figure 1.8). Part I summarizes introductory chapters. Chapter 2 discusses related work. Chapter 3 presents a detailed requirements analysis, which is based on two case studies, an online survey, and a literature survey. Goal is to identify fundamental requirements regarding the POIL framework.

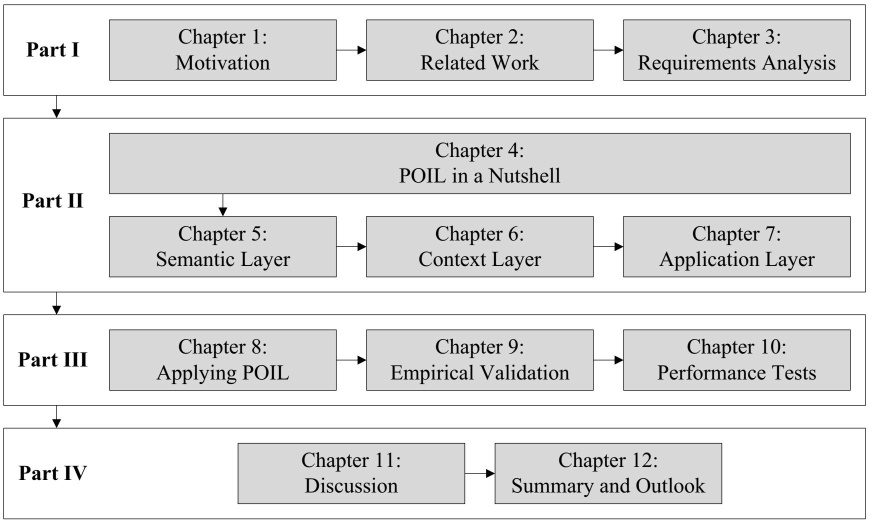

Figure 1.8: Outline of the thesis.

Part II introduces the POIL framework. Chapter 4 sketches POIL and its architectural layers. The latter are addressed in detail in the following chapters. Chapter 5 introduces the semantic layer, which is responsible for the integration and analysis of process information and business processes. Then, Chapter 6 presents the context layer, which integrates and analyzes context information. Finally, Chapter 7 introduces the application layer, which enables the process-oriented and context-aware delivery of relevant process information to process participants.

Part III of the thesis validates the POIL framework. Chapter 8 introduces proof-of-concept prototypes to demonstrate the applicability of POIL in real-world scenarios. Chapter 9 summarizes the results from a survey and a case study in the automotive domain as well as an experiment in the agricultural domain, which demonstrate the feasibility and benefits of POIL. Chapter 10 then presents performance study results of a SIN created in the automotive domain.

Finally, Part IV summarizes the main contributions of the thesis. Chapter 11 discusses POIL, whereas Chapter 12 summarizes the thesis and gives an outlook on future work.

2 Related Work

This chapter discusses related work and provides background information for understanding the thesis. Sections 2.1-2.3 introduce the notions of *information-*, *context-* and *process-awareness*. Then, Section 2.4 presents a literature survey investigating the research field of *information logistics* (IL). Finally, Section 2.5 summarizes findings.

2.1 Information-Awareness

In practice, process information is stored in large, distributed and heterogeneous data sources [62]. Among others, shared drives, databases, enterprise portals, and enterprise information systems are used to manage process information [63]. In this context, many problems arise. Examples include heterogeneous file formats, varying granularity levels of process information [64], and cross-departmental exchange of distributed, heterogeneous process information [1]. Therefore, both the management and handling of process information are far from being trivial. In order to tackle these challenges, i.e., to deliver the right process information, in the right format and quality, to the right process participant, *information-awareness* constitutes a fundamental goal (cf. Figure 2.1).

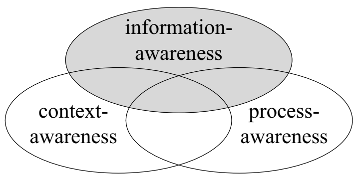

Figure 2.1: Problem dimension 1: information-awareness.

Information-awareness deals with the integration, analysis and delivery of enterprise (process) information. In addition, information quality[1] and information flows[2] within and across enterprises play an essential role for information-awareness [48, 67].

Regarding information-awareness, definitions of *data*, *information* and *knowledge* come into focus [31]. Respective definitions can be derived using the *data-information-knowledge-wisdom* (DIKW) hierarchy (cf. Figure 2.2). Ackoff is often cited as the initiator of the latter [68]. However, an article published earlier by Zeleny in 1987 also discusses the DIKW hierarchy [69]. The latter, which is often referred to as "knowledge

[1] There is no commonly accepted understanding of the term *information quality* apart from very broad characterizations such as "fitness for use" [65]. In this thesis, we define information quality as a set of quality dimensions such as completeness, consistency or topicality [66].

[2] *Information flows* addressing the transfer of information from a particular location to another one; i.e., from a data source to a process participant or from a process participant to another process participant.

hierarchy", "information hierarchy", "knowledge pyramid", or "wisdom hierarchy", constitutes one of the fundamental models in the information and knowledge literature [70]. It has been adopted many times (e.g., by North [71]) and is often quoted and used in definitions of data, information, knowledge, understanding, and wisdom [31, 72].

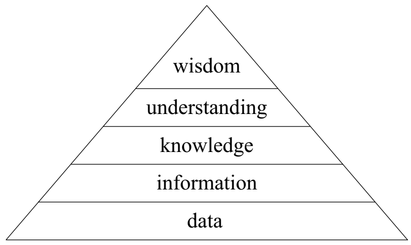

Figure 2.2: Entities of the DIKW hierarchy.

According to Ackoff, the DIKW hierarchy comprises five layers: data, information, knowledge, understanding, and wisdom [68]. The implicit assumption of the DIKW hierarchy is that an entity at a lower level (e.g., information) may be used to create an entity at a higher level (e.g., knowledge). Ackoff stated that each entity at a higher level includes the entities that fall below it [68]. Unlike Ackoff, Zeleny proposes a classification into data, information, knowledge, wisdom, and enlightenment [69]. Enlightenment is considered as an additional step beyond wisdom and reaching the sense of truth; i.e., "the sense of right and wrong, and having it socially accepted and respected" [69]. Zeleny confirmed that each higher level is meant to subsume the lower one [69].

Other authors picked up the DIKW hierarchy and extended it. For example, Chaffey and Wood added additional axes of meaning and value [73]. As assumption, an entity at a higher level (e.g., knowledge, wisdom) has a higher meaning and value than an entity at a lower level (e.g., data, information). Awad and Ghaziri, in turn, evaluated the entities with respect to algorithmic and programmability [74]. In this thesis, we define *data*, *information* and *knowledge* according to [30, 74, 75] as follows:

Definition 1 (**Data**). *Data* are raw facts or observations of things, events, activities, and transactions that are recorded and stored, but are not organized and processed. Therefore, they do not convey any specific meaning.

Definition 2 (**Information**). *Information* refers to data that has been organized and pre-processed for a specific purpose. Consequently, respective information has a meaning and provides some value to the recipient.

Definition 3 (**Knowledge**). *Knowledge* consists of the combination of data and information that have been organized and processed to convey understanding, experience, accumulated learning, and expertise as they applied to a current problem or task.

In other words, data turns into information if someone is interested in it; i.e., data is used or accessed in a specific context [1]. For example, picking up an example from the clinical domain, a doctor might be interested in the blood group of a patient or in the patient's maximum and minimum body temperature during a day. Thus, data is used in context and therefore turns into information. Besides, information can be derived from data as well. As example consider the average body temperature that can be calculated from a number of individual temperature data items (e.g., 36.8°C, 37.1°C). Consequently, the difference between data and information is *not structural*, but *functional* [68]. Information becomes knowledge, if the doctor figured out that patients with a certain blood group, a body temperature higher than a specific value, and at a certain age are more susceptible to specific infections or diseases.

As a consequence, doing the right thing, especially in enterprises and organizations, requires not only knowing *how*, but also knowing *why*. Many process participants know *what* to do, quite a few knowledgeable process experts know *how* to do it, but only a few wise people know *why* it should (or should not) be done [69]. Therefore, the following metaphor applies: data = *know-nothing*, information = *know-what*, knowledge = *know-how*, and wisdom = *know-why* [69, 76]. Based on the definitions of data, information and knowledge, we define *process information* as follows:

Definition 4 (Process Information). *Process information* refers to data that has been pre-processed to support process participants in the execution of business processes. Hence, *process information* has a meaning and value for the process participants' business processes and associated process tasks.

In summary, *information-awareness* enables the pre-processing, management and handling of process information such that it can be delivered to knowledge workers and decision makers [3]. However, only considering information-awareness is not sufficient to realize the POIL framework; i.e., to provide the right process information to the right process participant in a process-oriented and context-aware manner.

2.2 Context-Awareness

In practice, the delivery of process information is strongly influenced by the work context of process participants [4]. As example consider a patient record in a hospital: When having a specific process role (e.g., doctor, nurse), only selected parts of a patient record might be relevant for a process participant. A doctor, for example, needs more detailed information about diseases than a nurse. In turn, a nurse needs more detailed information about the care and treatment of a patient. Hence, in order to effectively provide the needed process information, the process participant's work context must be taken into account as well; i.e., *context-awareness* must be provided (cf. Figure 2.3) [42, 77].

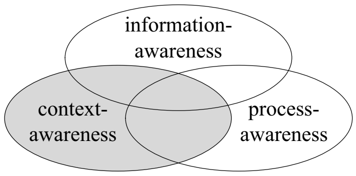

Figure 2.3: Problem dimension 2: context-awareness.

Context-awareness[3] deals with the integration, analysis and handling of context information; i.e., the characterization of a process participant's situation [4]. Besides process-related context information (e.g., temporal process constraints, milestones), user-related context information (e.g., user name, experience level), device-related context information (e.g., display size, bandwidth), location-based context information (e.g., position), time-based context information (e.g., current date), and environment-related context information (e.g., temperature, humidity) may be considered as well [84]. Based on these considerations, we define *context information* as follows:

Definition 5 (Context Information). *Context information* refers to data that has been gathered to determine the work context of process participants. It has a meaning and a value with respect to the process participant's situation.

In general, context-awareness comprises three basic aspects: (1) *sensor*, (2) *context*, and (3) *situation* (cf. Figure 2.4) [78]. Consider the clinical domain and assume that a doctor communicates with a patient. First, sensor data is provided by different sensors (e.g., keyboard input, *global positioning system* (GPS) modules). After that, sensor data is analyzed and harmonized to obtain respective context information. Based on the latter, the context can be identified. For example, process-related context information (e.g., process task: "communicates with patient"), user-related context information (e.g., first name: "Peter", last name: "Miller", role: "doctor"), time-based context information (e.g., day of the week: "Tuesday", day: "13th", month: "May", year: "2014"), location-based information (e.g., room: "A301"), and device-related context information (e.g., used device: "tablet computer") may be utilized. Thus, the situation at hand could be described as follows: "Doctor Peter Miller communicates with a patient on Tuesday, May 13th, 2014, in room A301 using a tablet computer" [4].

The term *sensor* is specified by Haseloff as "any hardware or software system that provides data about the entire or a part of the context of one or more entities" [42]. According to Indulska and Sutton, sensors may be classified into three categories [85]: *physical sensors* (e.g., thermometer, microphone), *virtual sensors* (e.g., keyboard input, touch display movement), and *combinatorial sensors* (e.g., detect a process participant's position by analyzing logins at devices and a mapping of devices to locations). The main task of a sensor is to provide sensor data representing the initial value of context

[3] *Context-awareness* has become somewhat synonymous with other terms, e.g., adaptive [78], reactive [79], responsive [80], situated [81], sensitive [82], or environment-directed [83].

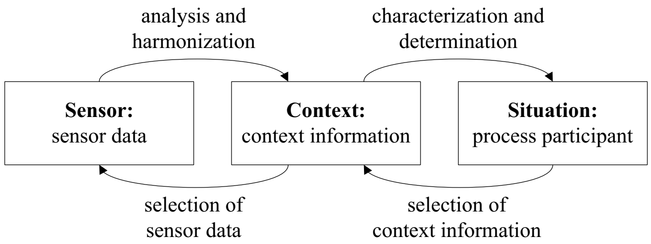

Figure 2.4: Interplay of sensor, context and situation.

information (e.g., the lightning sensor identifies the value of the context information lightning). In this thesis, we define the term *sensor* according to [42, 85] as follows:

Definition 6 (Sensor). A *sensor* is any hardware or software system that provides sensor data about the work context of one or more process participants. A *sensor* can be either a physical, virtual or combinatorial sensor.

Depending on the process participant's situation various context information may be relevant [86]. For example, to be able to update a patient record file, it is not important to know where the process task is performed. Conversely (e.g., in a case of an emergency), it is important to know which doctor is closest to the emergency department. Hence, we define *context* in a more general way according to [87, 88] as follows:

Definition 7 (Context). A *context* is any information that may be used to characterize the situation of an entity. The latter may be a person, location or entity being relevant for the delivery of context-aware process information to process participants.

Finally, the term *situation* can be characterized as "the world state at an instant of time" [77, 89]. Haseloff states that "a situation is a part of the world state at a specific point in time or within a specific time interval" [42]. In other words, a situation represents the instantiation of the context at an instant of time. However, to describe the situation of a process participant we do not need the entire world state, but only that parts which might be relevant for POIL. A *situation* can thus be defined as follows:

Definition 8 (Situation). A *situation* is a part of the world state relevant for the delivery of process information to process participants. It represents the real-world situation of one or more process participants at a certain point in time.

Context-awareness in POIL aims at the delivery of contextualized process information to process participants. This is achieved by the inclusion of context information. According to Dey and Abowd, two definitions of context-awareness can be distinguished [88]:

(1) the more specific case *adapting to context* and (2) the more general case *using context*. The former requires that applications (e.g., enterprise portals) should be able to change or adapt their behavior based on context information. The latter does not require this adaption and classifies applications as context-aware if they are able to display context information (e.g., the local temperature based on GPS). In this thesis, an application is considered as context-aware if it uses context information to deliver *relevant* process information to users. Thereby, relevancy depends on the user's task.

However, enabling both *information-* and *context-awareness* is still not sufficient to support process participants when performing business processes. Additionally, process-awareness is required to enable a process-oriented delivery of process information [3].

2.3 Process-Awareness

In practice, business processes and their tasks are often managed based on process management technology [36] and *process-aware information systems* (PAIS) [21, 37]. Business processes may be characterized by hundreds or thousands of process tasks, numerous process variants[4] [90, 91], and large amounts of process information. Regarding business processes, we must additionally distinguish between process schemas and enacted process instances [53]. Reason is that specific process instances require different process information. Consider a ward round in a hospital: A doctor needs different patient records depending on the patient being treated. Hence, in order to effectively provide the needed process information, *process-awareness* must be enabled (cf. Figure 2.5) [3].

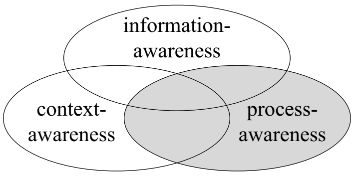

Figure 2.5: Problem dimension 3: process-awareness.

Process-awareness deals with the integration, analysis and delivery of business processes and their associated tasks. Goal is to guide the process-oriented delivery of process information to knowledge workers and decision makers. According to [21, 36, 92], this thesis defines *business processes* and their *tasks* as follows:

Definition 9 (Business Process). A *business process* is a set of one or more connected tasks which collectively realize a particular business goal. Usually, a *business process* is linked to an organizational structure defining functional and organizational relationships. Further, a *business process* may involve one or more departments.

[4] A *process variant* is configured from a base process by applying different change operations using rules.

Definition 10 (Task). A *task* is a logical and atomic unit of work being performed in the context of a business process.

Investigating business processes and tasks, PAIS come into focus [21, 36, 92]. PAIS are systems that "manage and execute operational processes involving people, applications, and/or information sources on the basis of process models" [93]. PAIS refer to systems that manage and execute business processes and their tasks. In particular, most PAIS describe a business process in terms of a *process model* providing the schema for business process execution [21]. An example of a process model is given in Figure 4.2. After creating an executable process model, it can be deployed to a PAIS which then allows creating, executing and managing *process instances* [21]. The latter is crucial to guide a process-oriented delivery of process information to process participants.

Within POIL, knowledge-intensive business processes are focused because they involve large amounts of process information, user expertise, and decision making. Recently, several attempts have been made to define knowledge-intensive business processes [54]. For example, Davenport et al. stated that knowledge-intensive business processes are characterized by the diversity and uncertainty of process input and output [72, 94]. In turn, Eppler et al. mentioned that knowledge significantly contributes to the values within a business process [95]. Mundbrod et al. stated that knowledge-intensive processes and their tasks are solely driven by professionals utilizing their skills and expertise whereas no support is provided by PAIS [25, 96]. Künzle et al. argue that knowledge-intensive processes require a tight integration of the process and data perspective; i.e., both process- and object-awareness need to be considered [97, 98]. Finally, Gronau et al. stated that knowledge-intensive processes are only partially mapped by a process model due to unpredictable decisions or tasks guided by creativity [54]. Based on these considerations, according to [54, 95], a *knowledge-intensive business process* is defined as follows:

Definition 11 (Knowledge-intensive Business Process). A *knowledge-intensive business process* comprises large amounts of process information, user expertise, user interaction, creativity, and decision making. *Knowledge-intensive business processes* are typically not or only partly automated (e.g., by process management technology).

2.4 Information Logistics

An approach enabling information- and context-awareness is *information logistics* (IL) (cf. Research Question 1). Traditional IL approaches deal with the question of how to deliver information to knowledge workers and decision makers as effectively and efficiently as possible [99]. For this purpose, basic principles from the fields of material logistics and lean management are applied [100]. Examples include just-in-time delivery of goods and satisfaction of customer needs [101, 102]. Particularly, IL aims at delivering that information to knowledge workers and decision makers fitting their needs best [4].

Consequently, *information-awareness* (e.g., awareness of information quality) and, to a smaller extent, *context-awareness* (e.g., awareness of the user context[5] for which personalized information shall be delivered) adopt key roles in IL (cf. Figure 2.6) [8].

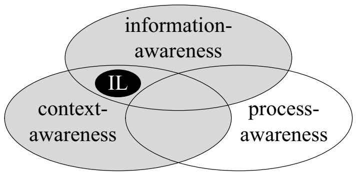

Figure 2.6: Problem dimensions: IL.

Although IL is independent from the use of ICT, the latter has been intensively used as an IL enabler for several years. As example consider ICT solutions in areas like data warehousing, business intelligence, management information systems, decision support systems, and enterprise content management. However, respective solutions also suffer from shortcomings, like their restricted applicability (e.g., only applicable within enterprises and not across them) [103], missing operational functionality (e.g., only serving the management level) [104], and lack of *process-awareness* (e.g., delivering information without considering the current process context of process participants). In this thesis, *information logistics* is defined according to [48] as follows:

Definition 12 (Information Logistics). The approach of *information logistics* (IL) focuses on the planning, execution and control of information flows. The main tasks of *information logistics* include the integration, analysis and delivery of information to individuals by taking available context information into account.

To better understand past, current and future IL developments, we present a literature survey in this section[6] in the research field of IL. This survey not only reveals recent developments in IL, but also discusses related work important for the thesis. Our survey has been guided by two research questions as depicted in Table 2.1.

#	**Research Questions**
RQ1	What is the state-of-the-art in the research field of IL?
RQ2	What are current research trends in the research field of IL?

Table 2.1: Research questions underlying the literature survey.

[5]The *user context* is a subset of the process participant's work context. It includes user-related context information such as user name, experience level, or personal settings.

[6]The section is based on the following referred paper:
[8] B. Michelberger, R. Andris, H. Girit, and B. Mutschler. *A Literature Survey on Information Logistics.* In: Proc 16th Int'l Conf on Business Information Systems (BIS'13), LNBIP 157, pp. 138–150, Poznań, Poland, 2013.

To answer these questions, we analyzed 53 IL-related articles (cf. Section 2.4.2) and classified them in ten research clusters (cf. Section 2.4.3). Note that the approach of POIL is one of the identified research clusters (i.e., the 3rd one) in the literature survey.

2.4.1 Research Methodology

In order to ensure the validity of the survey, we used survey protocol documents as proposed in the literature survey guide by Okoli and Schabram [105]. Our survey comprises four steps (cf. Figure 2.7): (1) *search*, (2) *selection*, (3) *analysis*, and (4) *classification*.

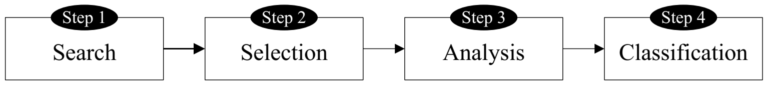

Figure 2.7: Steps of the literature survey.

Step 1: Search. First, a profound web-based search was conducted to identify potentially relevant IL articles. We considered an article as relevant based upon two selection criteria: (1) an article contains the term "information logistics" in its title and (2) the article was written in English. Specifically, we used Google Scholar, SpringerLink, the ACM Digital Library, the IEEE Xplore Digital Library, ScienceDirect, and Microsoft AS. We considered articles from books, journals as well as conference and workshop proceedings. We also took reports, editorials and PhD theses into account. Other kinds of articles, such as commercial white papers, were not considered in the literature survey.

Step 2: Selection. This step reassessed the number of articles identified in Step 1. In particular, we removed irrelevant articles (e.g., an article with the title "Information, Logistics and Retailing Services") as well as duplicates.[7] Then, we identified and selected analyzable articles. We considered an article as analyzable if its full text was available. Finally, we enriched all remaining articles with metadata such as citation count, type of publication, or year of publication. This enabled a more in-depth analysis (cf. Step 3) and supported the subsequent clustering (cf. Step 4). In total, we had a list of 63 articles potentially relevant at the end of Step 2.

Step 3: Analysis. In the third step, we performed an in-depth content analysis of the 63 articles. Therefore, all 63 articles were reviewed by at least two reviewers according to the procedures suggested by Okoli and Schabram [105].[8] Among other things, a separate review was created for each article. Note that based on the reviews ten articles were excluded from the survey due to quality issues or other reasons. For example, some articles did not meet our content requirements, consisted only of a few sentences, or were literature surveys similar to ours (e.g., by Haftor et al. [106]).

[7] Note that some articles have been found by several search engines.
[8] Note that most articles have been reviewed by three or four reviewers.

Step 4: Classification. Based on the remaining 53 articles, the generated metadata, and the created reviews, we performed the clustering. Thereby, for example, we also took topics, authors and institutional relationships into account. Finally, we organized 53 articles in ten research clusters (cf. Figure 2.10).

Note that our survey has several limitations. First, we only considered articles with "information logistics" in their title. This limitation was made due to the large amount of search engine hits (4,400 hits) we obtained when we considered papers with the term "information logistics" in their full text. Second, only articles in English were considered.

2.4.2 Data Collection

Altogether, our initial web-based search resulted in 282 hits; i.e., 282 articles potentially being relevant for our survey. Google Scholar delivered most hits (139 hits), followed by Microsoft AS (94 hits) and the IEEE Xplore Digital Library (20 hits). Less results have been identified based on the ACM Digital Library (13 hits), SpringerLink (13 hits), and ScienceDirect (3 hits). Table 2.2 summarizes the raw results of our survey.

Search Engine	Total Hits	Irrelevant Hits	Relevant Hits	Precision
Google Scholar	139	62	77	55.4%
SpringerLink	13	1	12	92.3%
ACM Library	13	0	13	100.0%
IEEE Library	20	9	11	55.0%
ScienceDirect	3	0	3	100.0%
Microsoft AS	94	53	41	43.6%

Table 2.2: Raw results of our literature survey.

In Step 2, we identified articles that did not meet our selection criteria. As a result, we excluded 125 articles from the survey; i.e., 157 articles remained, implying an aggregated precision across all search engines of 55.7%. Out of these 157 articles, we removed duplicate articles and excluded articles we could not analyze due to a missing full text. At the end of this step, 63 articles were selected for further analysis.

Before starting our in-depth content analysis of the articles (i.e., Step 3), each of the 63 articles was associated with additional metadata. Among others, the year of publication, the citation count (according to Google Scholar), and the type of publication were documented. This enabled us to look for time-based trends and developments.

Figure 2.8 shows, for example, that the number of IL-related articles has significantly increased in recent years (i.e., when considering the period 1989 to 2012).

Figure 2.8 further illustrates the type of the considered articles. Most articles (44 ones) stem from workshop or conference proceedings, followed by journals (9 ones), technical reports (5 ones), PhD theses (2 ones), book chapters (2 ones), and editorials (1 article).

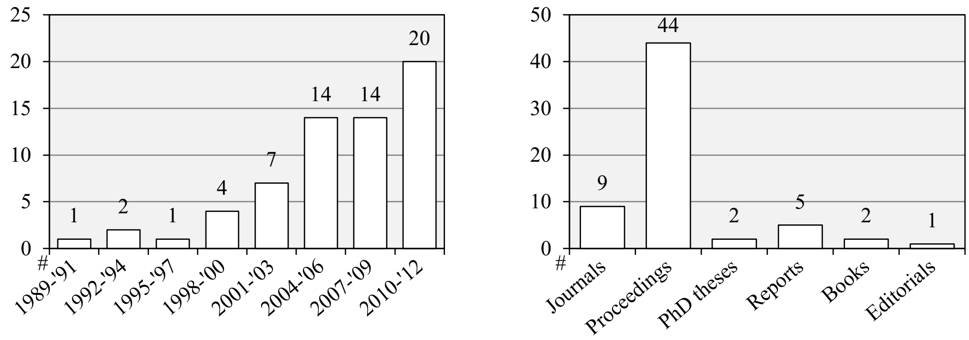

Figure 2.8: Publication date and type of analyzed articles.

Figure 2.9 illustrates the citation count of the articles. Most articles (21 ones) are not cited. 14 articles have 1-2 citations, 10 articles have 11-20 citations, 7 articles have 3-5 citations, and another 7 articles have 6-10 citations. The most three cited articles are Meissen et al. [77] (56 citations), Bucher and Dinter [48] (27 citations), and Deiters et al. [101] (32 citations) according to Google Scholar in the year of our survey.

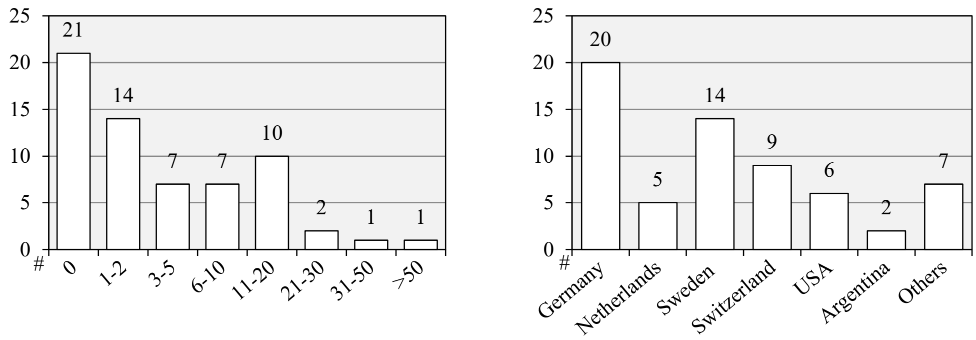

Figure 2.9: Citation count and country of origin of analyzed articles.

Figure 2.9 illustrates the country of origin as well. Most articles (20 articles) stem from Germany, followed by Sweden (14 articles), Switzerland (9 articles), USA (6 articles), The Netherlands (5 articles), and Argentina (2 articles). In addition, there are 7 articles from other countries (e.g., France, Austria).

In Step 3, the 63 articles were carefully reviewed by at least two reviewers. For each article, a review containing a short summary, the full abstract, and key words were created. As aforementioned, we excluded ten further articles from the survey as a result of the reviews due to quality issues or other reasons. Thus, 53 articles were finally included in the literature survey (and are therefore the basis for identifying clusters).

2.4.3 Research Clusters

This section describes the ten IL research clusters (C1-C10) we identified based on our literature survey (cf. Figure 2.10). Table 2.3 additionally shows the most cited article of each cluster. Table 2.4 summarizes the main characteristics of each cluster. Note that there are overlaps between the clusters (also meaning that several of the identified IL articles could be assigned to more than one cluster).[9]

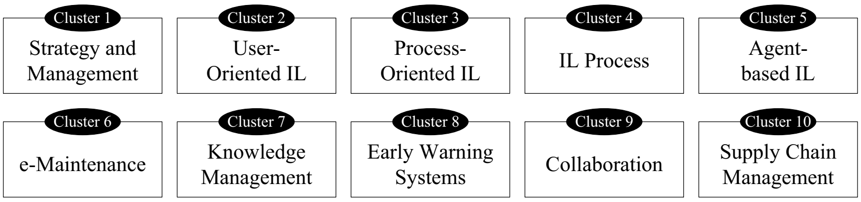

Figure 2.10: Identified research clusters.

Cluster 1 (C1): Strategy and Management. Articles belonging to this cluster concern strategy and management issues related to IL, in particular, the transformation of enterprises into IL organizations. Dinter and Winter [103], for example, discuss the state of IL strategies. As main finding, IL strategy depends on company size and structure. In addition, Dinter [107] investigates critical success factors for IL strategies. Examples include comprehensiveness, flexibility, support, communication, *information technology* (IT) strategy orientation, and project collaboration. Special focus of Klein [108] are IL management tasks enabling the use of IL concepts within an organization. Winter [104] discusses general and thus very broad IL management challenges. More specific conceptual models to better understand IL requirements in enterprises are presented in [109] and [110]. A study assessing the IL landscape of a global automotive company is presented in [111]. Another study assessing benefits, design factors, and realization approaches in IL is presented in Bucher and Dinter [48]. Finally, Bucher and Teich [112] present a case study on the design and implementation of IL in the healthcare domain.

Cluster 2 (C2): User-Oriented IL. The articles in this cluster address the challenges in user-oriented IL. In [113], Sandkuhl discusses challenges and solutions for user-oriented information supply in IL. According to Deiters et al. [101], IL can be understood as an approach enabling just-in-time delivery of information to users. Corresponding examples are given in the fields of wearable computing [114], weather forecast [115], and the healthcare domain [99]. Böhringer and Gaedke [116] argue that the success of information supply depends on successful user adoption and powerful front end technologies. In [116], therefore, a Twitter-like front end for IL is presented. Moreover, Sandkuhl et al. [117] present intelligent IL services and discuss integration challenges. In [118], an industrial case study on these IL services is presented. A similar, but more technical perspective on integration challenges is addressed in [119].

Besides these considerations, context-awareness adopts a key role in user-oriented IL. Levashova et al. [120], for example, present a study on context-based models for IL. Context definitions and representations from different perspectives (e.g., information

[9]The problem that IL articles could be assigned to more than one cluster is discussed in Section 2.4.4.

demand analysis, decision support) are presented by Lundqvist et al. [121]. A reference architecture for context-awareness in IL applications is presented by Haseloff [42]. Another context framework for IL, considering various situations, is presented by Meissen et al. [77]. This context framework has been tested by the authors in [122] using an automotive prototype to demonstrate its general applicability.

Cluster 3 (C3): Process-Oriented IL. This cluster deals with the alignment of process-related information (e.g., working instructions, best practices) with knowledge-intensive business processes. This should enable knowledge workers and decision makers to perform their tasks in the best possible way [3]. Specifically, process-oriented IL enables process-oriented and context-aware delivery of relevant information to knowledge workers and decision makers. For this task a *semantic information network* (SIN) is used, which integrates process information (i.e., information objects), business processes (i.e., process objects), as well as their relationships. In [1], quality dimensions of process-related information (e.g., completeness, punctuality) are investigated in order to determine the relevance of information along business processes. In [4], an ontology-based context framework for process-oriented IL is proposed. This framework aims at the context-aware delivery of process-related information to process participants.

Cluster 4 (C4): IL Process. In [123], IL is introduced as an approach (or process) transforming a given input (e.g., a project description, lessons learned) into some form of output (e.g., a best practice document). Goal is to transform fragmented information into usable information for the receiver. An IL transformation comprises three phases: information supply, information production, and information distribution. In order to realize this IL approach, Apelkrans and Håkansson [124] suggest an agent-based IL approach (i.e., the combination of multi-agent systems and IL). In [125], the notion of IL and basic ingredients of the IL process are discussed. Finally, in [126], the authors present a visual knowledge modeling approach of an IL process as defined in [125].

Cluster 5 (C5): Agent-based IL. This cluster concerns agent-based IL. In this context, an agent is a piece of software that acts for a user when searching for needed information. Knublauch and Rose [127], for example, argue that a multi-agent IL approach, providing techniques for autonomous, situated, social, and pro-active information services, is a well-suited approach for realizing IL. A different perspective is adopted by Timm et al. [128]. The authors discuss the use of adaptive multi-agents approaches. Winkler et al. [129] present an agent-based IL architecture for process management; i.e., to support processes which rely on informational inputs and produce information as an output. Finally, Bodendorf et al. [130] present an agent-based IL approach for the monitoring and coordination of processes.

Cluster 6 (C6): e-Maintenance. The articles in this cluster concern IL in the context of e-Maintenance. One maintenance problem is to manage system complexity. Some experiences from the aerospace domain are described in [131]. Specific e-Maintenance

IL solutions are discussed by Candell et al. [132]. Moreover, Haftor et al. [133] propose a framework for IL-driven e-Maintenance. In [134], maintenance and ICT are merged from an IL perspective. The role of IL and data warehousing in maintenance management is addressed by Vieira and Cardoso [135].

Cluster 7 (C7): Knowledge Management. The articles in this cluster deal with knowledge processing in and through IL. Czejdo and Baszun [136] and Czejdo et al. [137], for example, discuss an IL approach for knowledge processing. The presented knowledge processing approach aims at increasing the daily performance of knowledge workers in enterprises. Rudzajs and Kirikova [138] propose IL for conceptual correspondence monitoring. Finally, Willems [139] and Willems et al. [140] address the enabling role of IL approaches in *knowledge management* (KM). They conclude that an IL approach significantly improves a knowledge worker's daily performance.

Cluster 8 (C8): Early Warning Systems. Lendholt and Hammitzsch [141, 142] apply the concept of IL to hazard monitoring and early warning systems. Goal is to enable the generation of user-tailored warning messages considering user needs with respect to content, location or individual requirements. In addition, filter mechanisms to avoid information overload in emergency situations are presented.

Cluster 9 (C9): Collaboration. This cluster discusses the importance of IL to support collaboration in enterprises and organizations. In [143] and [144], IL is defined as the maintenance, tracking, monitor, and enactment of information flows within collaborative environments. Scherer [145] argues, in addition, that an IL approach is necessary to cope with the complexity of information flows. Nuntasunti [146] analyzes the information flow between participants of collaborative processes.

Cluster 10 (C10): Supply Chain Management. This cluster deals with IL approaches supporting *supply chain management* (SCM). Vegetti et al. [147] propose the design of an ontology to support IL supply chains. This ontology is described in more detail in [148]. Besides, Timlon and Harryson [149] propose a supply chain strategy to increase supply chain integration through organizational learning regarding IL activities.

Table 2.3 shows the most cited article of each cluster, whereas Table 2.4 summarizes the main characteristics (e.g. first and latest article, trend, foundation) of the clusters.

Criteria	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Article	[48]	[77]	[4]	[124]	[128]	[131]	[140]	[141]	[144]	[148]
Year	2008	2004	2012	2008	2001	2009	2009	2011	2000	2005
Citations	27	56	3	11	13	25	6	7	12	1

Table 2.3: Most cited article in each research cluster.

Criteria	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
First Article	1993	1999	2011	2003	2000	2009	2008	2011	2000	2005
Latest Article	2012	2012	2012	2008	2007	2010	2011	2012	2004	2006
Trend	↗	↗	↗	↓	↓	→	↗	→	↓	↓
Foundation	↑	↑	↘	↗	↗	↑	→	→	↗	↘
Articles	↗	↑	→	→	→	↗	↗	↘	→	→
1989–'91	–	–	–	–	–	–	–	–	–	–
1992–'94	1	–	–	–	–	–	–	–	–	–
1995–'97	–	–	–	–	–	–	–	–	–	–
1998–'00	–	1	–	–	1	–	–	–	2	–
2001–'03	–	4	–	1	1	–	–	–	1	–
2004–'06	–	5	–	1	1	–	–	–	1	3
2007–'09	5	1	–	2	1	3	2	–	–	–
2010–'12	3	3	3	–	–	2	3	2	–	–
Total	**9**	**14**	**3**	**4**	**4**	**5**	**5**	**2**	**4**	**3**

Table 2.4: Articles in the research clusters.

2.4.4 Discussion

The number of IL-related articles, both from researchers and practitioners, has significantly increased in recent years. For example, 20 new articles have been published since 2010. As can be seen, we were able to identify a large number of IL methods, concepts and approaches in the literature survey. As main problem, the broad field of IL makes the comparison of methods, concepts and approaches a big challenge. In fact, the term "information logistics" is the only commonality between many IL articles [133]. Reason is that IL addresses and recombines a number of well-known research areas, e.g., material logistics [101], process management [3], information management [108], ubiquitous computing [42], or semantic technologies [113]. Additionally, ideas from data warehousing [43], business intelligence [44], location-based services [85], knowledge management [74], or enterprise content management [46] are picked up as well.

In our survey, we classified articles along ten clusters (RQ1). However, there are overlaps between these clusters (i.e., several of the identified IL articles could be assigned to more than one cluster). For example, both C2 (i.e., user-oriented IL) and C3 (i.e., process-oriented IL) focus on the delivery of needed information to users. However, while C2 concerns respective requirements and solutions for human users [101], C3 focuses on the support of both business processes and process participants (as articles assigned to C2 neglect business processes and process orientation). Still, topics are similar in C2 and C3. As another example for overlapping clusters consider C4 (i.e., IL processes) and C5 (i.e., agent-based IL). In order to establish IL processes, [124] (assigned to C4) suggests to use an agent-based IL approach, like the one introduced in [130] (assigned to C5). Also consider C3 and C5. In [129], an agent-based IL architecture for process management is given. This work, however, could be assigned to C3 as well. In addition, C7 (i.e., knowledge management) and C10 (i.e., supply chain management) overlap as well. For example, both [136] (from C7) and [147] (from C10) discuss ontologies in the context of IL. Finally, IL-based early warning systems [142] in C8 (i.e., early warning systems) adopt approaches we assigned to C2 (e.g., a weather forecast prototype [115]).

The literature survey considers IL-related articles till the end of 2012. However, in the years 2013 and 2014 further articles were published in the context of IL (RQ2). In the following, the most important developments are discussed. For example, Söderholm and Norrbin [150] describe an IL-based framework for continuous dependability improvement that is applied to improve the operational availability performance of the railway system (C6). Sandkuhl [151], in turn, proposes a pattern-based knowledge architecture for implementing IL in networked organizations (C2). In [152], an ontology for the field of IL services in transportation is proposed (C2). Rudzajs and Kirikova [153] discusses challenges of multimode IL in monitoring conceptual correspondence (C7). In [6] and [10], two applications are presented applying IL in the clinical and automotive domain (C3). Moreover, Meister et al. [154] apply *complex event processing* (CEP) to cope with the problem of information overload in healthcare according to the principles of IL (C2). Ignáczová [155] proposes an IL solution ensuring and managing information flows of an institution for higher education (C2). Moreover, Straka et al. [156] introduce an IL approach which provides information services for freight transport needs (C10). Gaidukovs and Kirikova [157], in turn, take a closer look at the time dimension of IL and suggest the extension of IL-related models with a time dimension model (C2). Finally, Stamer et al. [158] propose a conceptual architecture for enterprises to support demand-oriented information supply; i.e., to provide currently needed e-mails to users (C2).

2.5 Summary

This chapter discussed related work and background information relevant for the thesis. We introduced the notions of information-, context- and process-awareness. Moreover, we considered an approach enabling information- and context-awareness; i.e., *information logistics* (IL). In addition, we presented the results of a profound literature survey in the field of IL. More specifically, the main objective of the survey was to better understand past, current and future developments in IL. In total, we included 53 articles in the survey. These 53 articles have been classified into ten research clusters.

3 Requirements Analysis

This chapter[1] presents requirements for realizing POIL. Section 3.1 provides a short motivation and Section 3.2 describes the research design of the requirements analysis. Then, Sections 3.3-3.5 derive the requirements based on two exploratory case studies in the automotive and clinical domain as well as an online survey with 219 participants. Furthermore, Section 3.6 approves the requirements and identifies further ones by a literature survey. The chapter not only reveals fundamental POIL requirements in Section 3.7, but also allows us to better understand the role of process information when performing business processes. Finally, Section 3.8 summarizes the chapter.

3.1 Motivation

The mere availability of process information is not sufficient to adequately support process participants [159]. Only when considering the process participant's process context[2] it becomes possible to effectively provide context-aware process information [42]. In this context, many problems arise. Examples include heterogeneous file formats, varying granularity levels of process information [64], and cross-departmental exchange of distributed process information [1]. Moreover, think of inconsistencies or an increased communication overhead due to the different structures of digital and paper-based process information; i.e., structured, semi-structured or unstructured information [2, 63].

Overall, our analysis has been guided by four research questions (cf. Table 3.1):

#	**Research Questions**
RQ1	How are business processes and tasks documented?
RQ2	What different kinds of process information are used?
RQ3	How can a process context be determined?
RQ4	How can relevant process information be determined?

Table 3.1: Research questions underlying the requirements analysis.

[1]The chapter is based on the following referred paper:
[2] B. Michelberger, B. Mutschler, and M. Reichert. *On Handling Process Information: Results from Case Studies and a Survey.* In: Proc BPM'11 Workshops, 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333–344, Clermont-Ferrand, France, 2011.

[2]The *process context* is a subset of the process participant's work context. It includes process-related context information such as temporal process constraints or the number of running process instances.

First, we need to know how business processes and tasks are documented to identify the process information needed during process execution (RQ1). Second, we need to know what kinds of process information are used during daily work in order to provide the right process information to the users (RQ2). Third, it is crucial to know what criteria influence the process context and how context information can be utilized to provide context-aware process information (RQ3). Finally, we need to know what criteria influence the relevance of process information and how relevance can be evaluated in order to provide the process information to users fitting their demands best (RQ4).

3.2 Research Design

The requirements analysis comprises three parts (cf. Figure 3.1). First, we performed two qualitative exploratory case studies based on face-to-face interviews and paper-based questionnaires (Part 1). Second, we conducted an online survey supporting the case study findings (Part 2). Third, we conducted a literature survey collecting additional data to identify further requirements on POIL and to confirm existing ones (Part 3).

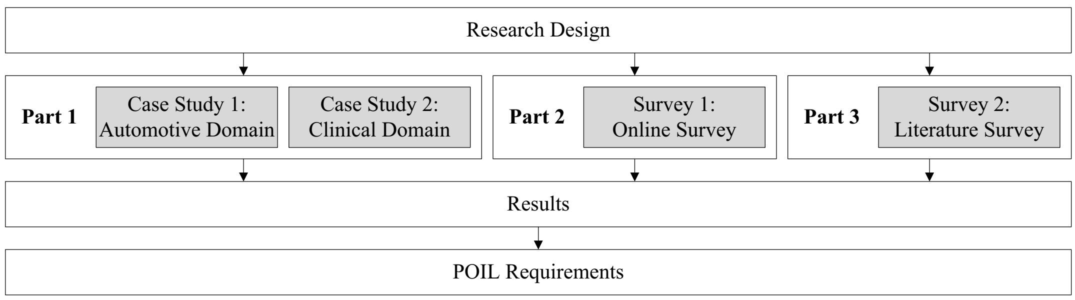

Figure 3.1: Parts of the requirements analysis.

Part 1: Case Studies. Our case study research is of explorative nature. According to Yin, case studies are a research method to answer *why* and *how* research questions [160]. Kitchenham et al. add to this statement that case studies usually investigate what is happening in "typical" project settings, so it is *research-in-the-typical* [161].

In the automotive case study eight persons were interviewed and in the clinical case study nine persons. The interviewees worked in different areas (e.g., development, research, accounting, management). Therefore, both knowledge workers and decision makers were involved. Participants were selected in cooperation with each organization. None of the participants was a member of the research team.

In both case studies, data was gathered through face-to-face interviews following a semi-structured interview guideline. After each interview, an additional questionnaire had to be filled out to collect further data. Each interview lasted about 90 minutes.

The interviews addressed three main topics: (1) *business processes* in which the interviewees participate (RQ1), (2) *types of process information* needed (RQ2), and (3) factors determining the *process context* (RQ3) (cf. Table 3.2).

Part 2: Online Survey. For enabling *research-in-the-large*, i.e., to capture what is happening broadly over "large" groups, surveys can be used [161]. Therefore, we conducted a survey to collect further data that shall help us to generalize the case study results.

The online survey was conducted via a web questionnaire. Overall, 219 users from more than 100 enterprises participated. The survey was available both in English and German language. It was advertised via private contacts, business contacts, mailing lists, and groups in social platforms (e.g., LinkedIn [162], Xing [163]). The questionnaire comprised 23 questions on (1) *demographic issues*, (2) *business process management in general* (RQ1), and (3) *handling of process information* (RQ2 and RQ4) (cf. Table 3.2).

Part 3: Literature Survey. The literature survey was conducted using different search engines (e.g., Google Scholar, Microsoft AS) and digital libraries (e.g., SpringerLink, ScienceDirect) in order to identify relevant literature. On one hand, the goal of our literature survey was to better understand past, current and future developments in IL (cf. Section 2.4). On the other hand, goal was to substantiate previously collected empirical data. The survey addressed two major topics: (1) *handling of process information* (RQ2 and RQ4) and (2) *context-aware delivery of process information* (RQ3) (cf. Table 3.2).

#	Case Studies	Online Survey	Literature Survey
RQ1	×	×	
RQ2	×	×	×
RQ3	×		×
RQ4		×	×

Table 3.2: Research questions along parts.

3.3 Case Study 1: Automotive Domain

We considered selected business processes such as the review of product requirements or the identification of system specifications in the first case study. Participants stemmed from electric/electronic engineering departments, but also from the departments responsible for project management and safety planning. These departments were selected because of the knowledge-intensive business processes they are involved in.

RQ1. The business processes from the involved departments were mainly available in a documented form, specifically, in PowerPoint files. They can be characterized as knowledge-intensive; i.e., their execution requires a large amount of process information. However, business processes were documented from a high-level perspective solely. In order to reach POIL goals, business processes and their elements (e.g., tasks, roles) should be defined in a uniform way (e.g., using a modeling notation such as *business process model and notation* (BPMN) [164] or *event-driven process chain* (EPC) [165]).

Regarding business processes, we must distinguish between process schemas and enacted process instances. Reason is that specific instances require different process information.

Although responsibilities are defined, there are still ambiguities. For example, roles are often unknown or not adequately documented. As another difficulty, processes may cross departmental borders. In this context, a specific challenge is the collaboration between the departments (e.g., each department only looks at its own tasks and usually has no knowledge of the entire business process). Hence, defined business processes can be performed in many different ways. To avoid this issue, responsibilities of roles must be defined and a detailed description of tasks and the entire process must be given.

Based on this, we derive the first requirement from the viewpoint of business processes.

Requirement 1 (R1). POIL should be able to gather business processes from process repositories, transform business processes into uniform structured process objects (on the process element level), and integrate them into a comprehensive approach.

This requirement ensures that business processes are available in POIL and enables *process-awareness*. To answer RQ2, we take a closer look at the location, format and quality of process information and how to deal with these issues.

RQ2. Yin states that research questions are usually too abstract and broad [160]. Therefore, we divide our second research question RQ2 *"What different kinds of process information are used?"* into three sub-research questions as depicted in Table 3.3.

#	Sub-Research Questions
SRQ1	Where is process information located?
SRQ2	What are important file formats/applications during daily work?
SRQ3	How is the quality of available process information?

Table 3.3: Sub-research questions of research question RQ2.

To answer SRQ1, we considered the IT application landscape of the involved departments. In the automotive industry this landscape is extremely complex. There are numerous applications in use that provide the needed process information. In addition to standard applications (e.g., Lotus Notes, Rational DOORS[3]), there exists a large number of individual applications (e.g., enterprise portals, *visual basic for applications* (VBA) macros). Furthermore, process information is available on shared drives, local drives, and the Internet. As a consequence, process information should be easily accessible for process participants from a centered point of access. Finally, not all process information is available in digital form; i.e., some information is only available in paper-based form (e.g., technical drawings, circuit diagrams).

[3] *Rational DOORS* is a requirement management tool.

Participants confirmed that most process information is available in databases, in applications, on the Internet, or on shared drives (cf. Figure 3.2A). Due to the extensive use of shared drives, a revision control system is used by the participants. The file explorer and the Intranet are the most common ways to access process information. However, access to process information is not always possible since not all employees may have the licenses required for using a specific application. Further, errors occur, as information is often printed, manually processed, re-entered, and further processed.

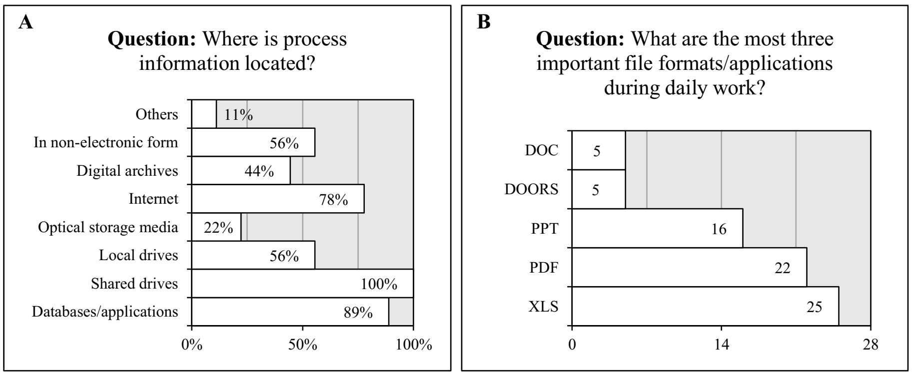

Figure 3.2: Location of process information and important file formats/applications.

To answer SRQ2, we examined used file formats and applications (e.g., Lotus Notes, Rational DOORS). All participants stated that they use Excel, PowerPoint and *portable document format* (PDF) files. Seven out of nine participants stated that paper-based files (e.g., technical drawings) are relevant as well. To establish a list of priority, we asked for the three most important file formats during daily work. Thereby, Excel, PDF and PowerPoint are considered being most important (cf. Figure 3.2B). Therefore, POIL must be able to deal with a large number of file formats and applications.

To answer SRQ3, we took a closer look at the quality of process information. Since the structure and quantity of process information affect its quality [166], we investigated these factors as well. Most process information is available in unstructured form. However, as unstructured process information is difficult to handle, users often try to store process information in a structured way (e.g., in folders). In seven of the eight interviews, it was stated that the existence of process information is more important than its quality. Reason is that employees are only able to perform processes in an effective manner if process information exists (regardless of quality). However, the interviews showed that users often have no overview on available process information due to its large amount; i.e., they often cannot say whether or not they have access to all necessary process information. In turn, this leads to decreased process quality. Moreover, the amount of process information is classified by the participants as too much (cf. Figure 3.3A).

By contrast, the quality of process information was rated differently (cf. Figure 3.3B). Some process information is rated as “excellent” (e.g., records in databases, own documents, information about own tasks), whereas other information is rated as “below average” (e.g., process documentation, information about third-party tasks).

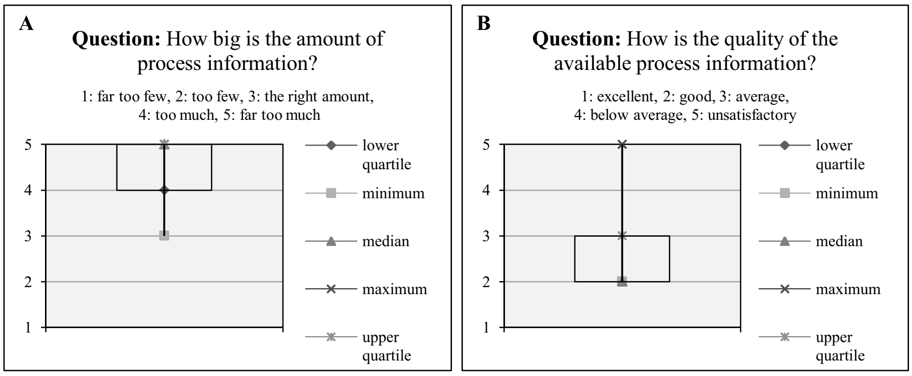

Figure 3.3: Amount and quality of process information.

Hence, we derive the second requirement from the viewpoint of process information.

Requirement 2 (R2). POIL should be able to gather process information from large, distributed and heterogeneous data sources, transform different process information into uniform structured information objects (on different granularity and quality levels), and integrate them into a comprehensive approach.

RQ3. To investigate the employees' process context we asked for factors that can be used to identify a specific context. The participants confirmed that process context is determined based on the progress of a process, e.g., using milestones or quality gates. Some interviewees stated that some documents have metadata comprising the relation to process tasks. Another possibility is the information progress (e.g., customer data available to 80%) from which the process progress can be derived. Moreover, the participants stated that a work context can be determined by folder names, which are often labeled with the name of a respective milestone. Other useful information to determine a specific work context are, for example, user names, roles, departments, project memberships, date, location, and time. In summary, the more information is considered, the more accurately a work context can be determined. Thus, context information should be used in POIL to determine the process context of process participants.

Hence, we derive the following requirement from the viewpoint of context information.

Requirement 3 (R3). POIL should be able to gather context information from sensors, transform context information into uniform structured context objects (on different granularity and quality levels), and integrate them into a comprehensive approach.

3.4 Case Study 2: Clinical Domain

In the second case study, we considered a process of an unplanned, stationary hospitalization in a surgical clinic. The process includes patient admission, medical indication in the anesthesia, surgical intervention, post-surgery stay on the ward, patient discharge, and financial accounting. Most of the case study participants were medical staff such as doctors or nurses, but we also interviewed employees responsible for financial accounting. We selected these employees because of the knowledge-intensive business processes (e.g., diagnostic procedures) they are involved in. Note that the second case study approves and generalizes identified requirements of the first case study (cf. Section 3.3).

RQ1. The case study showed that most processes are not documented. Especially, inexperienced staff does not know how processes shall be documented and which process information they need. As responsibilities for processes are not clearly defined, this often leads to difficulties and misunderstandings regarding the collaboration between departments. Consequently, communication between departments is error-prone.

There are delays in the entire process (e.g., delayed diagnosis) because departments focus only on their own tasks. Problems further exist due to the lack of adherence to agreements, e.g., about the procurement of hospital beds to patients. Hence, these results show that documentation of business processes is crucial to overcome these issues. Hence, the results of RQ1 confirm our Requirement R1.

RQ2. Like in the automotive case study, we investigated the second research question based on the three sub-research questions as depicted in Table 3.3.

To answer SRQ1, we investigated the location of process information. In the clinical sector, both standard applications (e.g., SAP) and individual ones are in use. Clinical staff interacts with these applications using fat clients (e.g., DIACOS[4]) as well as thin clients (e.g., iMED[5], CIRS[6]). Process information is available on shared drives, on local drives, on the Internet, in digital archives, and in paper-based form (e.g., patient files, medical reports, lab reports, and patient checklists). Our study has revealed that a large amount of process information is not available in electronic form at all (cf. Figure 3.4A). Therefore, exchange of process information between departments is often handled manually and only automated to a limited degree. Usually, the business processes and their tasks are not implemented but scattered over multiple systems. For example, patient data is entered in a SAP system, then printed, and further processed by different applications. Hence, these results confirm Requirement R2; i.e., ease of access.

To answer SRQ2, we analyzed file formats and applications (e.g., SAP, DIACOS). Six out of eight interviewees confirmed that they mainly use PDF and Word files. None of the participants uses audio files and only one of them uses video files (medical tutorials). Like in Case Study 1, we asked for the three most important file formats the participants

[4] *DIACOS* is an application for documenting clinical services.
[5] *iMED* is an enterprise-specific application of our case study partner.
[6] *CIRS* is an anonymous reporting system of critical events.

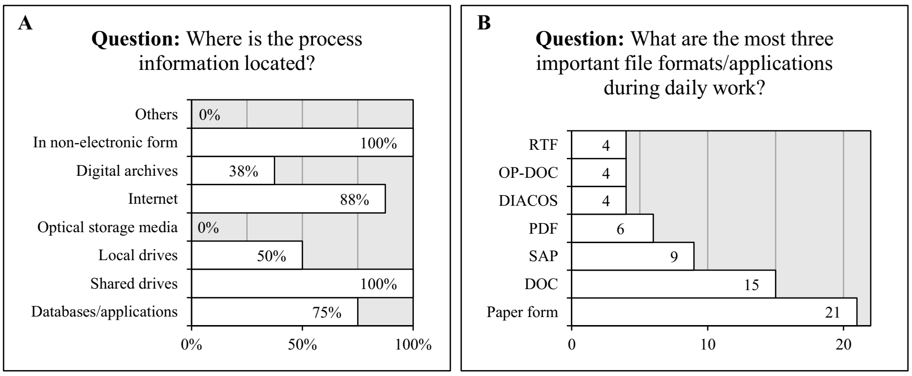

Figure 3.4: Location of process information and important file formats/applications.

need during their work. The formats most frequently used are paper-based documents, Word files, and SAP data records (cf. Figure 3.4B). Similar to Case Study 1, we have a large amount of different file formats and applications needed during daily work.

To answer SRQ3, we re-addressed quality issues of process information. Like in Case Study 1, we considered the structure and quantity of process information. Again, most process information is only available in unstructured form. Further, Case Study 2 shows that problems are the poor quality (e.g., poorly maintained data) and the incompleteness (e.g., not all necessary information is available on the emergency protocol) of process information. Besides, process information is often outdated, e.g., due to the lack of responsibilities concerning maintenance. The amount of process information is classified by most interviewees as “too few” (cf. Figure 3.5A). Process information is typically paper-based and only one person at a certain point in time can access this information (e.g., the patient file is needed for investigations, reporting, patient care, and accounting).

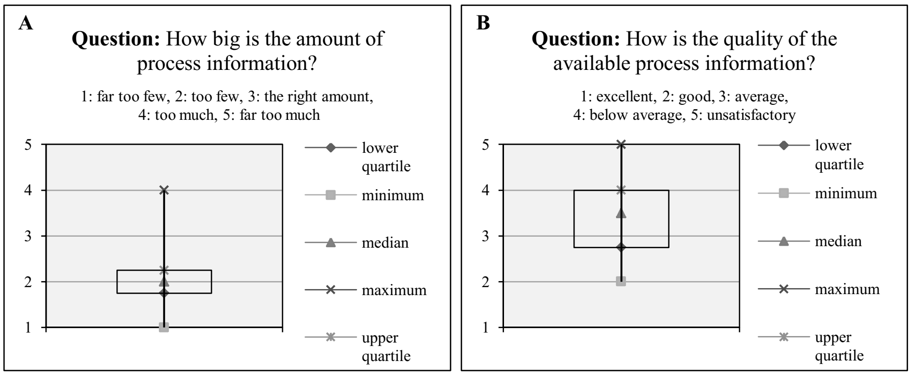

Figure 3.5: Amount and quality of process information.

Quality of process information is rated differently (cf. Figure 3.5B). Reason is that the quality of process information differs depending on the data source. For example, self-made process information is ranked higher than third-party one. The issues considered confirm Requirement R2; i.e., process information should be available in an adequate quality to effectively and efficiently perform business processes.

RQ3. Useful information to determine a process context can be time, users or individual computers (because some devices are only used for performing certain process tasks). Also, the user's location can be helpful, e.g., in order to determine whether a doctor is currently on ward or in the operating theater. However, four out of eight interviewees believe that it is difficult to determine a process context in healthcare processes. In particular, there are no fully pre-specified processes. Instead, they dynamically evolve and many tasks are performed manually without any IT support. When considering tasks supported by applications, the process context can be determined based on the information progress (e.g., is a patient ready for accounting or already settled). Information state changes (e.g., State 1: "patient is in the operating theater" or State 2: "patient is on the ward") in applications can be used to determine the process context as well.

When considering the results of RQ3, it is obvious that these results are consistent with the results of our case study in the automotive domain. Therefore, Requirement R3 has been confirmed by the clinical case study as well.

Hence, Requirements R1-R3 ensure *process-*, *information-* and *context-awareness* in POIL. Therefore, they can be considered as the most important POIL requirements.

3.5 Online Survey

In our online survey, 219 employees from more than 100 enterprises participated. 26% of the participants have been decision makers and 57% knowledge workers. The remaining 17% provided no information about their position. The majority of participants was between 26 and 35 years old (54%). With 23%, participants between 36 and 49 years represent the second largest group of the online survey.

RQ1. In the first part of the online survey, we wanted to know whether or not business processes are documented. Most processes are fully or partially documented (cf. Figure 3.6). No one from the production industry reported that processes are undocumented. Only a small group of participants stated that business processes do only exist in their minds or their work is not oriented towards business processes.

We further wanted to know whether the employees' daily work is guided by documented business processes (cf. Figure 3.7). More than half of the respondents stated that they follow predefined business processes. 27.4% of them follow at least self-defined processes. Only 13.2% of respondents said that they perform their daily work without considering predefined business processes.

Interesting results were also given by means of individual statements of survey participants. Several participants confirmed that people are the most important information source since they can deal with difficult questions or explain other people's tasks. Participants also pointed out that inexperienced staff will benefit most from the delivery

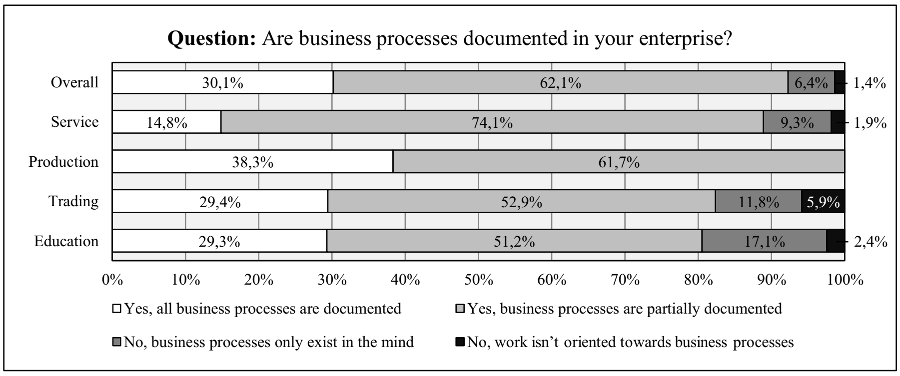

Figure 3.6: Documentation of business processes.

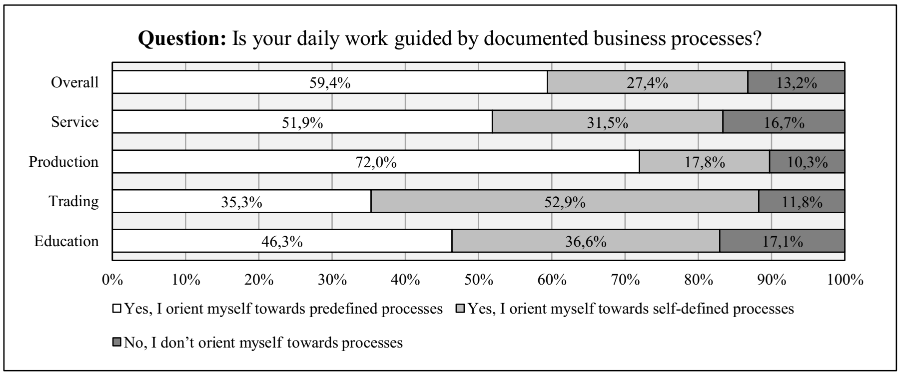

Figure 3.7: Guidance of business processes.

of relevant process information. Another participant said that if processes are undocumented, the identification of a process context will gain importance. Based on these considerations, we can confirm Requirement R1 by our online survey.

RQ2. We asked where needed process information is located. Most participants referred to databases, applications, shared and local drives, and the Internet as the most important sources of process information (cf. Figure 3.8A). When comparing shared and local drives it becomes evident that the majority of process information is stored on shared drives (86%). The most important file formats are PDF, Excel, PowerPoint, and Word (cf. Figure 3.8B). Based on these considerations, we can confirm Requirement R2.

RQ4. We further investigated the relevance of process information. Many participants stated that self-made process information (e.g., own documents, e-mails) have a greater relevance than third-party information. The participants stated that relevance of process information often depends on specific topics and concepts (e.g., a specific disease in the clinical domain). To enable *conceptual* relationships between process information, business processes, and context information, we derive a fourth requirement.

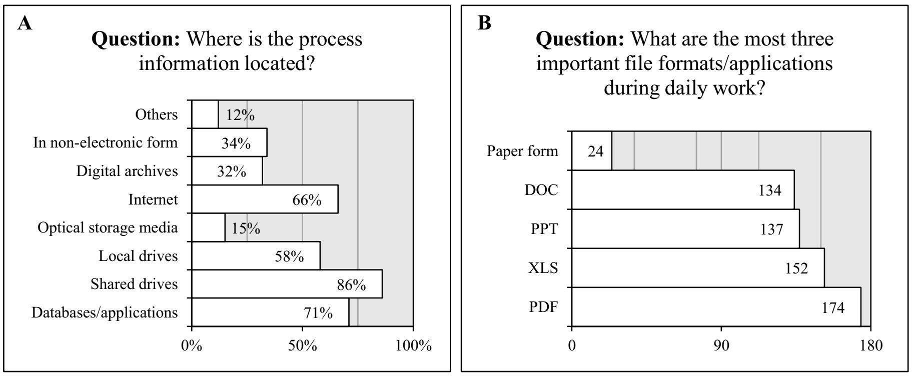

Figure 3.8: Location of process information and important file formats/applications.

Requirement 4 (R4). POIL should be able to analyze information, process and context objects on a conceptual level (i.e., based on the topic of the object) in order to identify conceptual relationships (= conceptual relevance) between objects.

The survey results further reveals a direct relationship between the frequency a particular information is accessed and its relevance. The more frequent a particular process information is accessed, the higher will be its relevance. Hence, participants confirmed that standardized process information is more relevant than non-standardized one. In this context, participants confirmed that the relevance of process information is significantly influenced by the reliability of the information source. Additionally mentioned relevance factors include the frequency of information changes, the date of the last access, and the amount of metadata assigned to a process information. Moreover, participants stated that only up-to-date and complete process information can be relevant.

The accessibility to process information is denoted as a prerequisite. Other participants stated that process information, which is similar to other one, is important. For example, when preparing a review for a product specification, existing reviews for similar products are of great importance. Therefore, we derive the following requirements enabling *syntactic* and *semantic* relationships between process information in POIL.

Requirement 5 (R5). POIL should be able to analyze information, process and context objects on a syntactic level (i.e., based on metadata of the object) in order to identify syntactic relationships (= syntactical relevance) between objects.

Requirement 6 (R6). POIL should be able to analyze information, process and context objects on a semantic level (i.e., based on the content of the object) in order to identify semantic relationships (= semantical relevance) between objects.

We further analyzed the amount of process information (cf. Figure 3.9). Obviously, decision makers are confronted with too much process information. 45.1% of the decision makers confirm that they have too much or far too much process information (knowledge workers: 24%). Knowledge workers, by contrast, face the problem of being confronted with insufficient process information. 48.1% of the knowledge workers mentioned that they have too little or far too little process information (decision makers: 27.5%). As a consequence, relevance of process information should be determined taking the process participant's work context into account.

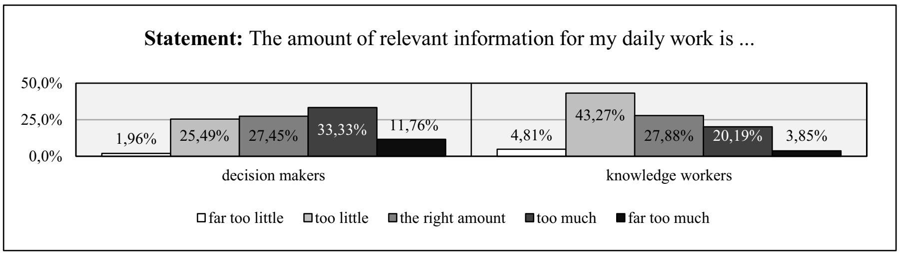

Figure 3.9: Amount of process information.

3.6 Literature Survey

Based on a literature survey, we confirmed the requirements identified in the empirical studies. Additionally, the survey allowed us to derive additional POIL requirements.

For example, Bodendorf et al. confirm that information have to be proactively gathered from different information sources (Requirement R2) [130]. In addition, Knublauch and Rose state that information needed for the execution of business processes is also distributed across different information sources (Requirement R2) [127]. According to Lendholt and Hammitzsch, different users are, on the one hand, interested in different parts of information (Requirements R3 and R8) and, on the other, should be delivered with information taking the area of interest into account (Requirement R4) [141, 142]. Karim et al. state that there is a considerable amount of information generated within the phases of business processes (e.g., execution, monitoring) (Requirement R2) [134]. Moreover, several authors [42, 136, 137, 141, 142] state that objects (i.e., process information, business processes, or context information) and their relationships should be represented in a user-adequate and machine-interpretable manner to identify objects linked with each other. Therefore, the following requirement applies.

Requirement 7 (R7). POIL should be able to represent information, process and context objects as well as their relationships in a meaningful machine- and user-interpretable form and, moreover, in a structured way.

According to Karim et al., one of the major requirements in respect to IL is the file format issue, which means that a file must be transformed into a uniform format

(Requirement R7) [134]. Another requirement described by Karim et al. concerns the structure of the file's content, which means that content must be structured in accordance with a known, accepted and committed model that describes both the structure (Requirement R7) and semantics (Requirement R6) of the content [134]. However, a representation of objects and relationships is still not sufficient; i.e., objects and relationships should be used to deliver relevant process information to process participants [48, 77, 109, 122, 134, 140]. Therefore, we derive our eighth requirement.

Requirement 8 (R8). POIL should be able to use information, process and context objects to guide a process-oriented and context-aware delivery of relevant process information (i.e., information objects) to process participants.

According to Czejdo et al., providing the right information is a major requirement in IL (Requirement R8) [136, 137]. Willems et al. indicate that context cannot be overlooked when IL is considered (Requirements R3 and R8) [140]. Lahrmann, in turn, mentions that the high-quality information supply of the organization is a key requirement in IL (Requirement R8) [109]. Moreover, Haseloff mentions that information should be represented in a structured and human-readable way (Requirement R7) [42]. Further note, Haseloff states that in IL a large amount and variety of information is gathered and processed (Requirements R2 and R3) [42]. Meissen et al., for example, state that retrieval techniques (Requirements R4-R6) are useful to filter relevant information when considering a user's information demand (Requirements R3 and R8) [77, 122].

Moreover, Bucher and Dinter describe research on the design factors of IL [48]. The latter are as follows: (1) "excellence in information supply" (Requirement R8), (2) "integration of analytic information into process execution" (Requirements R2 and R3), (3) "utilization of advanced instruments (e.g., simulations) for information access and analysis" (Requirements R4 and R6), and (4) "utilization of basic instruments (e.g., standard reports) for information access and analysis" (Requirement R5).

3.7 Requirements at a Glance

Table 3.4 summarizes Requirements R1-R8, which are fundamental for realizing POIL. The requirements reflect needs of process participants such as knowledge workers and decision makers. They further concern technical issues necessary to enable the delivery of relevant process information. All requirements have been derived based on results of the empirical studies (i.e., two case studies in the automotive and clinical domain as well as an online survey) and were confirmed by a literature survey.

Note that Requirements R1-R3 ensure process-, information- and context-awareness in POIL. Therefore, they can be considered as the most important requirements for realizing the POIL framework. Further note, that Requirements R4-R6 enable bridging the gap between process information and business processes in enterprises.

As shown in Table 3.4, missing integration of business processes and tasks (i.e., process-awareness) in contemporary IL solutions has guided the POIL development.

#	Requirements	CS1	CS2	OS	LS
R1	*Integration* of business processes	×	×		
R2	*Integration* of process information	×	×		×
R3	*Integration* of context information	×	×		×
R4	*Conceptual* analysis of objects			×	×
R5	*Syntactic* analysis of objects			×	×
R6	*Semantic* analysis of objects			×	×
R7	*Representation* of objects and relationships				×
R8	*Delivery* of relevant process information				×

CS1 = Case Study 1, CS2 = Case Study 2, OS = Online Survey, LS = Literature Survey

Table 3.4: Fundamental requirements enabling POIL.

3.8 Summary

We have presented the results of two exploratory case studies as well as an additional online survey with 219 participants from more than 100 enterprises. Furthermore, we have presented the results of a literature survey that supports the empirical findings. Based on these findings, we have revealed eight fundamental requirements (R1-R8) that need to be addressed when realizing POIL. Note that all these requirements have been considered when designing and developing the POIL framework. Moreover, we may assume that these requirements apply to other domains (e.g., financial services) as well.

Part II

Process-Oriented Information Logistics Framework

4 Process-Oriented Information Logistics in a Nutshell

This chapter[1] presents the developed POIL framework. Section 4.1 provides background information needed for understanding this chapter. Section 4.2 introduces a running example that will be used throughout the chapter. Additionally, Section 4.3 describes five levels of IL. The most advanced one (i.e., the 5th IL level), which corresponds to POIL, is presented in Section 4.4. Then, Section 4.5 applies the POIL framework to the running example and shows how three different real-world use cases can be supported. Section 4.6 summarizes the chapter.

4.1 Background Information

The presented research is performed in the *niPRO* project[2], in which we apply semantic technology to realize intelligent, user-adequate *process information portals*. The overall project goal is to support both knowledge workers and decision makers with personalized process information depending on their current work context. The niPRO project itself is based on two pillars: *process-oriented information logistics* (POIL) [3, 7] and *process navigation and visualization* (ProNaVis) [12, 167] (cf. Figure 4.1).

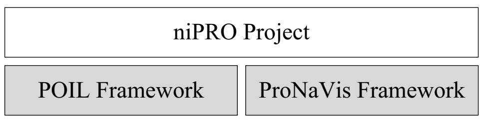

Figure 4.1: Pillars of the niPRO project.

POIL targets at the delivery of the right process information, in the right format and quality, at the right place and the right point in time, to the right people [3]. The latter need not actively search for process information anymore, but will be automatically supplied with one, even if their work context is dynamically changing.

ProNaVis, in turn, aims at enabling flexible navigation within complex business processes and related process information [12]. The ProNaVis framework applies innovative visualization and navigation concepts to visualize and deliver business processes and related process information in an intelligent and user-adequate manner.

[1]The chapter is based on the following referred paper:
[3] B. Michelberger, B. Mutschler, and M. Reichert. *Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes.* In: Proc 16th Int'l Enterprise Computing Conf (EDOC'12), IEEE Computer Society Press, pp. 21–30, Beijing, China, 2012.
[2]More information can be found at http://nipro.hs-weingarten.de.

4.2 Running Example

This section introduces a use case from the clinical domain that will be used throughout this chapter. The use case (cf. Figure 4.2) is based on the results from a case study we performed at a large German university hospital [2, 63]. It deals with the prescription, procurement and administration of drugs. The underlying process is knowledge-intensive; i.e., it comprises tasks such as patient examination and diagnosis, and involves a lot of process information (e.g., patient records, lab reports, medical orders, drug stock lists), user interaction (e.g., examine patient, create medical orders, define drug administration), and decision making (e.g., on the drugs to be prescribed for a patient).

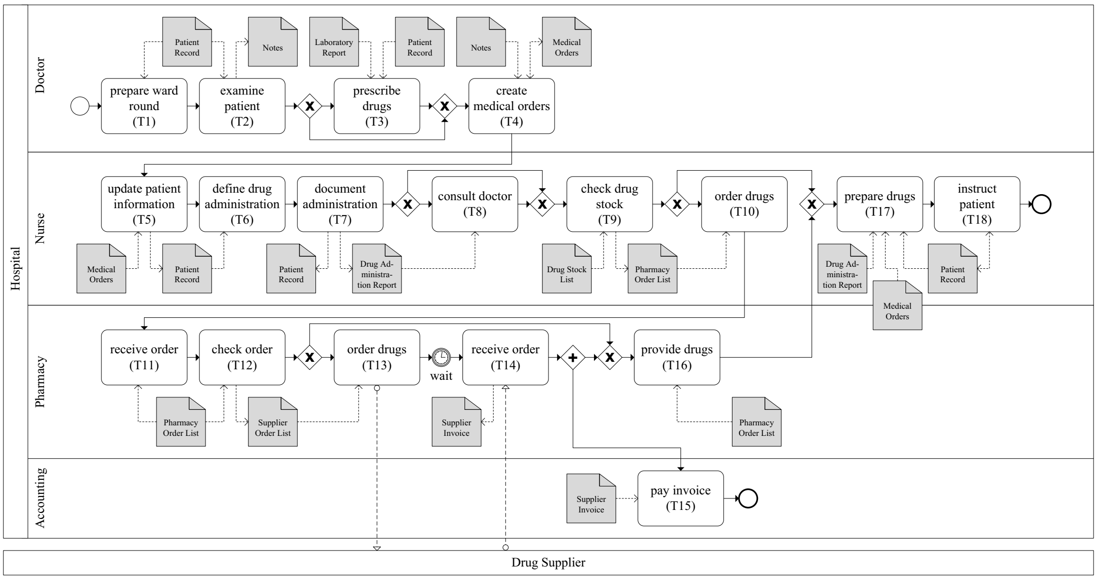

Figure 4.2: Running example: procurement of drugs.

The process comprises five roles: (1) *doctor*, (2) *nurse*, (3) *pharmacy*, (4) *accounting*, and (5) *drug supplier*. During the ward round, the doctor prescribes drugs (Task T3) for a particular patient. Based on that prescription, a nurse updates the patient record accordingly (Task T5); i.e., the drugs and their use are documented. After the ward round [168], the prescribed drugs are ordered (Task T10); i.e., an order form is filled out by the nurse and sent to the hospital pharmacy. The latter checks whether the drugs are available (Task T12), and – if yes – delivers them to the nurse. If needed drugs are not available, they will be ordered (Task T13) by a pharmacy assistant and delivered to the nurse as soon as they will be available. If the needed drugs are available, the nurse will prepare them (Task T17) and instruct the patient about their effects (Task T18). Finally, the doctor examines the effects during the next ward round (Task T2).

Regarding this process, we must distinguish between the *process schema* (as shown in Figure 4.2) and enacted *process instances* [53]. Likewise, we distinguish between *process information schemas*, *process information instances*, and *abstract process information*. More specifically, process information schemas include, for example, any kinds of templates, e.g., for medical reports, order forms, or patient records. Process information instances, in turn, are instantiated process information schemas, e.g., patient-specific medical reports, records, or filled forms. Finally, abstract process information cannot be instantiated. As examples consider working guidelines, manuals, and best practices.

Note that not all types of process information occur in both process schemas and process instances (cf. Table 4.1). Process information schemas only occur in process schemas, while process information instances only occur in process instances. Finally, abstract process information can occur in both, i.e., process schemas and instances [3].

#	**PI Schema**	**PI Instance**	**Abstract PI**
Process Schema	×		×
Process Instance		×	×

PI = Process Information

Table 4.1: Use of process information in process schemas and instances.

Based on the presented running example, we introduce different levels of IL (cf. Section 4.3) and the concepts underlying the notion of POIL (cf. Section 4.4).

4.3 Levels of Information Logistics

This section describes five different levels of *information logistics* (IL). Existing IL approaches (cf. Section 2.4) realize the 1st, 2nd and 3rd level. Building upon level 3, we introduce two additional levels. The most advanced one corresponds to POIL.

Level 1: Hard-wired Information Logistics. This initial level comprises two architectural layers. A *data layer* manages data sources and an *application layer* delivers process information from the data sources to the users (cf. Figure 4.3). For example, think of an application or a simple Intranet portal that allows users to access process information. Process information and applications (e.g., Intranet portals) are manually linked; i.e., hard-wired and usually based on pre-defined categories such as organizational units, project milestones, quality gates, or specific process schemas.

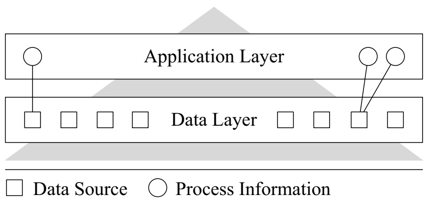

Figure 4.3: Level 1: hard-wired IL.

Figure 4.3 visualizes the manual linkage of process information and applications. On this level, there is no possibility to realize advanced IL features such as the handling of different levels of process information granularity [1]. Regarding our example from Section 4.2, when a doctor prescribes drugs (cf. Task T3 in Figure 4.2), only entire lab reports can be delivered to the doctor, and not parts of it.

Level 2: Conventional Information Logistics. This second level comprises three architectural layers. Besides the data and application layer (cf. Level 1), an additional *integration layer* is introduced (cf. Figure 4.4). It corresponds to a conventional middleware layer providing a uniform data interface. Still, process information and applications are manually linked and therefore hard-wired.

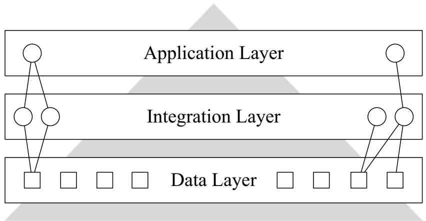

Figure 4.4: Level 2: conventional IL.

Based on the integration layer, additional IL features can be realized like the handling of different process information granularity levels [1] or the handling of technical, syntactical and structural heterogeneity on the data level [169]. In the running example, the provision of certain parts of lab reports thus becomes possible (cf. Task T3 in Figure 4.2). However, it is still not possible to detect relationships between related process information, e.g., between patient records and related lab reports.

Level 3: Intelligent Information Logistics. This level comprises three architectural layers, but the integration layer is replaced by a *semantic layer* (cf. Figure 4.5). Unlike the conventional integration layer from the 2nd level, the semantic layer not only realizes a uniform data interface, but also provides advanced analysis features for examining integrated process information. In order to apply these analysis features, a *semantic information network* (SIN) is created [170]. Similar to an ontology-based model, the SIN not only comprises *information objects* (i.e., process information), but also relationships between information objects (cf. Section 4.4). Note that the semantic layer and its conceptual elements are described in detail in Chapter 5.

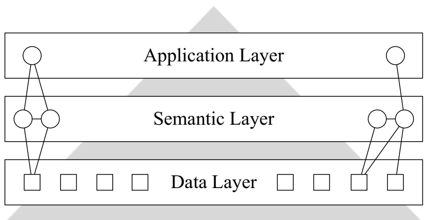

Figure 4.5: Level 3: intelligent IL.

Regarding the example, this means that the doctor can be automatically supplied with process information related to the process information that is currently considered. For example, by viewing a lab report, the corresponding patient record or related lab reports can be automatically determined and delivered to the doctor.

Level 4: Context-aware Information Logistics. This level comprises four architectural layers. Besides the data, semantic and application layer (cf. Level 3), an additional *context layer* is introduced (cf. Figure 4.6). The purpose of this additional layer is to enable the context-aware delivery of process information. The context layer, therefore, continuously analyzes a knowledge worker's situation based on available context information [42]. The latter is gathered from different data sources (i.e., sensors) and includes, for example, user name, responsibilities, time, location, and used device. Note that the context layer and its conceptual elements are described in detail in Chapter 6.

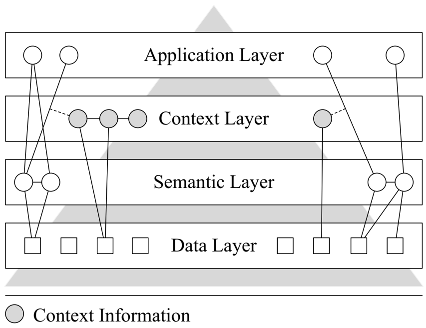

Figure 4.6: Level 4: context-aware IL.

Available context information allows filtering the previously discussed SIN (and process information accordingly). This allows us, for example, to identify lab reports or patient records which are currently needed according to the doctor's work context.

Level 5: Process-Oriented Information Logistics. This level comprises four layers as well. The main difference to the 4th level is the additional consideration of business processes (besides the consideration of process information and context information) (cf. Figure 4.7). More precisely, business processes (i.e., process schemas and instances) are integrated. This is achieved by splitting them up into their constituent elements (e.g., tasks, events). Each process element is treated as a single *process object* in the SIN. Hence, the SIN not only contains information objects, but process objects as well.

The SIN is enriched and becomes more comprehensive as it includes both process information and business processes. Section 4.4 will show that a more effective alignment of process information with business processes (both on the process schema and the process instance level) can be achieved based on this enriched SIN. Referring to the clinical example, a doctor can now be provided with the needed patient records when performing respective process tasks (e.g., examine patient) (cf. Task T2 in Figure 4.2).

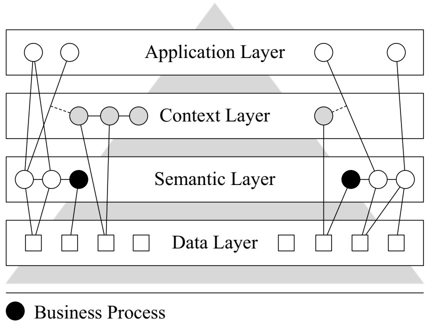

Figure 4.7: Level 5: process-oriented IL.

Generally, the goal of POIL is to provide the right process information, in the right format and quality, at the right place and the right point in time, to the right people. Therefore, users need not actively search for relevant process information anymore since they are automatically supplied with the latter – even if their work context is changing.

Altogether, POIL combines information-, context- and process-awareness. POIL is *information-aware* as it allows effectively handling process information and its meaning. POIL is *context-aware* as it supports the use of context information to characterize the process participant's situation. Finally, POIL is *process-aware* as it allows integrating and analyzing business processes (both process schemas and process instances).

Based on these characterizations, POIL can be defined as follows:

Definition 13 (**Process-Oriented Information Logistics**). *Process-oriented information logistics* (POIL) deals with the planning, execution and control of process information flows within or across enterprises to support knowledge-intensive business processes involving large amounts of process information, user expertise, user interaction, creativity, and decision making. In particular, POIL aims at the process-oriented and context-aware delivery of process information to knowledge workers and decision makers. Goal is to no longer manually link process information to business processes and/or their tasks, but to automatically identify and deliver the process information relevant for process participants (i.e., knowledge workers and decision makers).

4.4 The Framework

Missing process-awareness in contemporary IL solutions (cf. Section 2.4) has guided our development of *process-oriented information logistics* (POIL) [3]. POIL aligns process information with business processes, both at the schema and instance level [53], enabling a process-oriented, context-aware delivery of process information to process participants. The main idea of POIL is to split up business processes into their constituent process elements and to integrate the latter with comprehensive process information [7].

As aforementioned, enabling POIL requires four architectural layers: (1) *data layer*, (2) *semantic layer*, (3) *context layer*, and (4) *application layer* (cf. Figure 4.8). These layers and their interplay are described in the following. We put a strong focus on the semantic layer since it is the most important layer of the POIL architecture.

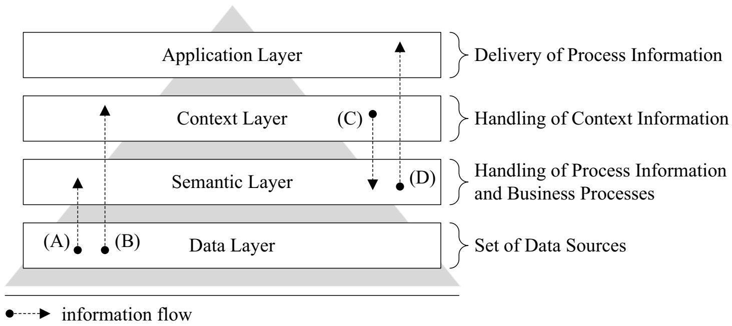

Figure 4.8: Interplay of POIL architecture levels.

Data Layer. The *data layer* makes the set of data sources (e.g., databases, shared drives, enterprise information systems) to be integrated into POIL available; i.e., the sources of process information, context information, and business processes.

Semantic Layer. The *semantic layer* is responsible for the integration and analysis of process information and business processes. After integrating these into the SIN (cf. Figure 4.8A), the SIN is analyzed by using various algorithms. Gathered context information (cf. Figure 4.8B) is then used to filter the SIN (cf. Figure 4.8C). This enables the semantic layer to deliver currently needed (i.e., relevant) and context-aware process information to knowledge workers and decisions makers (cf. Figure 4.8D).

The most important component of POIL is the SIN (created by the semantic layer). It is created following a bottom-up approach and comprises *information objects*, *process objects*, and inter-object *relationships* (cf. Section 5.3). Information objects include process information schemas, process information instances, and abstract process information. Process objects, in turn, include elements of process schemas and process instances. Each of these objects may be associated with metadata (e.g., file size, file format, modification date). Relationships may exist among information objects (e.g., a file which is similar to another one), among process objects (e.g., an event which triggers a particular task), and between information and process objects (e.g., a file required for the execution of a task). Furthermore, relationships are labeled (with the reason of the relationship) and weighted (with the relevance of the relationship) [171]. This allows determining why objects are related and how strong their relationship is.

Context Layer. The *context layer* is responsible for integrating and analyzing context information. In [4], we have described a framework realizing the context layer. Context information is gathered from data sources called sensors. We distinguish between physical (e.g., thermometer), virtual (e.g., keyboard input), and combinatorial sensors (e.g., sensors which allow detecting a process participant's position by analyzing logins at devices and a mapping to locations) [85]. In addition, other context information can be also derived from existing one as well (e.g., by aggregation or interpolation).

A *context model* (CM), which is created based on available context information, allows characterizing a process participant's work context which can then be used to filter the SIN. The CM is completely independent from the SIN; i.e., context objects (e.g., role, time, location) are only stored in the CM but not in the SIN (cf. Section 6.4).

Application Layer. Finally, the *application layer* is responsible for the joint presentation and context-aware delivery of process information and business processes to users. Note that the application layer and its conceptual elements are described in Chapter 7.

4.5 Use Cases

Along three use cases, we demonstrate how the POIL framework can support process participants with personalized and contextualized process information.

Use Case 1. Consider Task T2 of our clinical use case (cf. Section 4.2), i.e., the patient examination for which the doctor needs access to patient records (cf. Figure 4.2). In practice, patient records include extensive and various kinds of information (e.g., master and transaction information, department-specific information, or historical information). However, only small parts of a record (e.g., former diseases, pre-existing conditions, course of diseases) are actually needed in the context of a patient examination. POIL is able to provide the needed parts to doctors when patients are examined.

A SIN is created as follows: First, the business process, i.e., our clinical business process (cf. Section 4.2) is integrated. Second, needed process information is integrated; i.e., the patient records. Then, relationships among process objects or information objects and between process objects and information objects are determined. Performing these steps is the prerequisite to handle this use case. Figure 4.9 shows an exemplary SIN for the role doctor. In order to provide contextualized process information the CM is then created (cf. Section 4.4). The creation of the CM is initiated at specific points in time (e.g., by a scheduled job) or when a user performs a certain task.

Let us assume that the doctor is on his ward round: Based on the context layer, the doctor's work context can be determined. For this purpose, location identification technologies (e.g., satellite networks, cellular networks, or indoor networks) can be used. Having identified the doctor's location, the technical position (e.g., GPS coordinates) is mapped to a logical one (e.g., a room number) using a hospital building map. Analogous to the integration and analysis of the doctor's location, other context information such as time, device or role may be considered. Further, additional context information (e.g., bed occupancy) allows determining which patients are staying in which room.

Combining this context information with process information, the doctor can be provided with relevant patient records according to his location. For example, if patient Jon

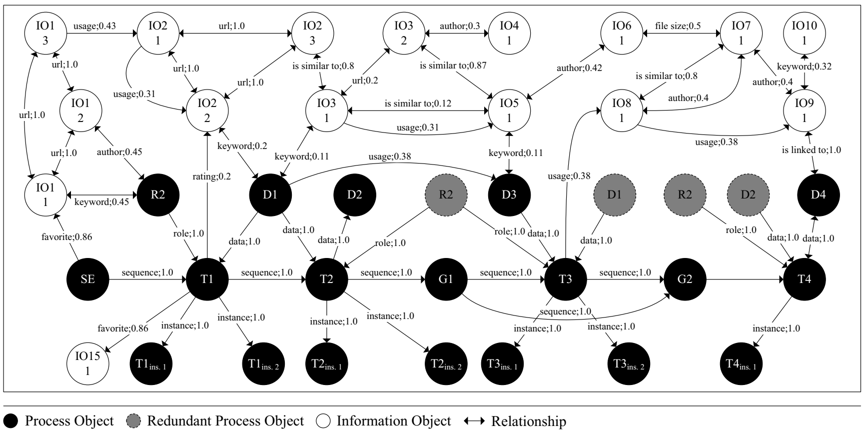

Figure 4.9: Exemplary SIN of our running example (role: doctor).

Doe is in room A301 and the doctor enters the room, the latter automatically shall get access to the patient record of Jon Doe (e.g., on his tablet computer). The granularity level of the patient record depends on the user's role and the current task; i.e., only parts of the patient record may be provided which are necessary for examining the patient.

Use Case 2. The second use case is based on Task T3 of our use case (cf. Figure 4.2); i.e., the prescription of drugs. This task is knowledge-intensive (due to questions like which disease the patient has and which treatment he should get) and includes, among others tasks, the interpretation of symptoms (e.g., excessive thirst, tiredness), the determination of diseases (e.g., diabetes mellitus type 2), and the identification of treatment options (e.g., physical activity, balanced diet, prescription of drugs) (cf. Figure 4.10).

POIL can support these tasks. For this purpose, we assume that different process information about symptoms, diseases, treatment options, and drugs are integrated in the SIN. Data sources are, for example, digital medical libraries or health web portals. Without POIL, the relationships between symptoms, diseases and treatment options are not given. With POIL, the semantic layer determines relationships and meaning of process information (e.g., which symptoms belong to which disease and how to treat the disease). After the doctor has entered the symptoms, POIL is able to make suggestions which diseases the patient may have and which treatment options exist. For this use case we have developed a proof-of-concept prototype, called iCare (cf. Section 8.1).

Use Case 3. The third use case is based on Task T17 (cf. Figure 4.2) of our running example; i.e., the preparation of drugs. To perform this task, certain process information is needed, e.g., drug administration reports, medical orders, lab reports, and patient records. The range of available data sources and the amount of process information often makes it difficult for nurses to find needed process information. For example, they often do not know which process information is needed and where it can be found. Therefore, they often need a very long time to search for process information.

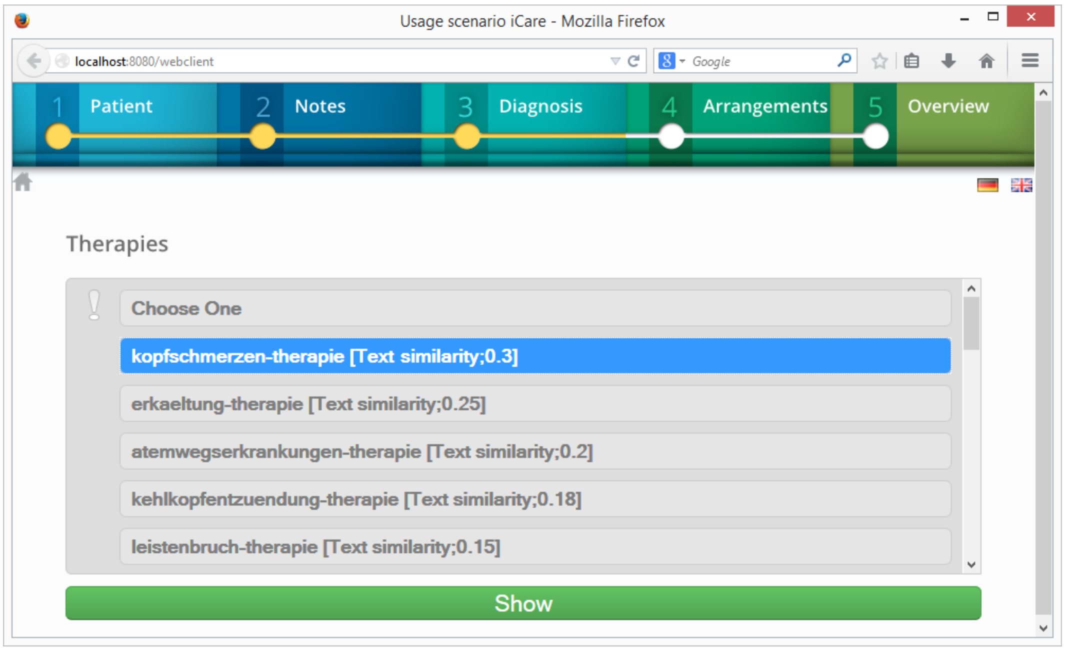

Figure 4.10: iCare: identification of treatment options.

The POIL framework may accomplish this task. By integrating business processes (both on the process schema and instance level), process information can be effectively and efficiently aligned with business processes and their tasks. Figure 4.11 shows an exemplary SIN for the role nurse. Note that we assume that for all considered process instances in POIL the corresponding process schemas are available. If the context layer determines that a nurse currently prepares drugs (by the use of the CM), the POIL framework can automatically provide the needed process information to the nurse.

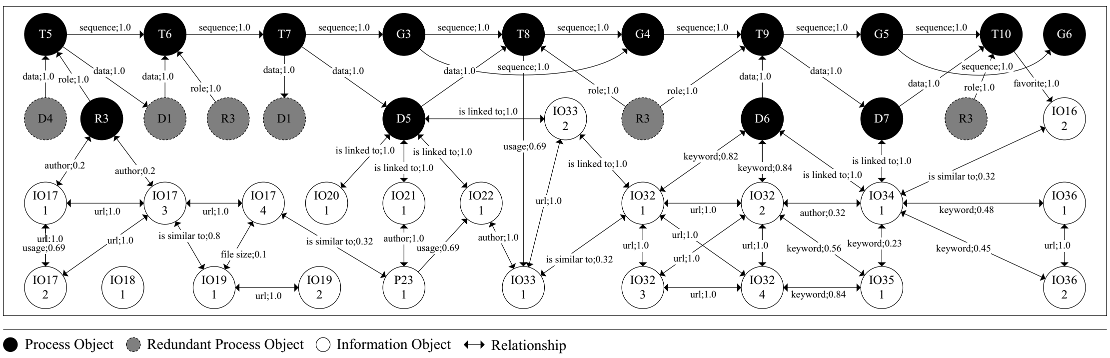

Figure 4.11: Exemplary SIN of our running example (role: nurse).

Besides the mentioned three use cases, POIL is relevant for numerous other use cases such as process monitoring, time management, expert finding, collaboration, and decision support [4]. More information can be found in Chapter 8 of this thesis.

4.6 Summary

This chapter has presented a novel approach, called *process-oriented information logistics* (POIL), to bridge the gap between process information and business processes. The contribution of the chapter is threefold: First, we sketched the path from conventional IL to POIL. Second, we introduced basic POIL concepts and presented architectural layers of POIL. Third, we demonstrated the application and benefits of the approach along three characteristic use cases.

5 Semantic Layer

This chapter[1] discusses the *semantic layer* of the POIL framework. Section 5.1 gives an introduction. Section 5.2 then describes the semantic layer in detail and introduces its underlying core component, i.e., the *semantic information network* (SIN). Section 5.3 shows how the latter is used to represent process information and business processes in a meaningful as well as machine- and user-interpretable way. Section 5.4 deals with algorithms related to the maintenance of the SIN, whereas Section 5.5 presents two additional algorithms for determining the relevance of process information. Finally, related work is discussed in Section 5.6 and a summary is provided in Section 5.7.

5.1 Introduction

The goal of POIL is to provide process participants with relevant process information when working on business processes and associated tasks. The key layer of the four POIL layers presented in the previous chapter is the *semantic layer*. It will be introduced in this chapter showing how both *information-* and *process-awareness* in the POIL framework can be enabled (cf. Figure 5.1).

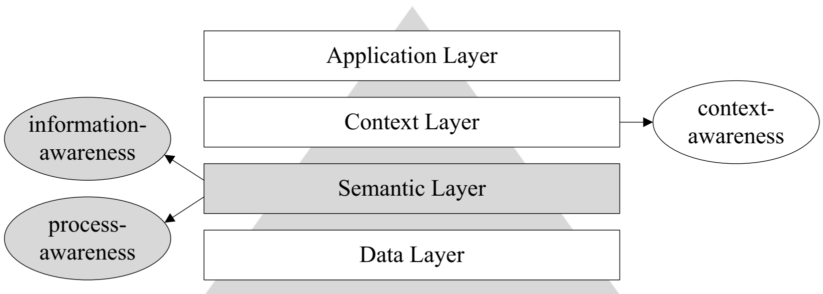

Figure 5.1: Semantic layer enabling information- and process-awareness.

[1]The chapter is based on the following referred papers:
[3] B. Michelberger, B. Mutschler, and M. Reichert. *Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes.* In: Proc 16th Int'l Enterprise Computing Conf (EDOC'12), IEEE Computer Society Press, pp. 21–30, Beijing, China, 2012.
[7] B. Michelberger, B. Mutschler, M. Hipp, and M. Reichert. *Determining the Link and Rate Popularity of Enterprise Process Information.* In: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS'13), LNCS 8185, pp. 112–129, Graz, Austria, 2013.
[13] B. Michelberger, K. Ulmschneider, B. Glimm, B. Mutschler, and M. Reichert. *Maintaining Semantic Networks: Challenges and Algorithms.* In: Proc 16th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'14), ACM Press, pp. 365–374, Hanoi, Vietnam, 2014.

The semantic layer allows bridging the gap between process information and business processes (cf. Research Question 2). More specifically, the semantic layer integrates process information and business processes. In particular, it discovers relationships between process information and business processes that have not been known so far.

Regarding business process support, we need to distinguish between process schemas and enacted process instances [53]. In particular, specific process instances may require different process information [172, 173]. Consider, for example, the review of product requirements: An automotive engineer needs different review templates, manuals, protocols, minutes, and guidelines depending on the product under review.

Based on these considerations, we identify requirements for the semantic layer in POIL. Requirements were gathered based on a literature survey and practical experiences. Table 5.1 summarizes nine fundamental requirements (R1-R9) for the semantic layer.

#	**Requirements**
R1	Process information schemas, process information instances, and abstract process information must be electronically available and accessible.
R2	Process schemas and related process instances must be electronically available and formally specified (e.g., using a process modeling language such as BPMN).
R3	Integration interfaces must be available to transform proprietary process information (e.g., e-mails, office files) into homogeneous information objects.
R4	Integration interfaces must be available to transform process elements (e.g., tasks, events, gateways, data objects) into homogeneous process objects.
R5	Analysis interfaces must be available to transform specific process objects (e.g., a "sequence flow" or "message flow") into process object relationships.
R6	A conceptual analysis interface must be available to determine conceptual relationships between objects (e.g., objects dealing with the same topic).
R7	A syntactic analysis interface must be available to determine syntactic relationships between objects (e.g., objects with the same author in their properties).
R8	A semantic analysis interface must be available to determine semantic relationships between objects (e.g., objects with similar or same content).
R9	A mechanism must be available to continuously check objects and relationships to ensure that they are complete, consistent and up-to-date.

Table 5.1: Requirements for the semantic layer in POIL.

In order to realize the semantic layer in POIL, process information schemas (e.g., templates), process information instances (e.g., filled templates), and abstract process information (e.g., manuals) must be electronically available (cf. Requirement R1). Likewise, process schemas (e.g., a process model of a process) and enacted process instances (e.g., currently running process instances of a process) have to be available as well (cf. Requirement R2). Satisfying Requirements R1 and R2 is a prerequisite to manage both process information and business processes by POIL. More specifically, in order to integrate process information and business processes (and hence to make them available for POIL), integration interfaces are needed (cf. Requirements R3 and R4). The latter transform proprietary formats into uniform ones; i.e., information and process objects. Note that this integration constitutes a preliminary step for the subsequent analysis. Overall, the semantic layer should enable the following types of analyses (cf. Section 3.7): (1) *conceptual analysis*, (2) *syntactic analysis*, and (3) *semantic analysis* (cf. Requirements R5-R7). Major goal is to determine previously unknown relationships between information and process objects. Additionally, the semantic layer needs a mechanism to continuously ensure that information and process objects meet certain quality standards (cf. Section 6.5) to ensure that only complete, consistent and up-to-date process information is provided to process participants (cf. Requirement R8).

This chapter introduces the semantic layer of the POIL framework in detail. Further, it discusses how process information and business processes must be handled to enable the delivery of relevant process information to process participants.

5.2 The Layer

The semantic layer aims to integrate and analyze process information and business processes. The core component of the semantic layer is the *semantic information network* (SIN). The latter comprises information objects (e.g., e-mails, office files), process objects (e.g., tasks, events, gateways), and relationships between information and process objects (e.g., "is similar to", "is used after"). The SIN has been influenced, for example, by related work from the fields of *associative networks* [174], *topic networks* [175], *fact networks* [176], and *ontologies* [177, 178]. However, the fundamental difference between the SIN and existing approaches is the first class treatment of business processes and the alignment of process information with business processes. In turn, this means that existing approaches and requirements may only be partially transferred (cf. Table 5.1).

Basically, the semantic layer comprises two components: (1) *integration component* and (2) *analysis component*. Note that the former is part of the data layer since it provides process information and business processes from different data sources. The analysis component, in turn, is realized by the semantic layer and comprises the SIN. In our approach, the semantic layer is realized based on a middleware infrastructure (cf. Section 6.3). Figure 5.2 illustrates the layered architecture of the POIL framework as well as the alignment of the components needed to create the SIN.

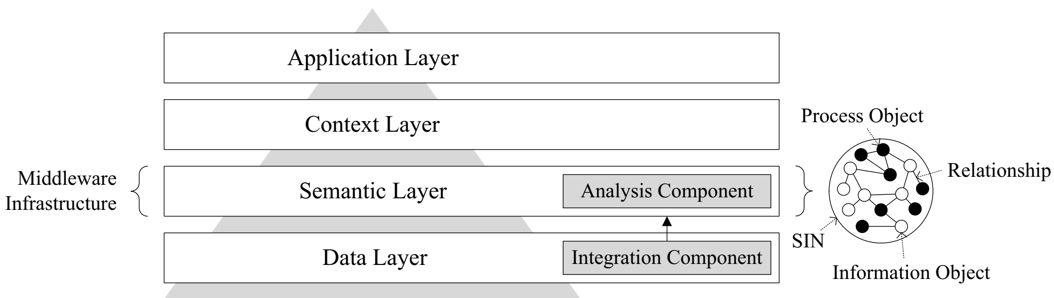

Figure 5.2: Semantic layer.

5.2.1 Integration Component

The *integration component* is responsible for integrating process information and business processes that stem from heterogeneous data sources. In turn, data sources are connected to the integration component by integration interfaces. For each data source one interface must be provided. Interfaces transform process information and business processes specified in proprietary formats (e.g., Excel, PowerPoint, Word, or PDF files) into uniform ones (i.e., information or process objects) (cf. Figure 5.3A).

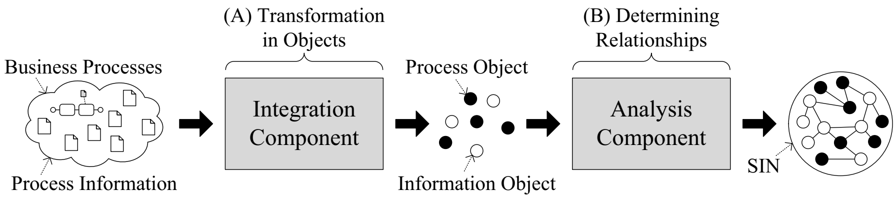

Figure 5.3: Interplay between the components of the semantic layer.

As an advantage of this bottom-up-approach, no pre-defined structure is required for the SIN. Instead, the SIN's structure is automatically created by integrating process information and business processes. More detailed information is provided in Section 5.3.2.

5.2.2 Analysis Component

The *analysis component* is responsible for the syntactic, semantic and conceptual analysis of integrated process information and business processes. Thereby, relationships between them shall be discovered (cf. Figure 5.3B). The analysis is performed in several steps: First, basic properties of integrated process information and business processes, such as authorships or file formats, are compared (*syntactic analysis*). This allows, for example, linking objects in a SIN with the same author (e.g., a specific engineer). Second, the contents of all available information and process objects are analyzed (*semantic analysis*). This allows, for example, linking objects with similar contents (e.g., similar reviews, similar process tasks). Third, topics (e.g., development, project management, quality management) are extracted from information and process objects (*conceptual analysis*). This allows, for example, linking objects dealing with the same topic (e.g., development of car control units). The goal is to further classify and group correlated objects. Finally, user behavior is investigated, e.g., the frequency of using certain information objects in the context of a specific business process. As a final result of the analysis, the SIN is created. In particular, the SIN allows identifying objects linked to each other in the one or other way, e.g., process information needed when performing a particular process task [7]. Thus, the overall goal of the analysis is to discover previously unknown relationships between process information and business processes (cf. Figure 5.3B). More detailed information is provided in Section 5.3.2.

5.3 Semantic Information Network

As motivated, the core component of the semantic layer is the *semantic information network* (SIN). We use it to represent process information and business processes in a meaningful machine- and user-interpretable form. First, *meaningful* necessitates that the semantics of process information and business processes must be captured by the SIN [179]. Second, *machine-interpretable* means that the machine can make inferences based on the relationships of the SIN. Third, *user-interpretable* means that SIN objects and SIN relationships should be comprehensible for process participants as well.

A SIN can be created following a bottom-up approach; i.e., starting with the integration of process information and business processes from different data sources. Following this, integrated process information and business processes are analyzed. The resulting SIN (cf. Figure 5.4) comprises *information objects*, *process objects*, and inter-object *relationships*. More specifically, information objects represent process information in a uniform format. In turn, process objects represent elements of business processes.

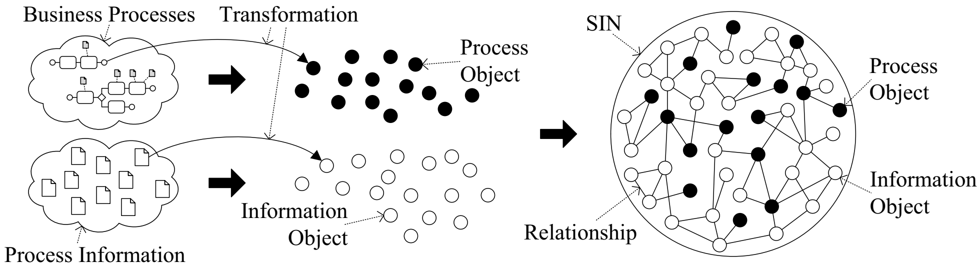

Figure 5.4: Creation of the SIN.

Generally, relationships captured in a SIN may exist between information objects (e.g., a guideline similar to another one), between process objects (e.g., an event triggering a sub-process), and between information and process objects (e.g., an instruction required for executing a particular task) (cf. Figures 5.5A-5.5C). Furthermore, a relationship can either be *explicit* (i.e., hard-wired) or *implicit* (i.e., not hard-wired). An example of explicit relationships are modeled sequence flows in a business process. In turn, implicit relationships can be automatically identified based on a variety of algorithms that link objects related to the same topic or objects used in the same work context (cf. Section 5.3.2). Moreover, inter-object relationships are labeled (e.g., "is a template", "is linked to") and weighted (e.g., "0.4"). A weight is expressed in terms of a number ranging from 0 to 1, with 1 indicating the strongest possible relationship [171]. This allows determining why objects are inter-linked and how strong their relationship is.

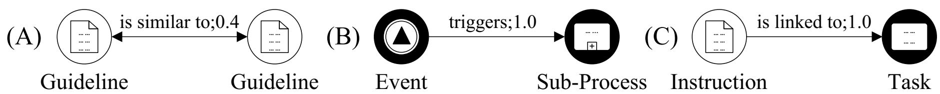

Figure 5.5: Relationships in a SIN.

5.3.1 Definitions

Generally, a SIN is a labeled and weighted *directed graph* [3]. Each directed *edge* $e = (v, v')$ represents a relationship and is associated with an ordered pair of *vertices* (v, v'). Thereby, v and v' correspond to information or process objects; v is the source and v' is the destination of e. Based on this, we define a SIN as follows:

Definition 14 (**Semantic Information Network**). A *semantic information network* (SIN) is a tuple (V, E, L, W, f_l, f_w), where V is a set of *vertices* with each $v \in V$ representing an *information* or *process object*. In turn, E is a multiset of *edges* with each edge $e = (v, v') \in E$, and $v, v' \in V$ representing a *relationship* between objects. Function $f_l\colon E \to L$ labels each edge $e \in E$ with a *label* from the set of labels L. Function $f_w\colon E \to W$ assigns a *weight* from the set of weights W to each edge $e \in E$. Given an edge $e = (v, v') \in E$, we denote v as the *source* and v' as the *destination* of e.

A SIN constitutes a *finite graph*; i.e., V and E are finite sets [3, 180]. A SIN may also contain *slings* (i.e., $\exists\ e \in E\ with\ e = (v, v)$, cf. Figure 5.6A), *parallelism*[2] (i.e., $\exists\ e, e' \in E\ with\ e = (v, v') \ \wedge\ e' = (v, v')$, cf. Figure 5.6B), and *anti-parallelism* (i.e., $\exists\ e, e' \in E\ with\ e = (v, v') \ \wedge\ e' = (v', v)$, cf. Figure 5.6C).

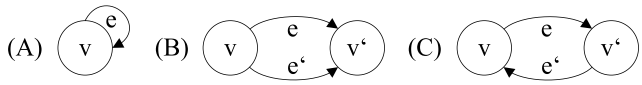

Figure 5.6: Slings, parallelism and anti-parallelism in a SIN.

Each vertex v may have several incoming and outgoing edges. The number of incoming edges of a vertex is denoted as *incoming degree*, whereas the number of its outgoing edges is denoted as *outgoing degree*. The *total degree* of a vertex corresponds to the sum of its incoming and outgoing degrees. Vertices having no incoming edges are denoted as *unreferenced*. In turn, vertices without outgoing edges are called *non-referencing*. Finally, vertices being both unreferenced and non-referencing are called *isolated* [181]. Based on these considerations, we define the degree of a vertex in a SIN as follows:

Definition 15 (**Degree of a Vertex in a SIN**). Given a vertex $v \in V$, the number of incoming and outgoing edges is denoted as *degree* of v. The *incoming degree* of v, denoted as $deg^-(v)$, corresponds to $|E^-(v)| = |\{e = (x, y) \in E \mid y = v\}|$. The *outgoing degree* of v, denoted as $deg^+(v)$, is defined as $|E^+(v)| = |\{e = (x, y) \in E \mid x = v\}|$. Finally, the *total degree* of v, denoted as $deg(v)$, is defined as $|E(v)| = |E^-(v)| + |E^+(v)|$.

Vertices directly linking to a vertex v are called *internal neighborhood* of v, whereas vertices referenced by another vertex v are denoted as *external neighborhood* of v. The

[2] Note that E is a *multiset* of edges and each edge e comprises a set of *properties* $P(e)$ (cf. Definition 17).

total neighborhood then corresponds to the union of internal and external neighborhood. Based on this, we define the neighborhood of a vertex in a SIN as follows:

Definition 16 (Neighborhood of a Vertex in a SIN). Given a vertex $v \in V$, the *internal neighborhood* of v, denoted $\Gamma^{-}(v)$, is the set of vertices $\{v' \mid (v', v) \in E\}$. Analogously, the *external neighborhood* of v, denoted $\Gamma^{+}(v)$, is the set of vertices $\{v' \mid (v, v') \in E\}$. Then, the *total neighborhood* of v is the union of the internal and external neighborhood, denoted $\Gamma(v) = \Gamma^{-}(v) \cup \Gamma^{+}(v)$.

For example, given two edges $e = (v, v'), e' = (v', v'') \in E$, and $v, v', v'' \in V$, we call v an internal neighbor of v' and v'' an external neighbor of v'. The total degree of v' is 2.

Note that we often refer to vertices as objects (e.g., information and process objects) and to edges as relationships. Next, we define *properties* for vertices and edges as follows:

Definition 17 (Properties of a Vertex and Edge in a SIN). Each vertex $v \in V$ and each edge $e \in E$ comprises a set of *properties* $P(v)$ and $P(e)$, respectively. Thereby, each $p \in P(v) \cup P(e)$ is a pair (key, val) with *key* being the unique name and *val* being the value of p. For short, $key(v)$ and $key(e)$ denote val.

As set out in Definition 14, function f_w assigns a *weight* to each edge e. This weight indicates the relevance of an edge and hence the strength of the relationship between two vertices. In a SIN, however, there may be multiple edges between vertices having different weights. To determine the overall strength between two vertices, therefore, we calculate the average weight of all edges between them. The *average weight* $avg_{\emptyset}(E')$ of a set of edges E' can be calculated as follows (cf. Formula 5.1):

$$avg_{\emptyset}(E') = \sum_{e' \in E'} \frac{f_w(e')}{|E'|} \tag{5.1}$$

In practice, certain edges may have to be weighted higher. As an example consider an "is similar to" relationship, which is usually more important than a "has same file format as" relationship. Therefore, we additionally introduce a significance function f_s with $f_s : E \to \mathbb{N}$ assigning to each edge $e \in E$ a significance value $f_s(e) \in \mathbb{N}$. The higher a significance value of an edge, the more important this edge will be. The *significance weight* $sig_{\Delta}(E')$ of a set of edges E' can be calculated as follows (cf. Formula 5.2):

$$sig_{\Delta}(E') = \sum_{e' \in E'} \frac{f_s(e') \; * \; f_w(e')}{\sum_{e'' \in E'} f_s(e'')} \tag{5.2}$$

5.3.2 Creation and Maintenance Phases

Generally, a SIN is created in six consecutive phases (cf. Figure 5.7) following a bottom-up approach; i.e., we start with the integration of business processes and process information that origin from heterogeneous sources such as databases, shared drives, enterprise portals, applications, process repositories, or enterprise information systems [7].

In Phase 1, business processes relevant for POIL need to be integrated.[3] Thereby, relevancy depends on which processes have to be supported by POIL. For this purpose, all relevant *process objects* (e.g., tasks, events, gateways, data objects, pools, lanes, sequence flows, message flows, or associations) are identified. In turn, the resulting objects are then used to create the SIN's first stage of expansion. In Phase 2, relevant process information (e.g., e-mails, office files, manuals, templates, forms, checklists, or guidelines) is added to the SIN; i.e., the already existing SIN is extended by adding *information objects* of different granularity levels, ranging from fine-grained information (e.g., database tuple) to coarse-grained one (e.g., multi-page office document).

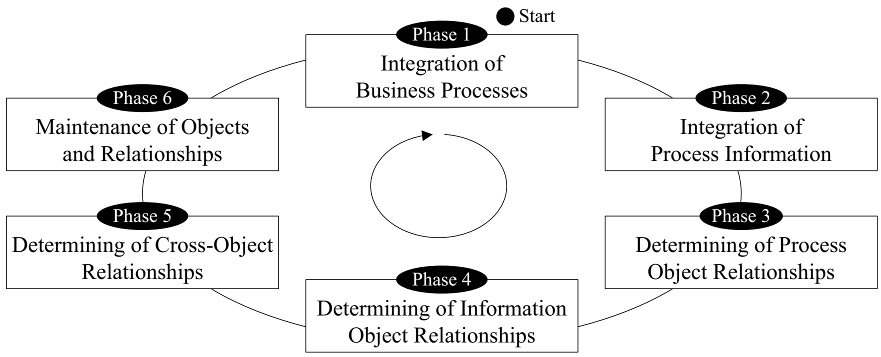

Figure 5.7: Creation and maintenance phases of the SIN.

In Phase 3, the relationships between process objects are identified; i.e., process objects such as sequence flows, associations, or message flows are transformed into *process object relationships*. In Phase 4, the *information object relationships* between the information objects of a SIN are discovered. Explicit relationships, like hyperlinks in documents, are discovered first. Then, algorithms from the fields of data mining, text mining, pattern-matching, and machine learning are applied to discover implicit relationships as well [182, 183]. In Phase 5, *cross-object relationships* between information and process objects are identified. For this purpose similar algorithms as used in Phases 3 and 4 are applied. In addition, pre-defined business rules (e.g., conditional constraints, derivations, or process rules) are used to detect further relationships [182].

Finally, in Phase 6, the SIN is maintained. This phase deals with the continuous integration as well as analysis of information and process objects and their relationships (cf. Section 5.4). Note that SIN maintenance constitutes a prerequisite for providing relevant information objects to knowledge workers and decision makers.

[3]The business processes to be integrated must be explicitly specified, e.g., using a process modeling language such as BPMN or EPC. Only such an explicit process description allows us to automatically transform a process schema and the corresponding process instances into process objects. The transformation algorithms we developed and use in this context (e.g., Signavio ContentProvider, Signavio AnalyzerTask, Activiti AnalyzerTask) can be found at http://sf.net/directory/?q=nipro.

In the following, we describe each phase of the described creation and maintenance process in detail. We illustrate the individual phases along the running example introduced in Section 4.2; i.e., the procurement of drugs in a hospital (cf. Figure 4.2).

Phase 1: Integration of Business Processes. In a first step, the process schemas (e.g., the procurement of drugs in a hospital) relevant for POIL are integrated. This means that all relevant process schema elements such as tasks, events, data objects, roles, sequence flows, message flows, associations, or gateways are identified and then used to create the SIN's first stage of expansion. In a second step, existing process instances (e.g., the procurement of drugs of specific patients) of the integrated process schemas are included as well. Besides the process elements themselves (both from the process schema and the process instances), corresponding metadata such as author, creation date, deadline, or modification date are also considered and associated with the process elements. Some metadata is automatically available (e.g., creation date, modification date, *uniform resource locator* (URL), unique identifier), whereas other has to be defined manually (e.g., deadline, project milestone, temporal process constraint, quality gate).

Figure 5.8 shows the SIN after Phase 1 for the first three process tasks of our use case as introduced in Section 4.2 (only the process schema is considered). The main task of Phase 1 is to transform business process elements into process objects.

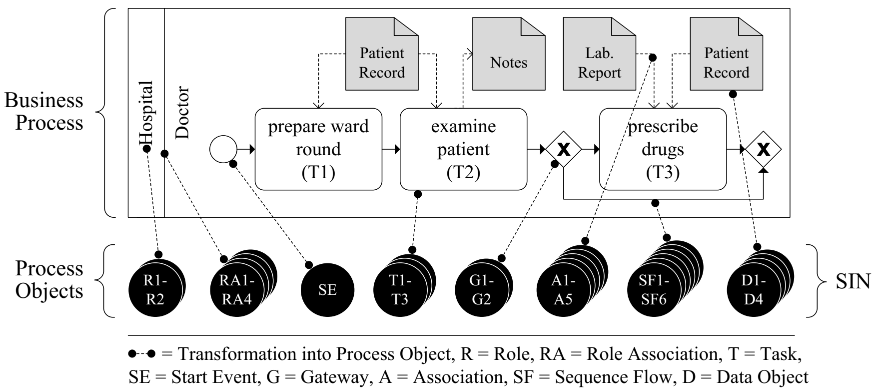

Figure 5.8: Phase 1: integration of business processes.

Note that the implemented algorithms[4] are published as open-source plugins for the iQser GIN server. The main steps of the algorithms are similar to the extract, transform and load steps in data warehousing or business intelligence [184].

Phase 2: Integration of Process Information. The second phase deals with the integration of process information schemas, process information instances, and abstract process information into the SIN (cf. Section 4.2). Only process information from data sources connected by the *integration component* (cf. Section 5.2.1) may be integrated.

[4]The full implementation of the algorithms can be found at http://sf.net/projects/signaviocontent (for process schemas) and at http://sf.net/projects/activiticontent (for process instances).

The already existing SIN, which resulted from Phase 1, is extended by information objects of different granularity levels, ranging from fine-granular information (e.g., database tuple, single-page office document) to coarse-granular information (e.g., database table, multi-page office document). Like in Phase 1, metadata is also added such as author, creation date, modification date, number of characters, file format, file size, or revision number. More specifically, metadata is attached to the information objects.

Figure 5.9 shows the resulting SIN for the procurement of drugs in a hospital (cf. Section 4.2) after Phase 2. At this stage, a SIN may already include up to hundreds or thousands of both information and process objects.

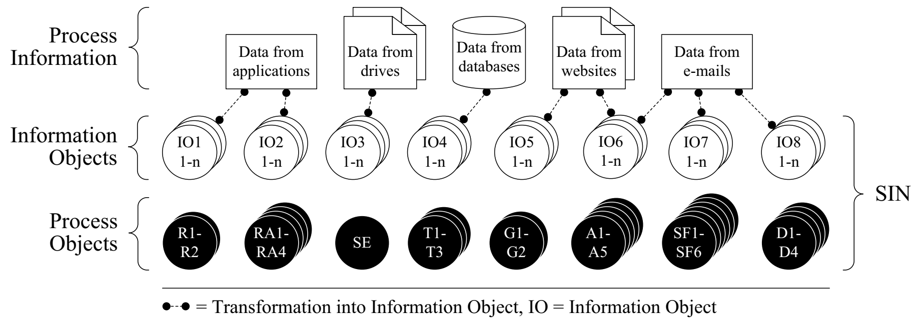

Figure 5.9: Phase 2: integration of process information.

As a prerequisite to transform process information into information objects, process information to be integrated must be electronically available and accessible. The implemented algorithms[5] are part of our prototypes in Chapter 8.

Phase 3: Determining of Process Object Relationships. The third phase deals with the identification of relationships between process objects. For example, if process schemas are modeled in terms of BPMN, process objects of types *sequence flow*, *association*, *role association*, and *message flow* will be first transformed into relationships (i.e., edges between process objects), and then removed from the SIN (cf. Figure 5.10).

Figure 5.10 shows the SIN after completing Phase 3. As explained, process objects (i.e., sequence flows SF1-6, role associations RA1-4, associations A1-5) have been transformed into process object relationships, and are then removed from the SIN.

Any relationship corresponds to an edge between two vertices. In turn, each edge is labeled with a *relationship reason* and a relationship relevance, called *relationship weight* (cf. Section 5.3.1). Especially, the labeling of relationships allows supporting different scenarios such as "find experts" using the relationship reason "has same author as" or the scenario "find related information" using a "is similar to" or "is used after" relationship. When completing Phase 3, all processes relevant for POIL are present in the SIN.

[5] The full implementation of the algorithms can be found at http://sf.net/projects/iqserwebplugin (for websites) and at http://sf.net/projects/iqserfileplugin (for files).

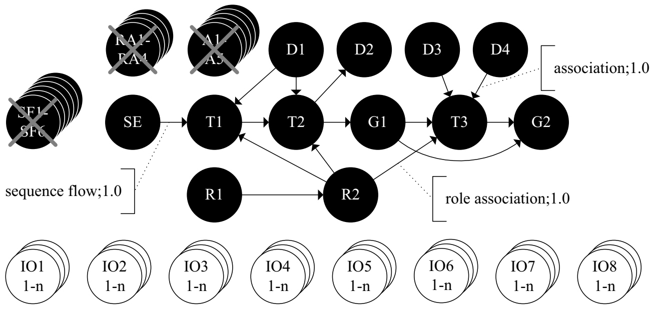

Figure 5.10: Phase 3: determining process object relationships.

Phase 4: Determining of Information Object Relationships. The fourth phase deals with the discovery of the relationships existing between information objects. Explicit relationships, like hyperlinks or foreign key relationships, are discovered in the first step. In the second step, algorithms from the fields of data mining, text mining (e.g., text preprocessing, linguistic preprocessing, vector space model, clustering, classification, and information extraction) [183], pattern-matching, and machine learning (e.g., supervised learning, unsupervised learning, reinforcement learning, transduction) are applied [182].

Figure 5.11 shows the SIN we obtain after Phase 4. Note that in practice, the actual number of relationships between information objects will be by far higher.

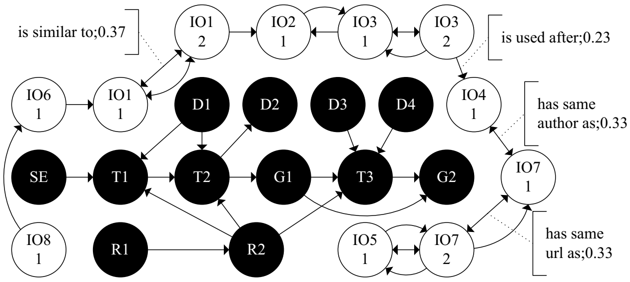

Figure 5.11: Phase 4: determining information object relationships.

More precisely, we apply syntax- and semantic-based algorithms[6] to discover the meaning of information objects, e.g., such as (inverse) term frequency algorithms, link popularity algorithms, or utilization context algorithms. In this thesis, we use a commercial semantic middleware platform implementing these algorithms [171].

As examples of discovered relationships between information objects consider metadata matches (e.g., author, creation date, keyword, file format, URL), text similarities, utilization context similarities, and cluster similarities. Like in Phase 3, relationships are represented by edges that are labeled with relationship reasons and weights. Like for process objects, there may be multiple edges between information objects.

[6]The implementation of the algorithms can be found, e.g., at http://sf.net/projects/equalityanalyze and at http://sf.net/projects/ratinganalyzer.

Phase 5: Determining of Cross-Object Relationships. The fifth phase deals with the analysis of the relationships existing between process and information objects. For this purpose, the algorithms from Phases 3 and 4 are reapplied. In addition, metadata matcher and pre-defined business rules are used to detect further relationships.

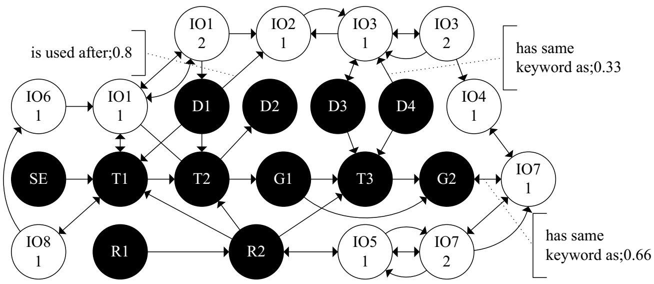

Figure 5.12: Phase 5: determining cross-object relationships.

Figure 5.12 shows the SIN that results after completing Phase 5. As illustrated, there now exist cross-object relationships between process and information objects.

As examples of discovered relationships between information and process objects consider utilization context similarities (e.g., a process information is used after clicking on a process task), text similarities (e.g., a description of a process task is similar to a manual), or metadata-matches (e.g., a process model has the same author as a guideline).

Phase 6: Maintenance of Objects and Relationships. The final phase deals with the continuous integration of process information and business processes, and the analysis of process and information objects [13]. The most important tasks (cf. Table 5.2) related to this phase include the discovery of relationships between recently added and already existing objects, and validation checks in respect to existing objects and relationships.

#	Maintenance Tasks
T1	Add object (e.g., add a manual or guideline).
T2	Update object (e.g., modify a manual).
T3	Delete object (e.g., delete a guideline).
T4	Add relationship (e.g., two manuals have the same author).
T5	Update relationship (e.g., two manuals become more similar).
T6	Delete relationship (e.g., two guidelines no longer have the same author).

Table 5.2: Phase 6: maintenance tasks.

In summary, Phase 6 deals with the repeated execution of Phases 1-5 and additional tasks (e.g., to decide about the consistence of objects with respect to maintenance).

In order to enable continuous integration, the semantic layer supports a *push* and *pull principle* (cf. Section 5.4). Regarding the push principle, data sources provide notifications about changed business processes and process information respectively. In turn, the pull principle is based on time-based scheduled integration jobs. Continuous analysis further includes cleansing of outdated or no longer available process and information objects. Thereby, process participants may manually modify the SIN; i.e., they can rate individual information objects or create (public and private) relationships.

Figure 5.13 shows the SIN after Phase 6 based on our running example from Section 4.2. Overall, we consider two running process instances. In the first one, medical orders are currently created. In the second one, a patient is examined by the doctor.

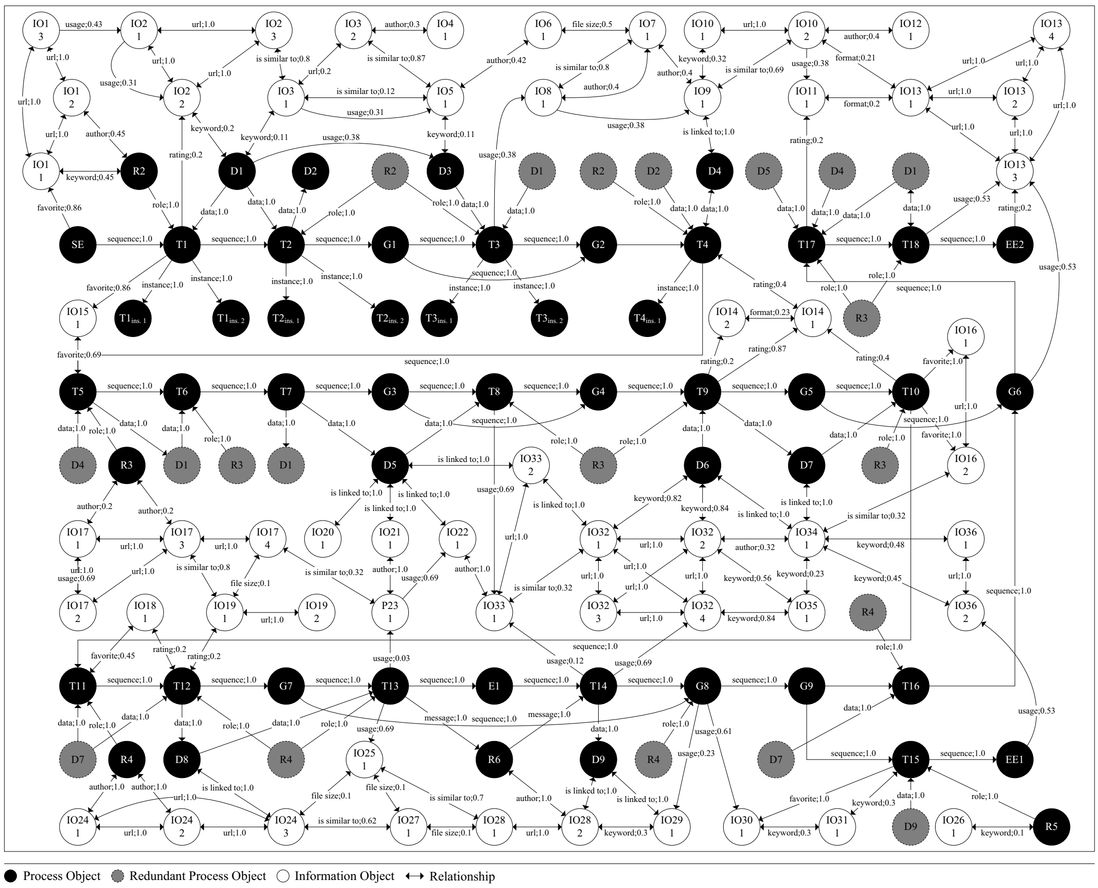

Figure 5.13: Exemplary SIN of our running example.

Note that we limit the illustration of the SIN to process tasks that are performed when no drugs have to be ordered. Information objects are visualized by white circles, whereas process objects are illustrated by black ones. For a better understanding, we also depict some process objects twice. The latter can be recognized by grey circles. Note that relationships reasons of edges are partially abbreviated. Also, for simplicity, role hierarchies are not shown in the exemplary SIN (cf. Figure 5.13).

5.4 Maintaining the Semantic Information Network

In order to provide that process information to process participants fitting best to their demands, respective information and process objects in a SIN must be complete, consistent and up-to-date; i.e., a SIN must be continuously maintained (cf. Phase 6).

SIN maintenance constitutes a prerequisite for providing required process information (i.e., information objects) to process participants. For example, objects may be integrated (e.g., new guidelines are created), updated (e.g., a process task is modified), or deleted (e.g., a checklist is no longer valid). Likewise, relationships may be established (e.g., two documents have the same author or are stored in the same file format), updated (e.g., two forms become more similar to each other), or deleted (e.g., two documents have no longer the same author). On one hand, such changes may happen outside the SIN (e.g., a checklist may have been changed in a database). We then talk about *exogenous changes*. On the other, changes may occur inside the SIN (e.g., a guideline being outdated). They are then denoted as *endogenous changes*. Both exogenous and endogenous changes must be properly handled by the SIN (cf. Figure 5.14).

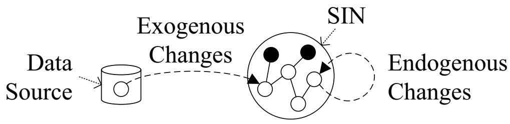

Figure 5.14: Exogenous and endogenous changes.

In the following, we propose an approach for maintaining SINs. Specifically, we show how SINs evolve over time. In this context, we identify characteristics of object and relationship properties as well as their influence on SIN maintenance. Finally, we introduce three algorithms dealing with exogenous and endogenous SIN changes.

5.4.1 Evolution

SIN evolution is driven by exogenous as well as endogenous changes (cf. Figure 5.14). Thereby, we further distinguish between evolution in depth and breadth [185]. *Depth* is defined by the size of all property values of a SIN; i.e., the amount of information (e.g., the information stored within all objects in the SIN). *Breadth* is defined as the number of relationships in a SIN; i.e., the cardinality of the set of edges.

Depth may be increased by adding objects (e.g., new documents on a shared drive), adding properties (e.g., adding keywords to an existing document), or updating property values (e.g., describing a property in greater detail) (cf. Figure 5.15A).

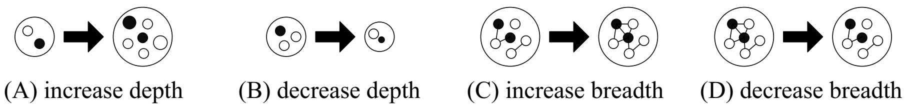

Figure 5.15: SIN evolution in depth and breadth.

In turn, deleting objects and properties decreases the depth of a SIN (cf. Figure 5.15B). Note that updates of property values might decrease depth as well. Breadth can be increased by adding relationships (e.g., a new link between two objects) (cf. Figure 5.15C). By contrast, deleting relationships (e.g., two objects no longer have the same author) decreases breadth (cf. Figure 5.15D). Hence, depth and breadth are indicators for the cost of performing maintenance tasks such as adding an object or relationship to a SIN.

We formalize *actions* changing objects and relationships in a SIN, e.g., by increasing or decreasing the depth and breadth of the SIN. Next, we define an *action* as follows:

Definition 18 (**Action changing a SIN**). An *action* changes a SIN. Each *action* a has a set of parameters $PA(a)$, where each $pa \in PA(a)$ is a pair (key, val). We call key the unique name and val the value of pa and write $key(a)$ to denote val. A parameter pa is either mandatory or optional. If pa with key key is mandatory, then, for each action a, there exists a value val_a such that $(key, val_a) \in PA(a)$.

Mandatory parameters of an action are an identifier "uri" (e.g., a number) and the function "func" (e.g., add, update and delete) to be executed. Actions are triggered by exogenous or endogenous changes (cf. Figure 5.16), e.g., when a document on a shared drive is deleted (*exogenous change*) or becomes outdated (*endogenous change*).

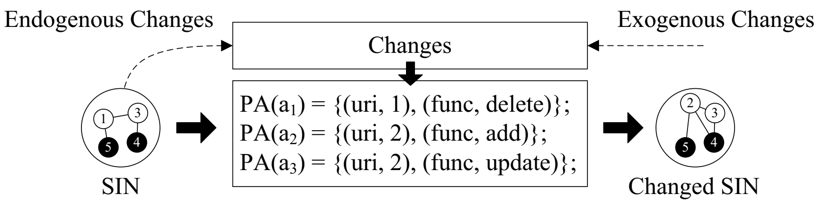

Figure 5.16: Actions changing a SIN.

As example consider an engineer conducting a review of product requirements documented in functional specifications. The goal is to improve respective specifications as well as to approve them. Due to a revision of the review process, an employee from the quality management department replaced an outdated review template. Thus, an action a_1 was triggered with $uri(a_1)$ = "H:/templates/review-v1.xls" and $func(a_1)$ = "delete". Thereby, another action a_2 was triggered with $uri(a_2)$ = "H:/templates/review-v2.xls" and $func(a_2)$ = "add". However, based on new guidelines the engineer noticed that the template was incomplete (e.g., a required question was missing). Therefore, the engineer adapts the template. Thus, another action a_3 is triggered with $uri(a_3)$ = "H:/templates/review-v2.xls" and $func(a_3)$ = "update".

5.4.2 Property Classification

When maintaining a SIN, not only the object-relationship level must be considered, but the properties of objects and relationships as well (cf. Section 5.3.1). For example, if the “title” of a document has changed, it is not necessary to overwrite the entire object. Instead, only the relevant (i.e., changed) parts need to be updated. While certain properties evolve over time (e.g., “file size”), others remain unchanged (e.g., “uri”). Respective issues have been addressed by the maintenance algorithms we developed; i.e., by focusing only on those properties relevant for a particular task. To consider this, we categorize properties into *existence* and *mutability* and define them as follows:

Definition 19 (**Existence of Properties**). *Existence* indicates whether a property is *mandatory* or *optional*, where a property p with key key is mandatory for the vertices V (edges E) of a SIN if, for each $v \in V$ ($e \in E$), there exists a value val_v (val_e) such that $(key, val_v) \in P(v)$ ($(key, val_e) \in P(e)$) and it is optional otherwise.

Definition 20 (**Mutability of Properties**). *Mutability* indicates whether a property's value is *dynamic* or *static*, where p is dynamic if val in (key, val) can change over time and it is static otherwise (i.e., val does not change).

For example, mandatory properties of objects are a unique identifier “uri”, a data source “source”, a creation date “cdate”, a modification date “mdate”, and a content “cont” (i.e., the raw full text). Both categories (i.e., existence and mutability) may be combined into a matrix comprising four quadrants to which we assign the properties of both objects and relationships (cf. Figure 5.17).

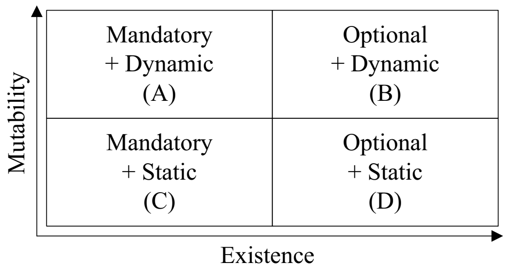

Figure 5.17: Property classification.

We illustrate the assignment of individual properties to different quadrants (i.e., A-D) of the property classification with examples: First, *mandatory + dynamic* properties apply in every SIN. As example consider the modification date “mdate” that changes whenever an object or relationship is updated. Other mandatory and dynamic properties of an object are the content “cont” or the total degree “deg”, which may be change over time as well (e.g., when new relationships are discovered). Second, *optional + dynamic* properties exist in a SIN. For example, the “title” of a document may change over time. However, not all file types (e.g., a text file) have a “title”. Therefore, the property “title” of a document is optional and dynamic. Third, *mandatory + static* properties exist in a SIN. For example, an identifier “uri” or a creation date “cdate” exists for all objects and relationships. Thus, these properties are mandatory. Since these properties do not change over time, they can be considered as static as well. Finally, *optional + static* properties apply in a SIN. For example, if a property does not change over time and not exist for every object or relationship it is both optional and static, e.g., the "file type".

Based on the property classification, we infer the following for adding, updating and deleting elements of a SIN: One must ensure that mandatory as well as static properties are given as a minimum requirement when adding elements (cf. Figure 5.18A). Note that the grey color in Figure 5.18 indicates affected quadrants for each function.

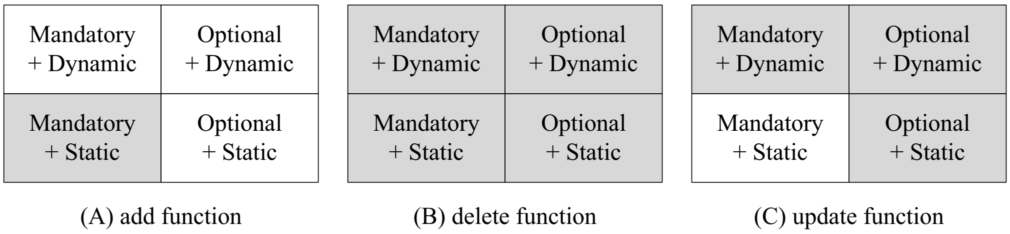

Figure 5.18: Property classification and functions.

When deleting objects or relationships in a SIN, properties within all quadrants must be considered (cf. Figure 5.18B). When executing updates, only properties which are not assigned to the mandatory and static quadrant must be considered (cf. Figure 5.18C). We now specify the three functions *add*, *delete* and *update* (see below) that are used by our maintenance algorithms. In the following, we introduce the latter as well.

5.4.3 Maintenance Algorithms

The *add* function inserts a vertex v_{add} and its properties in a SIN. Further, it determines which relationships exist between v_{add} and the already existing vertices. As mentioned, properties, being both mandatory and static, are the minimum input for the function.

Function add(SIN, v_{add})

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN,
v_{add}: vertex to be added and its properties $P(v_{add})$;
Output: SIN is updated;

begin
  $V := V \cup \{v_{add}\}$;
  foreach $v \in V$ **do**
    if $uri(v) \neq uri(v_{add})$ **then**
      $E := E \cup \{$new edge/s between v and $v_{add}\}$;

The *delete* function removes a vertex v_{del} and its properties from a SIN including its relationships with neighbored vertices. Note that all quadrants of the property classification are relevant for the delete function (cf. Figure 5.18B).

Function delete(SIN, v_{del})

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN,
v_{del}: vertex to be deleted and its properties $P(v_{del})$;
Output: SIN is updated;

begin
  $v :=$ get $v \in V$ so that $uri(v) = uri(v_{del})$;
  $E := E \setminus \{(v, \gamma), (\gamma, v) \mid \gamma \in \Gamma(v)\}$;
  $V := V \setminus \{v\}$;

The *update* function takes a vertex v_{upd} as input, which is used to update the vertex v in the SIN that is identified by the same uri as v_{upd}. The function also adds, deletes and updates relationships between the updated vertex v and existing vertices.

Function update(SIN, v_{upd})

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN,
v_{upd}: vertex used to update the SIN and its properties $P(v_{upd})$;
Output: SIN is updated;

begin
  $P(v_{upd}) := \{p \in P(v_{upd}) \mid p$ is not mandatory + static$\}$;
  $v :=$ get $v \in V$ so that $uri(v) = uri(v_{upd})$;
  foreach $(key, val) \in P(v)$ **do**
    if $(key, val_{upd}) \in P(v_{upd})$ **then**
      $val := val_{upd}$;
      $P(v_{upd}) := P(v_{\mathrm{upd}}) \setminus \{(key, val_{upd})\}$;
    else
      $P(v) := P(v) \setminus \{(key, val)\}$;
  $P(v) := P(v) \cup P(v_{upd})$;
  foreach $v' \in V$ **do**
    if $v' \in \Gamma(v)$ **then**
      $E :=$ update edge/s between v' and v;
      $E := E \setminus \{$obsolete edge/s between v' and $v\}$;
    if $uri(v') \neq uri(v)$ **then**
      $E := E \cup \{$new edge/s between v' and $v\}$;

Based on these functions, we propose three algorithms for maintaining SINs. The maintenance is based on two main principles: (1) *push principle* and (2) *pull principle.*

Regarding the *push principle*, the data source pushes process information and business processes automatically to the SIN when they are added, updated or deleted in the data source. Regarding exogenous changes, however, as a prerequisite the data source must be able to send notifications if process information or business processes are changed. Regarding endogenous changes, the prerequisite is that the SIN detects changes automatically and triggers respective actions (cf. Definition 18).

Regarding the *pull principle*, the SIN gathers process information and business processes from a data source. Such a maintenance process is triggered by time-based schedulers; i.e., the SIN is maintained at certain points in time. The pull principle is used for data sources not capable of sending change notifications to the SIN.

For each of these two principles, we introduce a corresponding algorithm. As a prerequisite for both algorithms, the SIN must have access to underlying data sources. In case of exogenous changes, the SIN transforms process information and business processes into a uniform format. In case of endogenous changes no transformation is necessary.

Push Algorithm. The *push algorithm* deals with changes of a SIN according to the push principle, e.g., a policy may no longer be valid in 2015 and the corresponding SIN object has to be maintained accordingly. Thus, SIN maintenance is triggered by an action applied to the SIN by the push algorithm.

Algorithm 1: Push Algorithm.

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN, a an action;
Output: SIN is updated;

```
begin
    switch func(a) do
        case add
            v := create a vertex and its properties from
                the data source affected by the uri of a;
            add(SIN, v);
            break;
        case update
            v := create a vertex and its properties from
                the data source affected by the uri of a;
            update(SIN, v);
            break;
        case delete
            v := get v ∈ V so that uri(v) = uri(a);
            delete(SIN, v);
```

The push algorithm works as follows: In the add and update case, we create a vertex v and its properties from the data source affected by the action a (i.e., based on the "uri" of the action). After that, we call the corresponding add or update function. In the delete case, we identify the corresponding vertex $v \in V$ based on the "uri" of the action and call the according delete function.

Pull Algorithm. The *pull algorithm* deals with changes of a SIN based on the pull principle; i.e., data has changed in the data source and needs to be gathered by the SIN. For example, documents on a shared drive are updated and, therefore, respective changes must be made in the SIN. The maintenance of the SIN is triggered by a scheduler.

Algorithm 2: Pull Algorithm.

```
Input: SIN = (V, E, L, W, f_l, f_w) a SIN, ds the data source;
Output: SIN is updated;
begin
    V_ds := create a set of vertices and their properties from the data source ds;
    foreach v ∈ V so that source(v) = ds do
        v_ds := get v_ds ∈ V_ds so that uri(v_ds) = uri(v);
        if v_ds ≠ null then
            if v_ds is newer than v then
                update(SIN, v_ds);
            V_ds := V_ds \ {v_ds};
        else
            delete(SIN, v);
    foreach v_ds ∈ V_ds do
        add(SIN, v_ds);
```

The pull algorithm works as follows: First, we create a set of vertices V_{ds} from a data source ds. Then, for each vertex $v \in V$ that was created from ds (property "source"), we check whether a corresponding vertex $v_{ds} \in V_{ds}$ exists. If this is the case, we further check whether v_{ds} is newer than v (e.g., by comparing the creation or modification dates). If v is outdated, it is updated with the properties of v_{ds} by calling the update function. Then, v_{ds} is removed from V_{ds}. If no corresponding vertex exists in the data source, we delete vertex v in the SIN using the delete function. Finally, we add each remaining vertex $v_{ds} \in V_{ds}$ from the data source to the SIN by calling the add function. Hence, the pull algorithm allows maintaining the SIN at a certain point in time. Note that this procedure has to be repeated for each data source of the SIN.

Partial-Pull Algorithm. In practice, a SIN may comprise a large number of objects and relationships. Maintaining the SIN based on the pull principle, therefore, can be a time-consuming task. In a specific work context, however, a user might only be interested in a selected part of the SIN. During a review, for example, review templates, existing reviews, or results of a real-time evaluation (e.g., prioritization of projects in a workshop) are of great importance, while checklists and best practices for performing effective project management are less interesting. Thus, it is sufficient to maintain only those objects and relationships relevant for the user when querying the SIN. To reflect this, we introduce another principle, called the *partial-pull principle*, where the SIN gathers only process information and business processes from data sources as requested by a user.

Regarding the partial-pull principle, we introduce a third algorithm (i.e., the *partial-pull algorithm*) as a lightweight version of the pull algorithm. It does not maintain the entire SIN, but only those parts relevant for a given request.

Algorithm 3: Partial-Pull Algorithm.

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN, *req* the request to a SIN;
Output: SIN is partially updated, V_{req} contains the requested vertices;

begin
  V_{ds} := create a set of vertices from the data sources affected by *req*;
  foreach $v \in V$ *affected by req* **do**
    v_{ds} := get $v_{ds} \in V_{ds}$ so that $uri(v_{ds}) = uri(v)$;
    if $v_{ds} \neq null$ **then**
      if v_{ds} *is newer than* v **then**
        $update(SIN, v_{ds})$;
        v_{upd} := get $v' \in V$ so that $uri(v') = uri(v_{ds})$;
        $V_{req} := V_{req} \cup \{v_{upd}\}$;
      else
        $V_{req} := V_{req} \cup \{v\}$;
    else
      $delete(SIN, v)$;

The partial-pull algorithm works as follows: First, we create a set of vertices V_{ds} from the data sources affected by the user request *req*. Then, for each vertex $v \in V$ affected by *req* we retrieve the corresponding vertex v_{ds} from the affected data sources. Thus, if a corresponding vertex v_{ds} is in the data source, we check whether v_{ds} is newer than v (e.g., by comparing the creation or modification dates). If v is outdated, it is updated with v_{ds} by calling the update function. If there is no corresponding vertex in the data source, we delete vertex v in the SIN by calling the delete function. The partial-pull algorithm allows maintaining parts of a SIN based on a request and ensures that all requested objects are synchronized with affected data sources.

As opposed to the other principles, the partial-pull principle is completely *user-driven* as it is solely triggered by a user request (e.g., a search). In turn, the push and pull principle are *machine-driven*, e.g., triggered through notifications from schedulers.

Validation (Maintenance Algorithms). In order to demonstrate and validate the feasibility and applicability of the maintenance algorithms, we conducted a case study in the automotive domain. This section only provides a short summary of the case study results. The entire results can be found in Section 9.3. In the case study, we have shown that automatic maintenance of SINs with regard to both exogenous and endogenous changes is feasible with acceptable costs. Furthermore, we investigated the effect of depth and breadth on the runtime of the proposed algorithms. The algorithms performed satisfactorily in terms of adding, updating and deleting objects. The cost of detecting relationships, however, varies widely when using expensive linguistic or statistical algorithms. Moreover, the case study results have shown that it is crucial to provide up-to-date, integrated and homogeneous views on process information during business process execution. The empirical validation confirms that there is a high demand for a single point of access to process information in knowledge-intensive business processes.

5.5 On the Relevance of Process Information

We have already shown that process participants spend considerable efforts on handling process information. A challenging task in this context is to identify relevant process information. In practice, for example, there exist many specific review templates for review processes. Depending on the concrete process, therefore, specific review templates are relevant and hence need to be delivered to the process participants. In POIL, the SIN provides the basis for this task. However, specific techniques and algorithms are needed to determine relevant process information (cf. Research Question 4); i.e., currently needed information objects in a SIN dependent on the work context (cf. Figure 5.19). The reason for this is that the SIN just identifies objects linked to each other for some reasons, but does not consider additional influence factors (e.g., the work context, the relationship structure of a SIN, or user ratings of information objects).

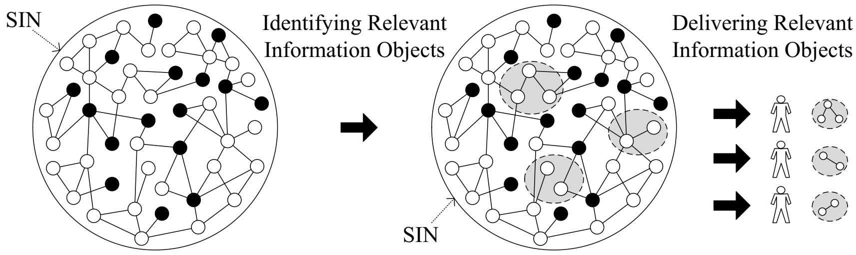

Figure 5.19: Delivering relevant information objects.

In the following, we introduce algorithms for identifying relevant information objects in a SIN. The first one determines the *link popularity* of information objects based on the relationship structure of a SIN (cf. Section 5.5.1). The second one calculates the *rate popularity* of information objects based on user ratings (cf. Section 5.5.2). Note that these algorithms may be used independently, but also in combination with each other.

5.5.1 Link Popularity Algorithm

In enterprises, process information is usually not explicitly linked to other process information or business processes. Therefore, it is not possible to take advantage of a rich relationship structure within an enterprise environment. Instead, process information is implicitly linked to other process information and business processes, e.g., dealing with the same topic or used in the same work context. A SIN makes such implicit relationships explicit by means of its edges. The relationship structure enables us to apply algorithms to identify strongly linked and hence popular objects. However, as we will show, existing link popularity algorithms are not sufficient in this context (cf. Figure 5.20). Thus, we extend them and introduce the *SIN link popularity* (LP) *algorithm*, which allows determining the link popularity of information objects in a SIN.

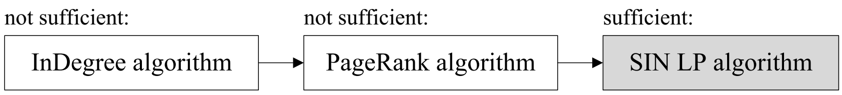

Figure 5.20: Link popularity algorithms.

Basic to any link popularity algorithm is an *InDegree algorithm* [186], which measures the *link popularity* $LP(v)$ of an information object v taking its number of incoming edges into account (cf. Formula 5.3). The higher the number of incoming edges is, the greater the popularity of an information object becomes:

$$LP(v) = deg^{-}(v) \tag{5.3}$$

In a SIN, the InDegree algorithm is not really helpful since certain relationships might be more valuable than others. In turn, this issue is picked up by the *PageRank algorithm* [187]: Relationships originating from information objects of high quality are considered as being more valuable than relationships originating from information objects of low quality. Accordingly, the link popularity $LP(v)$ of an information object v is calculated as follows (with d corresponding to a damping factor ranging from 0 to 1):

$$LP(v) = (1 - d) + d \sum_{w \in \Gamma^{-}(v)} \frac{LP(w)}{deg^{+}(w)} \tag{5.4}$$

Like the InDegree algorithm, the conventional PageRank algorithm, which was originally designed for the *world wide web* (WWW), is not applicable to a SIN since it only considers single relationships (cf. Formula 5.4). In a SIN, however, there are multiple, weighted and labeled relationships. Hence, we must extend the PageRank algorithm. First, we have to support multiple relationships:

$$LP(v) = (1 - d) + d \sum_{w \in \Gamma^{-}(v)} |\{e = (w, v) \in E\}| * \frac{LP(w)}{deg^{+}(w)} \tag{5.5}$$

In order to also support weighted relationships, we extend Formula 5.5 and include the average weighting function $avg_{ø}(E')$ (cf. Formula 5.1):

$$LP(v) = (1 - d) + d \sum_{w \in \Gamma^{-}(v)} avg_{ø}(\{e = (w, v) \in E\}) * |\{e = (w, v) \in E\}| * \frac{LP(w)}{deg^{+}(w)} \tag{5.6}$$

Note that Formula 5.6 only deals with equally weighted relationships. To finally support variously weighted relationships, we must extend it by the significance weighting function $sig_{\Delta}(E')$ (cf. Formula 5.2):

$$LP(v) = (1-d) + d \sum_{w \in \Gamma^{-}(v)} sig_{\Delta}(\{e = (w,v) \in E\}) * |\{e = (w,v) \in E\}| * \frac{LP(w)}{deg^{+}(w)} \quad (5.7)$$

Based on Formula 5.7, it becomes possible to determine the link popularity of SIN information objects. Note that this corresponds to the solution of a system of equations. In our approach, we use an approximate, iterative calculation of the link popularity; i.e., we assign an initial $LP(v) = init$ to each information object v. The link popularity $LP(v)$ is then iteratively determined for each information object v as follows (let $iter$ be the number of iterations and d the damping factor):

Algorithm 4: SIN LP Algorithm.

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN, d the damping factor, $iter$ the number of iterations, $init$ the initial link popularity;
Output: $LP(v)$ for each $v \in V$;

begin
  foreach $v \in V$ **do** $LP(v) = init$;
  foreach $e \in E$ **do** $f_s(e)$;
  for $i = 1$ **to** $iter$ **do**
    foreach $v \in V$ **do**
      $pop = 0$;
      foreach $w \in \Gamma^{-}(v)$ **do**
        $pop = pop + (sig_{\Delta}(\{e = (w,v) \in E\}) \; * \; |\{e = (w,v) \in E\}| \; * \; LP(w) \; / \; deg^{+}(w))$;
      $LP(v) = (1-d) + d \; * \; pop$;

To better understand the SIN LP algorithm[7], we apply it to an exemplary SIN comprising 23 information objects (cf. Figure 5.21). For the sake of simplicity we use bidirectional relationships (e.g., "is similar to", "file format") when visualizing the SIN.

In order to calculate the link popularity of grey information objects (cf. Figure 5.21), we use $sig_{\Delta}(E')$ and double weight "is similar to", "is linked to", "topic", and 'machine-learning" relationships. In addition, we apply the following settings: number of iterations $iter = 12$, damping factor $d = 0.5$, and initial link popularity value $init = 0.607$. The former two values are based on practical experiences, we gathered when applying the SIN LP algorithm. In turn, the initial link popularity value $init$ is calculated based on the average weight of all relationships in our exemplary SIN (cf. Figure 5.21).

[7] Our implementation can be found at http://sf.net/projects/linkinganalyzer.

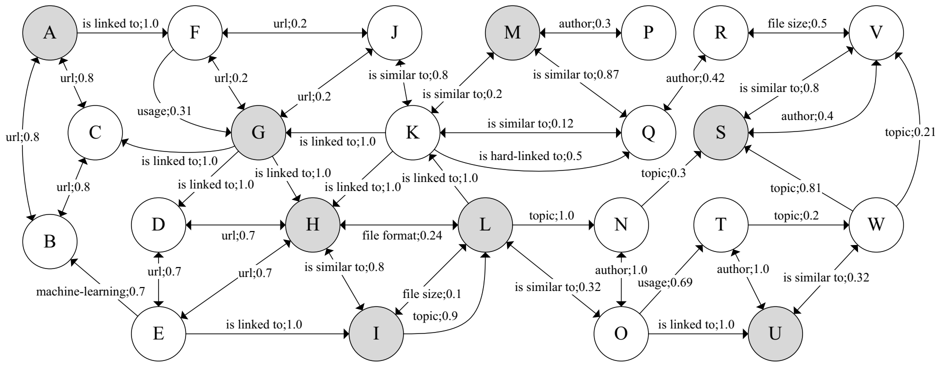

Figure 5.21: Exemplary SIN to determine the SIN link popularity.

When starting the algorithm, it can be observed that all information objects have the same link popularity; i.e., the initial link popularity value (cf. Table 5.3). Hence, we cannot make any distinction between information objects when ranking them. After a few iterations, the values slightly change. For example, it becomes obvious that information object A is more popular than information object G. The higher the number of iterations, the higher the precision of link popularity values will be.

#	$\boldsymbol{LP(v)}$ $iter = 0$	$\boldsymbol{LP(v)}$ $iter = 1$	$\boldsymbol{LP(v)}$ $iter = 2$	$\boldsymbol{LP(v)}$ $iter = 3$	$\boldsymbol{LP(v)}$ $iter = 4$	$\boldsymbol{LP(v)}$ $iter = 5$	$\boldsymbol{LP(v)}$ $iter = 12$
A	0.607	0.743	0.818	0.836	0.839	0.839	0.839
G	0.607	0.701	0.710	0.711	0.711	0.711	0.711
H	0.607	0.877	0.918	0.923	0.923	0.923	0.923
I	0.607	0.680	0.687	0.687	0.688	0.688	0.688
L	0.607	0.694	0.700	0.700	0.700	0.700	0.700
M	0.607	0.690	0.692	0.692	0.692	0.692	0.692
S	0.607	0.765	0.819	0.824	0.825	0.825	0.825
U	0.607	0.795	0.807	0.808	0.808	0.808	0.808

Table 5.3: Results obtained when applying the SIN LP algorithm to the example.

There exists a wide range of link popularity algorithms as well. Best known is the PageRank algorithm [187]. However, the idea to consider relationships being more valuable than others is picked up by other algorithms as well, e.g., the Hits algorithm [188] or the weighted PageRank algorithm [189]. An algorithm combining both PageRank and Hits is the Salsa algorithm [190]. Another evolution of the PageRank is provided by the Topic-Sensitive PageRank algorithm [191] and the Weighted PageRank algorithm [189]. The former additionally considers topics, whereas the latter considers the importance of links. However, all these algorithms have been originally developed for the WWW and cannot be applied to POIL without modification. Particularly, they do not allow dealing with the specific characteristics of a SIN. For example, relationships in a SIN are weighted and labeled, and multiple relationships exist as well (cf. Section 5.3).

Validation (SIN LP Algorithm). We use an automotive scenario to validate the applicability of the SIN LP algorithm. This section only gives a short summary of these results; the entire results can be found in Section 9.2. The empirical validation confirmed that most of the documents returned by the SIN LP algorithm are indeed relevant. Moreover, we showed that the link popularity is a good indicator for identifying relevant process information, especially since results of the SIN LP algorithm can be further refined for specific tasks by applying the SIN LP algorithm to only specific parts of a SIN (e.g., to a specific process, corresponding instances, or related information objects).

In summary, the SIN LP algorithm allows determining the link popularity of objects based on the relationship structure of the SIN in an iterative way.

5.5.2 Rate Popularity Algorithm

We introduce another algorithm that allows determining the rate popularity of process information based on user ratings. In enterprises, existing portals often allow users to rate the quality of process information, e.g., by means of "like buttons" or "five stars ratings". The set of ratings R can then be used to determine the *rate popularity* $RP(v)$ of an information object v. However, ranking information objects based on user ratings is a non-trivial task. First, we show that existing algorithms are not directly applicable to POIL. Second, we develop the *SIN rate popularity* (RP) *algorithm*, which allows determining the rate popularity of information objects in a SIN (cf. Figure 5.22).

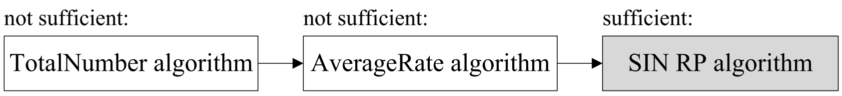

Figure 5.22: Rate popularity algorithms.

An approach to determine the rate popularity $RP(v)$ of an information object v is to rank information objects according to their *total number* of ratings (cf. Formula 5.8):

$$RP(v) = |R(v)| \tag{5.8}$$

Another approach is to determine the rate popularity $RP(v)$ based on the *average user rating* of an information object v (cf. Formula 5.9):

$$RP(v) = avg(R(v)) = \sum_{r \in R(v)} \frac{r}{|R(v)|} \tag{5.9}$$

Formulas 5.8 or 5.9 are not appropriate in the context of a SIN. Both formulas tend to prefer older information objects that have been available for a longer time; i.e., there has been more time for users to rate for these information objects. This shortcoming is rather problematic in enterprise environments with continuously emerging information objects. Moreover, the use of Formula 5.9 results in another problem: Assume that in a "five stars rating" there is an information object with an average rating of 4.8, which is based on hundreds of individual ratings. Additionally, assume that another information object is rated by one user with 5.0. The latter information object is then directly ranked on the first position. To avoid this, all ratings must be taken into account.

Thus, we calculate the rate popularity consistent with *Bayesian interpretation* (to evaluate the probability of the hypothesis) [192]. Formula 5.10 allows calculating the average rating $avg(R)$ of all information objects. Formula 5.11 then calculates the rate popularity $RP(v)$ of a single information object v, taking both the set of ratings R and the age of the information objects into account. Thus, we ensure that information objects with few, but favorable ratings are not ranked on first positions:

$$avg(R) = \sum_{v \in V} \frac{|R(v)| * avg(R(v))}{|R|} \tag{5.10}$$

$$RP(v) = \frac{\frac{\left(\frac{|R|}{|\{v \in V \,|\, R(v) > 0\}|} * avg(R)\right) + \left(|R(v)| * avg(R(v))\right)}{\frac{|R|}{|\{v \in V \,|\, R(v) > 0\}|} + |R(v)|}}{age(v)} \tag{5.11}$$

Based on Formulas 5.10 and 5.11, one can determine the rate popularity of information objects. The SIN RP algorithm[8] shows how the rate popularity value is calculated for each information object v taking the set of user ratings R into account:

Algorithm 5: SIN RP Algorithm.

Input: $SIN = (V, E, L, W, f_l, f_w)$ a SIN, R the set of ratings;
Output: $RP(v)$ for each $v \in V$ where $|R(v)| > 0$;

begin
  foreach $v \in V$ **do**
    if $|R(v)| > 0$ **then**
      $avg(R) \stackrel{+}{=} |R(v)| * avg(R(v)) \;/\; |R|$;
  foreach $v \in V$ **do**
    if $|R(v)| > 0$ **then**
      $pop = ((|R| \;/\; |\{v \in V \mid R(v) > 0\}| \; * avg(R)) + (|R(v)| * avg(R(v))))$;
      $pop = pop \;/\; (|R| \;/\; |\{v \in V \mid R(v) > 0\}| \; + |R(v)|)$;
      $RP(v) = pop \;/\; age(v)$;

To better understand the SIN RP algorithm, we compare it with other approaches, specifically with the TotalNumber and the AverageRate algorithm. For this purpose,

[8]Our implementation can be found at http://sf.net/projects/ratinganalyzer.

we use available user ratings of information objects that we have adopted from a real-world enterprise portal. Note that we exclude the age of information objects from our analysis such that results of the SIN RP algorithm are comparable to other algorithms not considering the age of information objects. Table 5.4 shows the comparison results.

#	**TotalNumber** (cf. Formula 5.8)	**AverageRate** (cf. Formula 5.9)	**SIN RP** (cf. Formula 5.11)
A	22	4.0	3.730
B	4	3.2	3.179
C	10	4.2	3.670
D	8	1.4	2.411
E	12	3.7	3.452
F	2	5.0	3.461
G	15	2.8	2.954
H	12	1.6	2.338

Table 5.4: Results obtained when applying the SIN RP algorithm to the example.

If we rank the rated information objects by the TotalNumber algorithm (cf. Formula 5.8), the most popular information object will be "A". However, when applying the AverageRate algorithm (cf. Formula 5.9), information object "F" will be the most popular one. However, as a problem, information objects with only few good ratings are ranked on the first positions in the ranking. The SIN RP algorithm (cf. Formula 5.11) addresses this problem. Information objects with many good ratings (e.g., "A" or "C") are now ranked higher than "F". The SIN RP algorithm ensures that an information object with two user ratings and an average rating of 5.0, for example, is not ranked higher than an information object with 50 user ratings with an average rating of 4.9.

Other research influenced the development of the SIN RP algorithm. An approach to improve search results based on user ratings, for example, is presented by Vassilvitskii and Brill [193]. In Lowd et al. [194], a study on rate popularity algorithms is presented and their advantages and disadvantages are discussed. Similar to the SIN RP algorithm, a self-learning algorithm is provided in Bian et al. [195]. The latter addresses both user ratings and content relevance. Like the link popularity algorithms, existing rate popularity algorithms cannot be directly applied to a SIN. Reason is that they do not allow dealing with specific characteristics of the SIN.

Validation (SIN RP Algorithm). We use an automotive scenario to validate the applicability of the SIN RP algorithm. Hence we summarize results in this section; further results are provided in Section 9.2. The results of the SIN RP algorithm were considered as useful by the case study participants. In fact, most participants stated that the ranking of process information as suggested by the SIN RP algorithm is both plausible and useful. Additionally, the SIN RP algorithm avoids the problematic situation that process information with only a few good user ratings is directly ranked on the first position of a ranking.

Altogether, the SIN RP algorithm allows determining the rate popularity of information objects based on user ratings. The algorithm can be further extended to include additional factors, e.g., the experience of knowledge workers and decision makers. For example, user ratings of experienced knowledge workers could be weighted higher.

5.6 Related Work

A SIN constitutes a semantic network; i.e., it represents domain-specific knowledge in a structured and machine-interpretable form [176]. Generally, various types of semantic networks exist. Figure 5.23 shows the most common approaches; i.e., *associative networks*, *topic networks*, *fact networks*, and *ontologies*. The x-axis represents the effort required to create a semantic network, whereas the y-axis represents the degree of support of a semantic network for particular use cases such as search refinement, semantic search, visualization, or reasoning. According to Reichenberger, we distinguish between *light-* and *heavy-weighted* semantic networks [176].

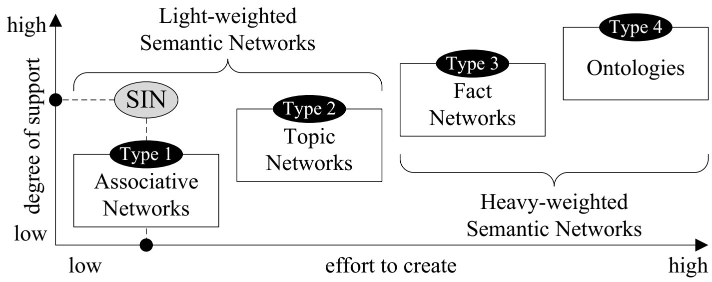

Figure 5.23: Different types of semantic networks.

As shown in Figure 5.23, the effort for creating *associative networks* [174] as well as their degree of support are low. Associative networks are mainly used for search refinement. In turn, *topic networks* [175] provide a higher degree of support than associative networks, but the effort for creating them is significantly higher. They are often used for realizing a simple navigation in semantic networks and for visualizing related topics. In turn, *fact networks* [176] provide an even higher degree of support (e.g., concepts are supported). They are used for realizing personalized views (e.g., context-aware search) and navigation trees (e.g., moderated search). Finally, the highest degree of support is provided by *ontologies* [177, 178]. They provide good results in respect to conceptualization and delimitation of concepts. However, high manual effort is needed to create high-quality ontologies. Important uses cases are semantic search and reasoning.

Unlike existing semantic networks, a SIN focuses on information and process objects as well as their relationships. No manual effort is needed to create or maintain a SIN [196]. For this purpose, we apply algorithms provided by the semantic middleware used to implement the SIN [3]. These algorithms, however, do not allow identifying relevant (i.e., currently needed) information objects within a SIN. As a consequence, we need additional algorithms (cf. Section 5.5) in order to reach the aforementioned goals of POIL; i.e., to provide the right process information to the right process participants.

Note that research in the field of semantic networks has mainly focused on the representation of domain-specific knowledge in a structured and machine-interpretable form. What has been neglected, however, is the maintenance of semantic networks. Generally, semantic networks have in common that they must be maintained. Depending on the type of the semantic network (cf. Figure 5.23), however, the level of maintenance effort varies widely. Commonly, the higher the effort to create a semantic network is, the higher the maintenance effort will be. Semi-automatic maintenance approaches are provided, for example, by Čapek [197], Gargouri et al. [198], and Dinh et al. [199]. However, these approaches cannot be directly applied to the SIN since they do not allow for fully automated SIN maintenance as it is necessary to realize POIL.

5.7 Summary

This chapter presented the semantic layer for enabling both information- and process-awareness in POIL. We motivated the need for them and showed why the handling of process information and business processes is success-critical with respect to overall POIL goals. Most important, we introduced the *semantic information network* (SIN) that represents process information, business processes, and their relationships in a meaningful as well as machine- and user-interpretable form. The SIN allows identifying objects linked to each other in the one or other way, e.g., process information needed when performing a particular process task. Moreover, a SIN must be complete, consistent and up-to-date. Thereby, we showed how it can be maintained by means of three algorithms; i.e., *push*, *pull* and *partial-pull* algorithms. Then, we presented two algorithms for determining the relevance of process information; i.e., the *link popularity* and *rate popularity* algorithms. The first one determines the link popularity of process information based on the relationships of a SIN, whereas the second one determines the rate popularity of process information based on user ratings.

6 Context Layer

This chapter[1] presents the *context layer* of the POIL framework while Section 6.1 provides an introduction. Section 6.2 presents a running example that will be used throughout the chapter. Section 6.3 then describes the context layer and introduces its core component; i.e., the *context model* (CM). Section 6.4 shows how the latter can be used to store and process context information in a user- and machine-interpretable way. Section 6.5 then presents contextual quality dimensions that help us to determine the relevance of process information. Finally, Section 6.6 discusses related work and Section 6.7 summarizes the chapter.

6.1 Introduction

In Chapter 5, we introduced the *semantic layer* that enables both information- and process-awareness in POIL. A remaining challenge is to provide contextualized process information (e.g., personalized forms, checklists, guidelines, or manuals) to process participants [4]. For this purpose, we introduce the *context layer*, which enables *context-awareness* in the POIL framework (cf. Figure 6.1).

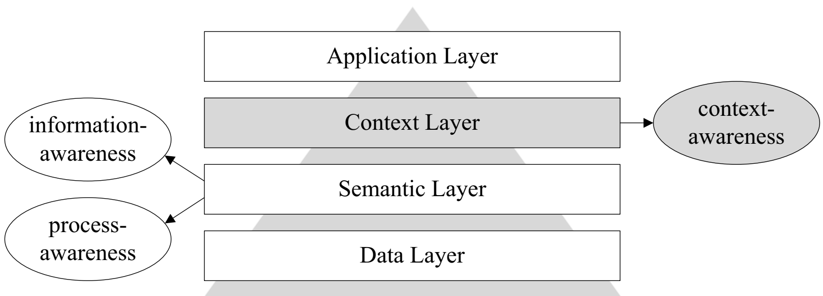

Figure 6.1: Context layer enabling context-awareness.

The context layer is required to be able to address the specific needs of a process participant. For example, less experienced process participants need more detailed process

[1]The chapter is based on the following referred papers:
[1] B. Michelberger, B. Mutschler, and M. Reichert. *Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter.* In: Proc 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107–120, Hamburg, Germany, 2011.
[4] B. Michelberger, B. Mutschler, and M. Reichert. *A Context Framework for Process-oriented Information Logistics.* In: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 260–271, Vilnius, Lithuania, 2012.

information than experienced ones. To enable such a differentiation, the work context of the process participants needs to be taken into account. The following steps are required to accomplish this: First, sensor data (e.g., location items) collected by sensors (e.g., a GPS sensor) is analyzed and harmonized. For example, location items must be represented by the same unit (e.g., decimal degrees[2]). Second, based on available sensor data, context information needs to be characterized and determined (cf. Section 2.2). Besides process-related context information (e.g., temporal process constraints, milestones, quality gates), user-related context information (e.g., role, user experience), device-related context information (e.g., display size or resolution, bandwidth), location-based context information (e.g., position, movement), time-based context information (e.g., current date), and environment-related context information (e.g., noise level) may be considered as well. Finally, the current situation of a process participant is documented.

The POIL context layer aims at the context-aware delivery of relevant process information to process participants [4]. It has been influenced by *mobile and ubiquitous computing* [200, 201]. The fundamental difference between the context layer and existing context frameworks (e.g., Schilit et al. [86], Kaltz et al. [202], Dey et al. [203], Moore and Hu [204]) is the explicit consideration of business processes and their tasks (both at the process schema and instance level [53]). However, context frameworks such as the ones of Pryss et al. [205, 206] and Grambow et al. [207, 208, 209] consider business processes and their tasks. The difference, however, is the scope of these frameworks. They are primarily designed for executing business processes and not for providing process information to process participants. In fact, existing context frameworks strongly focus on geographic services (e.g., provide the local temperature based on a GPS location), whereas ideas of POIL as discussed in Chapter 4 are not addressed. In turn, this means that existing context frameworks and corresponding requirements from the field of mobile and ubiquitous computing can only be transferred partially [4].

We gathered requirements based on two exploratory case studies, an online survey, and practical insights [2, 63]. Table 6.1 summarizes twelve fundamental requirements (R1-R12) for the POIL context layer. These requirements reflect wishes and needs of process participants concerned with POIL such as knowledge workers and decision makers. Moreover, they concern technical issues for realizing the context layer and underlying components, such as the context component, in practice.

Compared to existing frameworks, the POIL context layer does not directly provide any context information to applications (e.g., as a weather forecast application does). Instead, it utilizes context information to determine the process information needed by a particular process participant. The context layer integrates, analyzes and provides context information in order to enable the context-aware delivery of process information.

This chapter introduces the context layer of the POIL framework in detail and presents the processing of context information to enable the context-aware delivery of process information to knowledge workers and decision makers [3].

[2] *Decimal degrees* are commonly used in GPS-based systems and express latitude and longitude geographic coordinates as decimal fractions.

#	Requirements
R1	The context layer should be easy to use to enable application designers to easily translate real-world information to context information.
R2	The context layer should represent all context information relevant for the process participant's situation at a certain point in time.
R3	The context layer should be able to hide irrelevant context information in specific situations (e.g., location in non-mobile scenarios).
R4	The context layer should be able to categorize context information to achieve a better overview and to reduce the amount of context information.
R5	The context layer should be able to interpolate context information to cope with incomplete context information.
R6	The context layer should allow storing and handling historical context information (e.g., what happened at a certain point in time).
R7	The context layer should store context information taking privacy and security issues into account.
R8	The context layer should enable an efficient context analysis (e.g., reasoning, interpretation and aggregation) of context information.
R9	The context layer should allow for the efficient handling (e.g., fast processing, easy accessibility) of context information.
R10	The context layer should be flexible and scalable to cope with the challenges of different update intervals of context information.
R11	The context layer should be easy to use to enable applications to easily use context information.
R12	The context layer should be combined with the SIN to provide contextualized process information to process participants.

Table 6.1: Requirements for the context layer in POIL.

6.2 Running Example

We use a scenario from the clinical domain to motivate and demonstrate the context layer of the POIL framework. This scenario was derived from an exploratory case study performed at a large German university hospital [2, 63]. More specifically, the scenario (cf. Figure 6.2) focuses on a clinical ward round.[3]

First, the ward round is prepared (Task T1); i.e., the doctor scans patient information (e.g., name, gender and previous diseases) and current medical instructions (e.g., endoscopic investigations, physical therapies). After finishing initial preparations, the doctor visits his patients. During such a visit the doctor communicates with the patient and asks for information about his or her health status (Task T2). This information is written down by a nurse at the same time (Task T3). Afterwards, the patient is examined (Task T4). This task includes the analysis of lab values and further follow-up diagnosis.

[3]Note that a *ward round* may vary across different hospitals and even within one hospital, but, notwithstanding, this process can be found in every hospital [168].

Then, the doctor creates medical orders (e.g., take medicine) (Task T5). Finally, a nurse updates patient information and initiates additional medical orders (Task T6).

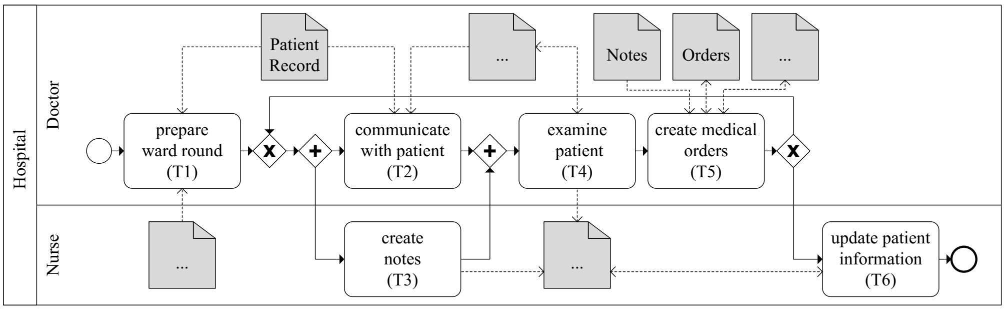

Figure 6.2: Running example: ward round.

For each of the six process tasks (T1-T6), a variety of process information is needed. For example, to perform process task "create medical orders" (Task T5) the doctor needs access to blood values, notes, and current medical orders. Note that the mentioned process information only constitutes a small part of all managed process information. In practice, there exist numerous different process information distributed across data sources (e.g., shared drives, databases) [2]. Typical process information include, for example, process descriptions, working guidelines, operational instructions, checklists, and best practices (e.g., documented in text documents and spreadsheets) [1, 39].

6.3 The Layer

Generally, the context layer can be based on different architectures depending on business requirements and needs. Chen [210], for example, distinguishes between three architectural designs: (1) *direct access to sensors*, (2) *context server*, and (3) *middleware infrastructure*. We adopt the latter architectural design for several reasons, e.g., the reduced complexity resulting from the reduced number of data connections as well as the separation of business logic from the presentation layer [211].

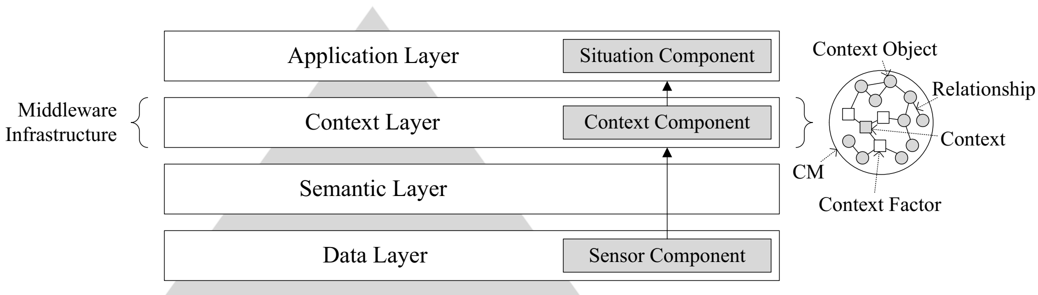

Figure 6.3: Context layer.

Overall, the context layer of the POIL framework comprises three main components (cf. Figure 6.3): (1) *sensor component*, (2) *context component*, and (3) *situation component*. Note that the sensor component is part of the data layer since it provides sensor data from different sensors (i.e., the data sources of the context framework). In turn, the context component constitutes the core component since it realizes the context layer and comprises the CM (cf. Section 6.4). Finally, the situation component is part of the application layer since it documents the situation of knowledge workers and decision makers at a certain point in time. Figure 6.3 illustrates the layered architecture of the POIL framework and the classification of the context layer's components.

6.3.1 Sensor Component

A *sensor* is any hardware or software system that provides sensor data about the work context of one or more process participants. A sensor can be either physical (e.g., thermometer, microphone, barometer), virtual (e.g., keyboard input, touch display movement), or combinatorial (e.g., detect a process participant's position by analyzing logins at devices and a mapping of devices to locations).

Based on this characterization, we provide a formal definition of the term *sensor*. Let SE be the sensor and val be the sensor value. We distinguish between *simple* and *logical* sensors (cf. Formulas 6.1 and 6.2). Note that all types of sensors (e.g., physical, virtual and combinatorial sensors) may either be simple or logical. For example, a simple sensor can be a GPS module determining the position of a user. A logical sensor, in turn, can be a software system determining the user name based on first and last name.

$$SE_{\text{simple}} := val \tag{6.1}$$

$$SE_{\text{logical}} := \{val_1, val_2, ..., val_{\text{n}}\},\ n \geq 2 \tag{6.2}$$

Both simple and logical sensors are handled by the *sensor component* being responsible for the management of sensor data collected by different sensors. The sensor component provides combinatorial functionality, for example, functions to identify the role of a user by analyzing his or her access rights. Furthermore, the sensor component allows adding, removing and switching sensors (e.g., the GPS module will be replaced by a *radio-frequency identification* (RFID) system) as well as encapsulating sensor communication (i.e., applications do not directly access sensor data).

6.3.2 Context Component

A *context* is any information that may be used to characterize the situation of an entity. The latter may be a person, location or object relevant for the context-aware delivery of process information to process participants. More specifically, context consists of *context information*. The latter can be categorized into *context factors* according to different criteria to achieve a better overview (cf. Figure 6.4).

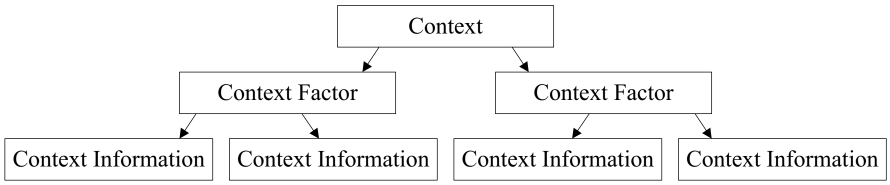

Figure 6.4: Correlation between context, context factors, and context information.

Based on this characterization, we can now provide a formal definition of the terms *context*, *context factor*, and *context information* (cf. Formulas 6.3-6.5). Let C be the context, CF be the context factor, CI be the context information, and SE the sensor (both simple and logical ones). Then, C, CF and CI can be defined as follows:

$$C := CF_1 \cup CF_2 \cup ... \cup CF_{\text{n}},\ n \geq 1 \tag{6.3}$$
$$CF := CI_1 \cup CI_2 \cup ... \cup CI_{\text{n}},\ n \geq 1 \tag{6.4}$$
$$CI := SE_1 \cup SE_2 \cup ... \cup SE_{\text{n}},\ n \geq 1 \tag{6.5}$$

Generally, context, context factors, and context information are managed by the *context component*. The latter includes a *context interface*, *context analytic engine*, *context model*, and *context information system*. The context interface enables the retrieval of sensor data from the sensor component as well as the provision of context information to the situation component. In turn, the context analytic engine allows reasoning, interpreting and aggregating context information (e.g., instead of GPS coordinates, the specific room number is provided) [212]. The context model (cf. Section 6.4) is responsible for representing and handling context information. Finally, the context information system[4] provides process information (e.g., which device belongs to which user, which room has a specific GPS position) to enrich available context information [214].

6.3.3 Situation Component

A *situation* is a part of the world state relevant for the delivery of process information to process participants. Further, it represents the real-world situation of one or more process participants at a certain point in time (e.g., "Doctor Peter Miller communicates with a patient on Tuesday, May 13th, 2014, in room A301 using his tablet computer").

Based on this, we can provide a formal definition of the term *situation* (cf. Formula 6.6). Let S be the situation, C the context, t_{start} the starting time of the situation, and t_{end} its end time. S can then be defined as follows:

$$S := \langle C, t_{\text{start}}, t_{\text{end}} \rangle \tag{6.6}$$

[4]In the area of mobile and ubiquitous computing, a *geographic information system* (GIS) has similar goals as a *context information system* (CIS), but is limited to geographically information solely (e.g., a GPS position) [213].

A situation of a process participant is handled by the *situation component*. The latter is responsible for the context-aware (i.e., personalized) delivery of relevant process information to knowledge workers and decision makers. More specifically, the situation component allows adding and removing third-party applications (e.g., enterprise portals or information systems) as well as encapsulating situation communication.

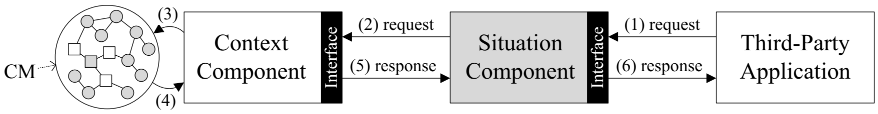

Figure 6.5: Encapsulating situation communication.

Figure 6.5 shows that third-party applications cannot directly access the underlying CM. Instead, third-party applications use the *situation interface* provided by the situation component to request context information from the CM.

6.3.4 Interplay of Components

Figure 6.6 shows the dependencies between the three components. The sensor component provides sensor data (e.g., user name, current process task) to the context component (Dependency D1). Analogously, the context information system provides certain process information (e.g., inventory lists, building maps) to the context component (Dependency D2). The context information system obtains its process information from third-party applications such as portals or information systems (Dependency D3).

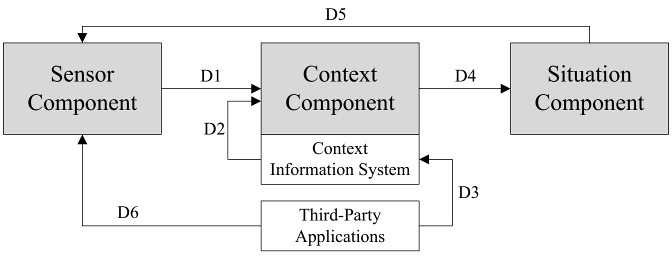

Figure 6.6: Interplay between the components of the context layer.

Based on the data/information flows of dependencies D1 and D2, the context component discovers context information and makes it available to the situation component (Dependency D4). The situation component, in turn, uses the context information to identify the process participants' situation. Besides, the situation component can be a sensor for the sensor component (e.g., in order to gather clickstreams, hits) (Dependency D5). In addition, third-party applications can be sensors as well (Dependency D6).

6.4 Context Model

The core component of the context layer is the *context model* (CM). We use it to represent context information in a user-interpretable, machine-interpretable, and meaningful form (cf. Research Question 3). Such a CM can be created using a bottom-up approach; i.e., starting with the integration of context information from sensors. Following this, the integrated context information is analyzed. The resulting CM as shown in Figure 6.7 comprises a context, context factors, context objects, and their relationships.

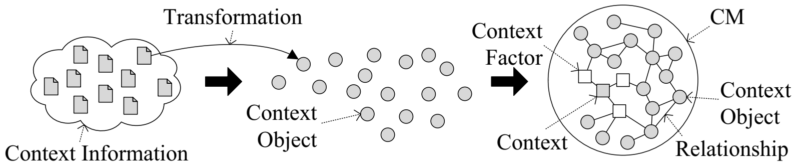

Figure 6.7: Creation of the CM.

More specifically, context objects represent context information in a uniform information format. Context and context factors, in turn, are used to categorize context objects to achieve a better overview and to reduce the amount of context objects.

6.4.1 Definitions

The CM is independent from the SIN; i.e., context objects are only stored in the CM, but not in the SIN. Hence, there exists a central SIN for all users, but a specific CM for each user. Like the SIN, the CM is represented by a labeled and weighted digraph. Based on these considerations, we define a CM as follows:

Definition 21 (Context Model). A *context model* (CM) is a tuple (V, E, L, W, f_l, f_w), where V is a set of *vertices* with each $v \in V$ representing a *context*, *context factor*, or *context object*. In turn, E is a multiset of *edges* with each edge $e = (v, v') \in E$, and $v, v' \in V$ representing a *relationship* between objects. Function $f_l \colon E \to L$ labels each edge $e \in E$ with an edge *label* from the set of labels L. Function $f_w \colon E \to W$ assigns a *weight* from the set of weights W to each edge $e \in E$. Given an edge $e = (v, v') \in E$, we denote v as the *source* and v' as the *destination* of e.

Note that we often refer to vertices as objects (e.g., context objects) and to edges as relationships. Next, we define *properties* for vertices and edges as follows:

Definition 22 (Properties of a Vertex and Edge in a CM). Each vertex $v \in V$ and each edge $e \in E$ comprises a set of *properties* $P(v)$ and $P(e)$, respectively. Thereby, each $p \in P(v) \cup P(e)$ is a pair (key, val) with key being the unique name and val being the value of p. For short, $key(v)$ and $key(e)$ denote val.

Altogether, the CM is responsible for storing and handling context objects. Based on these context objects, POIL is able to better identify relevant process information for process participants when working on business processes and their tasks.

6.4.2 Context Factors

Generally, a CM for the POIL framework should represent all context objects relevant in the current situation of a process participant. However, any CM must be restricted because the set of context objects is infinite [215]. Hence, any context modeling approach can only capture parts of all possible context objects. Thus, we need a classification (cf. Figure 6.8) of context objects that allows us to reduce the complexity of context modeling. For example, this enables us to process context objects (e.g., first name, last name) in the same category (e.g., user) using the same or similar algorithms (e.g., term frequency algorithms or methods of clustering).

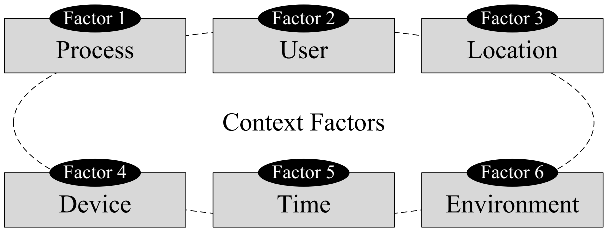

Figure 6.8: Context factors of the CM.

Regarding POIL, we use the following six context factors (CF1-CF6):

- **Process (CF1)**. This factor includes process-related context objects and reflects on what is currently happening and what happened in the past. This includes, for example, general process-based information (e.g., number of process instances, quality gates), time-based information (e.g., duration and time lags between process tasks, restricting execution times), responsibility-based information (e.g., process owner), and data-based information (e.g., input and output files).
- **User (CF2)**. This factor includes user-related context objects and reflects who is involved in a particular situation. Thereby, we distinguish between explicit user information (e.g., user name, first name, last name, birthday, or department) and implicit one (e.g., user experiences, interests).
- **Location (CF3)**. This factor includes location-based context objects and reflects where a situation takes place. This context factor includes both physical location information (e.g., GPS coordinates, Geo location, and RFID systems) and logical one (e.g., meeting room or office room).
- **Device (CF4)**. This factor includes device-related context objects and reflects which devices are used in a certain situation. It includes type information (e.g., personal computer, notebook, tablet, smartphone), hardware information (e.g.,

processor, disk space, display size), software information (e.g., operating system, installed applications), and others (e.g., display properties, bandwidth).

- **Time (CF5)**. This factor includes time-based context objects like current time, virtual time, time zone, business days, and calendar week.
- **Environment (CF6)**. This factor includes environment-based context objects and reflects what environmental information influence a situation. We distinguish between physical information (e.g., noise level, lightening), organizational information (e.g., corporate culture, enterprise policies, corporate identity guidelines), and legal requirements (e.g., privacy policy, regulations).

6.4.3 Creation Phases

As mentioned, a CM is created based on available context information [4]. The CM is an ontology-based model using the pre-defined context factors as introduced in Section 6.4.2. The CM allows characterizing the work context of a process participant. This work context can then be used to filter the SIN (cf. Chapter 7). Overall, the CM is created in two consecutive phases. We illustrate the individual phases along the running example introduced in Section 4.2; i.e., the procurement of drugs in a hospital.

Phase 1: Integration of Context Information. Phase 1 deals with the integration of available context information (e.g., user name, department, e-mail address, location). Context information is identified and gathered by sensors, and then used to create the CM (cf. Figure 6.9). Note that context information is transformed in context objects. This is a necessary prerequisite for the subsequent analysis in Phase 2. Moreover, note that the validity of context objects can rapidly change (e.g., when a process participant changes his location). Therefore, the context layer for POIL must support real-time and rapid processing of context objects (cf. Table 6.1).

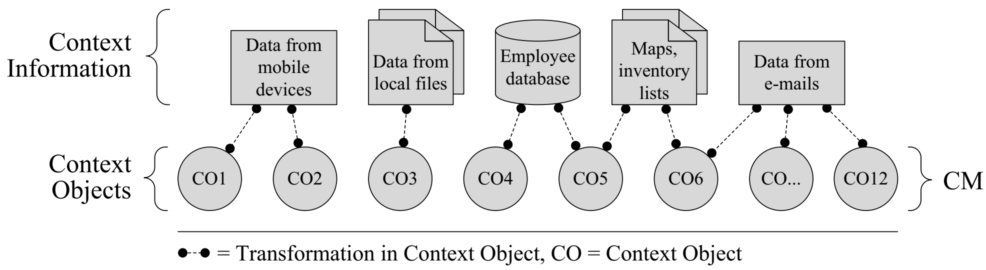

Figure 6.9: Phase 1: integration of context information.

Phase 2: Context Objects Relationships. Phase 2 deals with the analysis of context objects, i.e., the discovery of relationships between them. For this purpose, algorithms enabling the aggregation, interpolation and interpretation of context objects are used [87]. For example, instead of GPS coordinates, the room number is determined (aggregation) or incomplete context information is completed (interpolation).

Picking up the healthcare scenario from Section 4.2, Figure 6.10 shows an example of a CM based on context factors (i.e., process, user, location, device, time, environment) for the following situation: "Doctor Peter Miller communicates with a patient on Tuesday, May 13th, 2014, in room A301 using a tablet computer" [4].

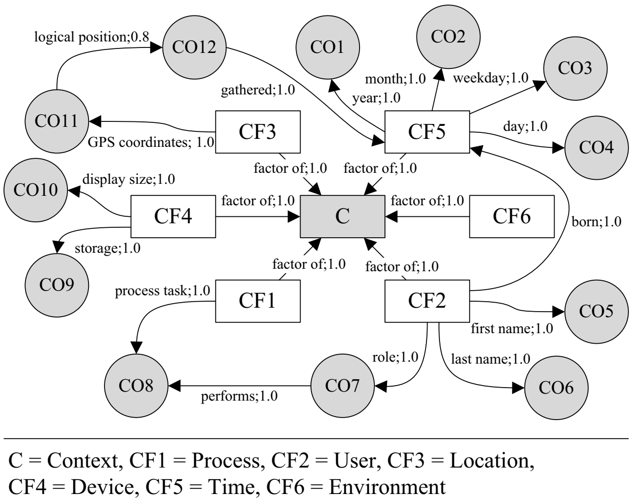

Figure 6.10: Phase 2: determining context objects relationships.

However, identifying the work context of a process participant is not sufficient in order to identify relevant process information [1]. Another crucial aspect influencing the relevance of process information is its quality.

6.5 On the Contextual Quality of Process Information

Based on the question whether process information fulfills certain quality requirements, the overall relevance of process information can be determined. Picking up the healthcare scenario again, typically, patient information should be up-to-date and complete in order to be able to charge services through the accounting and billing department. Therefore, outdated or incomplete patient information is not relevant, since it cannot be processed by the medical accounting. However, depending on a specific work context, different quality dimensions might be more or less important than others. For example, for a surgeon, patient information should be available punctual and up-to-date. Conversely, for an employee responsible for patient admission, information about the patient must be complete and error-free. Therefore, depending on the work context, a different weighting of individual quality dimensions becomes necessary. In fact, the consideration of work context and quality dimensions of process information is the key to identify relevant process information. Figure 6.11 shows the relationship between work context and process information quality. On one hand, the work context determines the process information a process participant needs to perform current tasks. On the other, the use of quality dimensions allows determining process information quality. Both issues allow determining the overall relevance of process information.

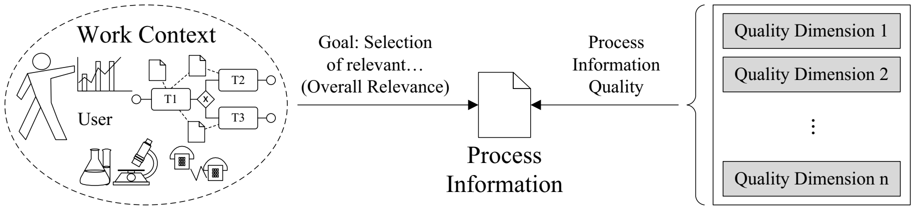

Figure 6.11: Determining the relevance of process information.

Process information quality can be investigated from various viewpoints, e.g., integration, transmission, security, storage, access, and representation. According to the POIL goals, as discussed in Chapter 4, we focus on the viewpoints of *integration*, *semantic analysis*, and *delivery*. Integration deals with the collection of process information from different data sources (e.g., shared drives, databases). The viewpoint semantic analysis implies to semantically process and link process information. Finally, the delivery viewpoint deals with the technical delivery of process information.

6.5.1 Quality Categories

Quality dimensions of process information can be combined into different *quality categories*. Each category subsumes a set of quality dimensions (cf. Figure 6.12).

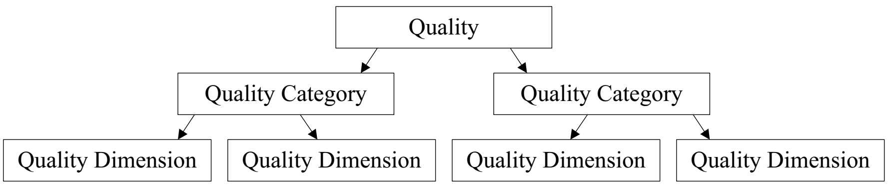

Figure 6.12: Correlation between quality, quality categories, and quality dimensions.

All dimensions belonging to the same category (cf. Figure 6.13) are affected by the same influencing factors such as work context (e.g., process- and user-related information) or information systems characteristics (e.g., representation of process information) [216]. Specifically, we apply the classification of Wang and Strong [166].

Regarding POIL, we use the following four quality categories (QC1-QC4):

- **Intrinsic (QC1).** This quality category integrates self-contained quality dimensions of process information. Dimensions from this category are independent on the work context. Examples include "believability" (e.g., to improve the believability of a medical diagnosis several doctors have to approve it) and "objectivity" (e.g., to guarantee the objectivity, the health status of patients must be determined by certain criteria and not by estimation). Another example is "free-of-error" (e.g., to achieve error-free patient lists, name and identification number of the patient must match).

- **Accessible (QC2).** This quality category combines quality dimensions being important for the access to process information. These are mainly affected by the information systems providing process information. Examples of respective quality dimensions include "accessibility" (e.g., to treat a patient the doctor needs the patient record) and "security" (e.g., ensure the security so that specific process information is only accessible to authorized users).

- **Representational (QC3).** This quality category subsumes quality dimensions concerning the representation of process information. This quality category is again mainly influenced by the information systems providing process information. As examples of respective quality dimensions consider "interpretability" (e.g., the exact unit of measurement is always indicated for the given values), "understandability" (e.g., addresses should not be displayed as GPS coordinates), "consistent representation" (e.g., patient information should be displayed consistently), and "concise representation" (e.g., current diseases are displayed separately from pre-existing diseases or associated symptoms).

- **Contextual (QC4).** This quality category integrates quality dimensions that are influenced by the work context of process participants. Contextual quality dimensions are, for example, "contextual relevance" (e.g., a doctor performing task "prepare ward round" receives other process information than in task "create medical instructions"), "completeness" (e.g., patient information must be completely available), and "punctuality" (e.g., blood values must be available when the doctor needs it). These quality dimensions always depend on the current work context.

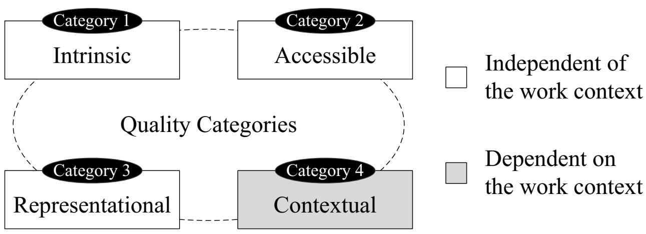

Figure 6.13: Quality categories of process information.

In the next section, we restrict ourselves to the contextual quality category since this quality category is particularly influenced by the work context and thus also by process-related information (e.g., process descriptions, process execution times).

6.5.2 Contextual Quality Dimensions

We distinguish between nine contextual quality dimensions (QD1-QD9) of process information (cf. Figure 6.14). Each dimension is described in the following.

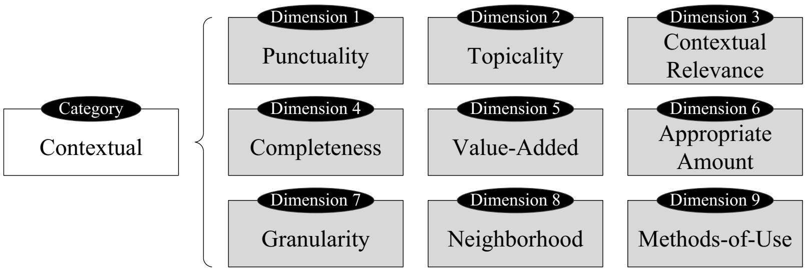

Figure 6.14: Contextual quality dimensions of process information.

Punctuality (QD1). This quality dimension indicates whether process information is provided punctually when the user needs it. Specifically, three different time points (t) have to be distinguished: (1) the point in time at which the process participant *requests* the process information, (2) the point in time at which the process information is *provided*, and (3) the point in time at which the process participant *applies* the process information. Based on this, we can determine whether process information is punctual.

Additionally, it becomes necessary to distinguish between *ad-hoc process information* and *regular one*. The former is requested spontaneously. For example, a doctor may request blood values in order to be able to make decisions. Ad-hoc process information is accurate in time if it is provided between the point in time it is requested and the one it is used (cf. Figure 6.15). The length of this period depends on the process participant.

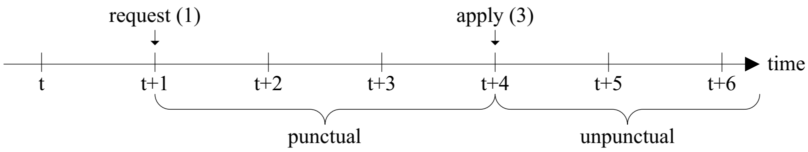

Figure 6.15: Punctuality of ad-hoc process information.

Conversely, regular process information is provided at pre-defined points in time. For example, every morning a doctor may receive a patient list in order to know which patients he must visit. The punctuality of regular process information can be distinguished between two time points: punctual in respect to the provision and punctual in respect to the use (cf. Figure 6.16).

Topicality (QD2). This quality dimension indicates whether process information captures the current characteristics (e.g., name, insurance agreement) of an object (e.g., patient) at the current point in time (t). Process information is out-of-date if the characteristics of the object have changed during the time point of *capture* and the time

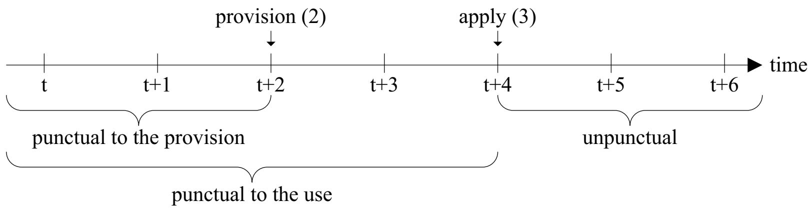

Figure 6.16: Punctuality of regular process information.

point at which the process participant *applies* the process information (cf. Figure 6.17). For example, the body temperature of a patient measured two days ago is most likely obsolete. In practice, to capture characteristics is often time-consuming. In particular, characteristics of an object may continuously change (e.g., body temperature or health status of a patient). The capture can be done either in real-time (e.g., using a heart rate monitor) or at pre-defined time points (e.g., during the ward round).

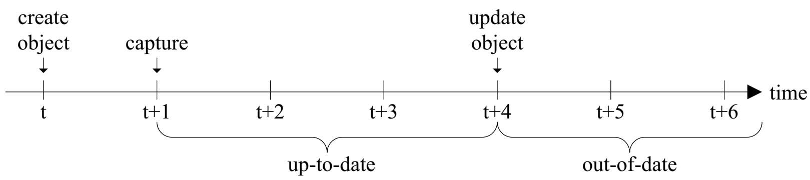

Figure 6.17: Topicality of process information.

Contextual Relevance (QD3). This quality dimension indicates whether process information is relevant in a specific work context. Process information has a high contextual relevance if it is needed to perform or support a task. For example, for preparing the ward round a doctor needs current diagnoses and medical instructions. The more precise a work context can be defined, the more accurate can the contextual relevance be determined. Therefore, it becomes necessary to not only consider process- and user-related information, but location-based, device-related, and time-based information as well.

Unlike the overall relevance (cf. Figure 6.11), the contextual relevance is not influenced by other quality dimensions. As an example reconsider the preparation of the ward round for which the doctor needs access to the patient record. Let us assume that the patient record is punctually available. In this case, the patient record has high contextual relevance and high overall relevance. Let us assume that the patient record is not punctually available. In this case, the contextual relevance is still high, but no overall relevance can be identified since quality dimension punctuality is not fulfilled.

Completeness (QD4). This quality dimension indicates whether all parts of a complex process information (comprising several information parts) are available. To perform task "create medical instructions", for example, different blood values (together representing a process information) must be available. Process information is incomplete if some parts are missing. It is important to mention that completeness thereby depends on the currently needed information; i.e., it depends on the current work context. Regarding the running example, this does not mean that all blood values must be available, but only those being needed for current patient treatment.

Value-Added (QD5). This dimension indicates whether it is possible to increase some "value" (e.g., patient satisfaction, diagnostic accuracy) by using process information. For example, information about patient needs is value-added because the fulfillment of the needs increases patient satisfaction. The value-added amount is calculated as the difference between the value that can be realized without using specific process information and the value that can be realized based on specific process information. Figure 6.18 shows this relationship. However, it is quite difficult to determine the value-added quality dimension since respective effects often cannot be exactly estimated.

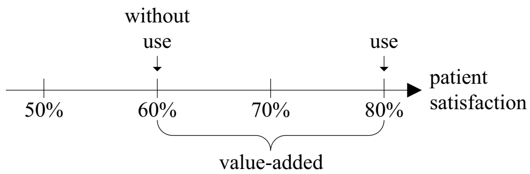

Figure 6.18: Value-added of process information.

Appropriate Amount (QD6). This quality dimension of process information indicates whether the amount of available process information is sufficient. This is the case if the amount meets the requirements of process participants. For example, a doctor might need the name of the patient as well as pre-existing diseases. The amount of process information will not be appropriate if he gets the entire patient record. In practice, this problem is solved by extracting process information via extraction algorithms (e.g., a document is divided into individual information objects). In our case studies, we analyzed the appropriate amount of process information. Obviously, decision makers are confronted with too much information. Knowledge-workers, by contrast, have the problem of being confronted with insufficient information.

Granularity (QD7). This dimension indicates whether the aggregation applied to process information meets the requirements of process participants. Process information will have the right level of granularity if immediate use is possible (cf. Figure 6.19). For example, if a doctor needs to know the average body temperature a patient had during the past week, he should immediately get the calculated average value instead of the individual values. According to Jung, three aggregation dimensions need to be distinguished [217]; i.e., (1) *time dimension*, (2) *area-specific dimension*, and (3) *value and quantity dimension.* As an example of the time dimensions consider the aggregation based on emergencies per day. Examples of the area-specific dimension include aggregation by organization units (e.g., number of patients on ward A) or patients (e.g., patients' age and gender). As an example for the value and quantity dimension consider aggregations relating to cost centers (e.g., research and development, patient service).

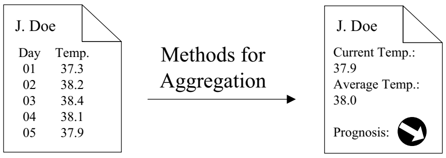

Figure 6.19: Granularity of process information.

Unlike the granularity, the appropriate amount (QD6) meets the requirements if non-aggregated information is provided. Assume that the doctor wants to know the average body temperature of a patient. If individual data items are provided, only QD7 meets the requirements. If the average body temperature is provided, both QD6 and QD7 meet the requirements.

Neighborhood (QD8). This quality dimension indicates how strong and how frequently process information is linked to other process information. Process information, which is strongly and frequently linked, tends to be more important. In addition, the semantics of the relationship is important. Examples include metadata-matching (e.g., author, file format), text similarities, and usage-patterns [171] (cf. Section 5.5.1).

Methods-of-Use (QD9). This quality dimension indicates how a process participant uses the process information. Suitable use cases are, for example, to read, create, update, and delete the process information. For example, a process participant cannot be provided with read-only process information if he wants to edit process information.

6.5.3 Measuring Quality Dimensions

Quality dimensions are an important means to determine process information quality and the overall relevance of process information for a particular process participant. Thus, they are important means to realize the POIL framework.

Generally, there exist different approaches that can be used to decide whether process information fulfills specific contextual quality dimensions: (1) *algorithmic methods*, (2) *semantic technologies*, (3) *social methods*, and (4) *convergent methods*.

Algorithmic methods are, for example, the vector space model, the term frequency algorithm and methods of clustering. The use of semantic technologies is another possibility to determine process information quality (e.g., via ontologies) [171, 218]. Social methods include, among others, collaborative tagging or human-based rating of process information [219]. Finally, convergent methods improve the aforementioned methods through their combination (e.g., algorithmic detected relationships between process information are editable by process participants). Table 6.2 illustrates which of these methods can be used to determine our introduced contextual quality dimensions.

#	Quality Dimensions	Algorithmic	Semantic	Social	Convergent
QD1	Punctuality	×		×	×
QD2	Topicality	×		×	×
QD3	Contextual Relevance	×	×	×	×
QD4	Completeness	×	×	×	×
QD5	Value-Added			×	
QD6	Appropriate Amount	×		×	×
QD7	Granularity	×		×	×
QD8	Neighborhood	×	×	×	×
QD9	Methods-of-Use			×	

Table 6.2: Methods to determine process information quality.

6.6 Related Work

Context and context-awareness in general are discussed by Pascoe et al. [84], Schilit et al. [86], and Dey [87]. In turn, context-awareness in IL is discussed by Haseloff [42], Meissen et al. [77], Levashova et al. [120], and Lundqvist et al. [121].

In recent years, different approaches have been proposed to deal with challenges of context-awareness and context modeling. Especially, in the research field of mobile and ubiquitous computing a numerous of context layers (or context frameworks) have been proposed (e.g., Context Toolkit [220], Hydrogen [221]). Further context frameworks exist, for example, in the field of IR (e.g., SAiMotion [222]). In turn, a broader view on context models supporting business process agility is given by Thönssen and Wolff [223]. Only a few approaches combine business processes with context-awareness as described in this work. Context frameworks such as the ones of Pryss et al. [205, 206] and Grambow et al. [207, 208, 209] consider business processes and their tasks. The difference, however, to our work is the scope. Existing approaches are primarily designed for executing business processes and not for providing process information to process participants.

Several authors investigated possible categorizations of context information into context factors. For example, Schilit et al. distinguish between location (where you are), identity (who are you with), and device (what resources are nearby) [86]. Kaltz et al. [202, 224] propose user and role, process and task, location, device, and time as possible context factors when representing web application scenarios. Dey et al. stated that certain context factors are more important than others [87, 203]. These are location, identity, activity, and time. In this thesis, we use process, user, location, device, time, and environment as context factors (cf. Section 6.4.2). Table 6.3 compares the chosen categorization of context information with the ones introduced by other authors.

The context factor *user* (also called identity or role) and *location* are proposed in all mentioned articles. Both Kaltz et al. [202] and Dey et al. [203] suggest *process* (also called

#	Context Factors	Thesis	[86]	[202]	[203]
CF1	Process	×		×	×
CF2	User	×	×	×	×
CF3	Location	×	×	×	×
CF4	Device	×	×	×	×
CF5	Time	×		×	×
CF6	Environment	×			

Table 6.3: Categorizations of context information into context factors.

task or activity) and *time* as important factors. The context factor *device* is mentioned by Kaltz et al. [202] and Dey et al. [203]. The context factor *environment* is addressed by none of the mentioned authors. Although examples are given for the environment, none of the mentioned author considers the environment as a context factor.

Based on context factors, it becomes easier to model a context. Different context modeling approaches can be used for this purpose: *key-value*, *markup scheme*, *graphical*, *object-oriented*, *logic-based*, and *ontology-based models* [225]. In the POIL context layer, ontology-based models (cf. Table 6.4) are used, since there exists powerful tool support for ontologies. Furthermore, partial validation and distribution of context information becomes possible and ontologies allow easy linking to other ontology-based models (e.g., ontology-based process information and business process models such as the SIN). Finally, ontologies have strengths regarding normalization and formality. Several authors (e.g., by Strang and Linnhoff-Popien [225]) share our assessment that ontology-based models provide a promising approach to deal with the challenge of context modeling.

Criteria	Key-Value	Markup	Graph.	Object	Logic	Ontology
Ease of use	++	+	o	o	−	o
Formalization	−	o	o	o	++	++
Expandability	− −	+	o	+	−	++
Expressiveness	−	o	+	+	++	++

++ = very good, + = good, o = neutral, − = bad, − − = very bad

Table 6.4: Comparison between context modeling approaches.

Quality dimensions of information in general have been considered in literature as well. Jung, for example, investigates data integration architectures and also sketches quality dimensions [217]. Wang et al., in turn, identify aspects of data quality based on empirical research and integrate their findings into a data quality framework [67, 226, 227]. Naumann et al. describe a framework for multi-database query processing taking information quality into account [228]. Table 6.5 compares our quality dimensions (QD1-QD9) with the ones of the aforementioned authors (i.e., [217], [227] and [228]).

#	**Quality Dimensions**	**Thesis**	**[217]**	**[227]**	**[228]**
QD1	Punctuality	×	×	×	×
QD2	Topicality	×	×	×	×
QD3	Contextual Relevance	×			
QD4	Completeness	×	×	×	×
QD5	Value-Added	×		×	
QD6	Appropriate Amount	×	×	×	×
QD7	Granularity	×	×		
QD8	Neighborhood	×			
QD9	Methods-of-Use	×	×		
*	Relevance		×	×	×
*	Periodicity		×		
*	Price				×

Table 6.5: Contextual quality dimensions from different viewpoints.

As shown in Table 6.5, quality dimension *relevance* is not a separate dimension in this thesis. The latter results from the combination of all identified quality dimensions (QD1-QD9). Moreover, the quality dimension *periodicity* is based on the information sources and is therefore not a quality dimension of process information. Further, quality dimension *price* can be omitted because commercial data providers are not in focus in this thesis. Note that Wang et al. subsume QD1 and QD2 under the term *timeliness* [227]. Jung subsumed QD4 and QD6 under the term *completeness* [217]. Naumann [228] closely follows Wang [227]. Due to different perspectives some quality dimensions of Wang [227] (e.g., QD5) are omitted in the research of Naumann [228].

6.7 Summary

This chapter presented the context layer that is fundamental for enabling context-awareness in POIL. We motivated the need for the latter and showed why the handling of context information is success-critical with respect to the context-aware delivery of process information. Most important, we introduced the context layer that allows gathering, representing, storing, analyzing, and providing context information along executed business processes. More specifically, we described the layer's architecture and introduced important context factors and context information (to be used in context modeling). Moreover, we introduced contextual quality dimensions of process information helping us to determine the contextual quality of process information.

7 Application Layer

This chapter presents the *application layer* of the POIL framework. Section 7.1 gives a short introduction. Section 7.2 then describes the application layer and introduces its core component; i.e., the *semantic information network facade* (SIN Facade). Section 7.3 shows how this facade allows for the provision of SIN information and SIN process objects during business process execution. Finally, Section 7.4 discusses related work and Section 7.5 summarizes the chapter.

7.1 Introduction

In Chapters 5 and 6, we introduced the *semantic* and *context layer* that enable information-, process- and context-awareness in the POIL framework. A remaining POIL challenge, however, is to provide process participants with the needed process information when performing business processes at the operational level. For that purpose we introduce the *application layer* (cf. Figure 7.1).

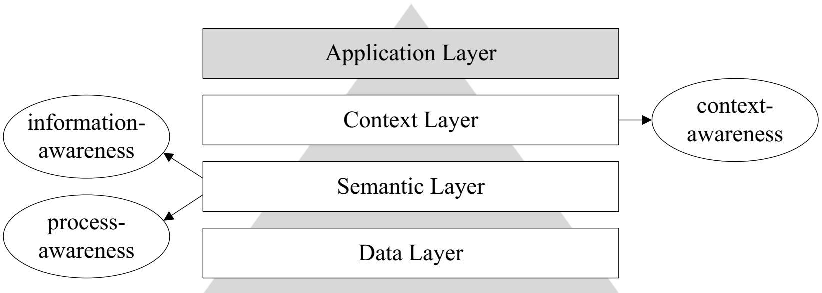

Figure 7.1: Application layer enabling delivery of process information.

Consider an engineer in the automotive domain who deals with the problem of finding colleagues working on similar process tasks. More specifically, this use case can be generalized as follows: "*show all roles within a business process that perform similar or identical process tasks to a given role, selected by a process participant.*"

To realize this use case, the engineer selects a specific role from the list of existing roles (cf. Figure 7.2A). Then, he defines a threshold (i.e., the minimum weight) that indicates the similarity between the selected role and the one to be found (cf. Figure 7.2B). The higher the threshold is, the higher the similarity between the two roles must be.

Relevant roles are identified in five consecutive steps: First, all process objects in the SIN representing a specific role must be identified. Afterwards, for each identified role, the SIN Facade must identify the process tasks the role is responsible for; i.e., based on "is responsible for" relationships. Then, the SIN Facade identifies similar process tasks for the ones identified in the second step. Therefore, "is similar to" relationships and the threshold are used. Following this, responsible roles for the process tasks identified in the third step are being gathered. Again, "is responsible for" relationships are used to identify roles. Finally, the latter are provided to the engineer (cf. Figure 7.2C).

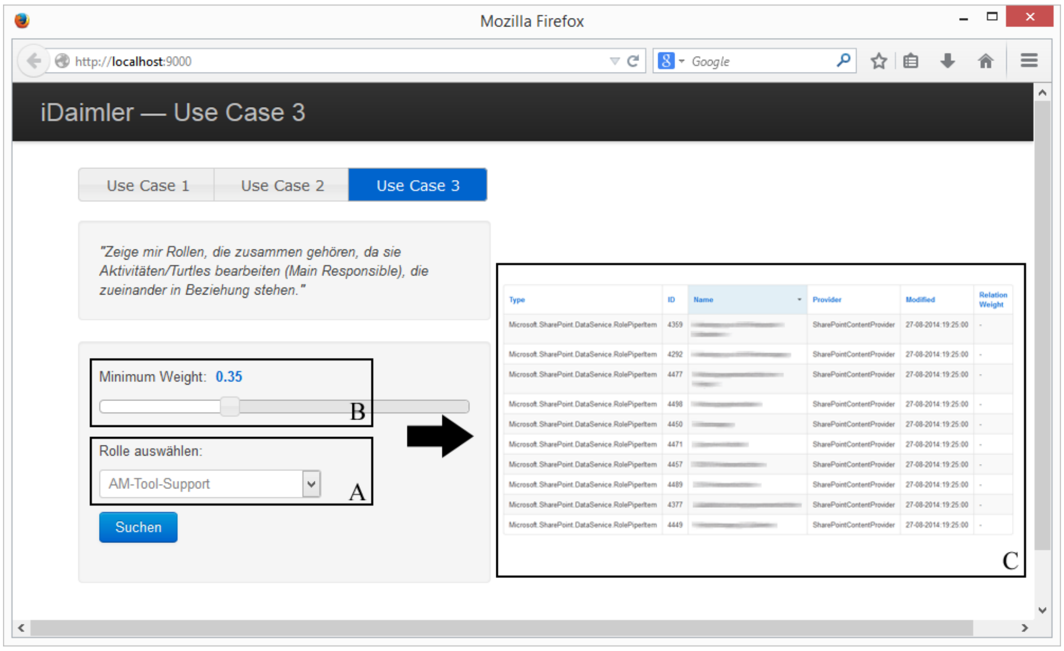

Figure 7.2: iDaimler: use case.

The remainder of this chapter introduces the application layer of the POIL framework. In particular, it shows how the SIN Facade provides information as well as process objects to process participants. Moreover, we show how the CM is used to filter the SIN enabling the delivery of contextualized process information being relevant for process participants.

7.2 The Layer

The application layer is realized by the SIN Facade aiming at the context-aware delivery of process information to process participants (cf. Research Question 5). Thereby, the SIN Facade constitutes an interface to retrieve both information and process objects from the SIN taking context objects from the CM into account (cf. Figure 7.3).

Overall, the application layer allows applications to query the SIN and to filter the latter by the use of the CM. Moreover, the application layer is responsible for security issues. For example, if a process participant queries the SIN, the layer should filter the results and remove the objects for which a process participant has no access rights.

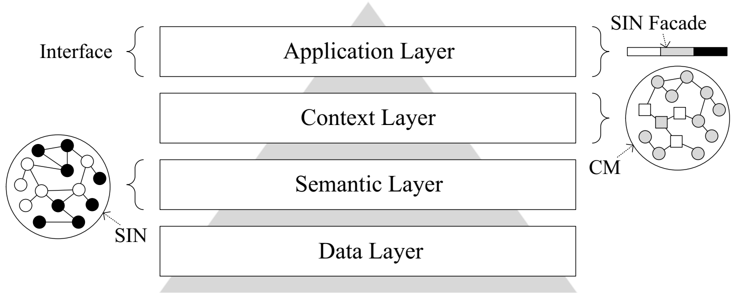

Figure 7.3: Application layer.

7.3 Semantic Information Network Facade

The underlying core component of the application layer is the *semantic information network facade* (SIN Facade). The SIN Facade is responsible for the processing of queries (e.g., get all objects comprising "review car control unit") as well as the delivery of respective information and process objects to the user (cf. Figure 7.4).

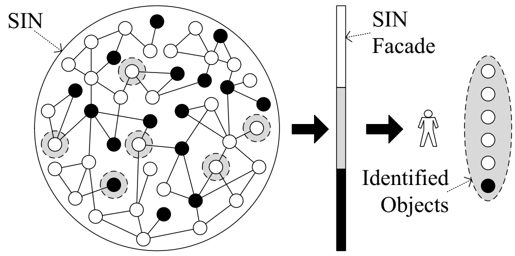

Figure 7.4: Delivering relevant information and process objects.

The SIN Facade can be queried in various ways from different actors (e.g., applications, process participants). We distinguish between *explicit* and *implicit* information demand.

Examples of an explicit information demand include *full-text information retrieval* (e.g., delivery of objects matching a search query), *concept-based information retrieval* (e.g., delivery of objects dealing with a topic such as "review"), and *graph-based information retrieval* (e.g., delivery of information objects related to a set of process objects). In turn, an example of an implicit information demand is *context-based information retrieval* (e.g., delivery of objects taking the process participant's work context into account). These information demand examples (E1-E4) are depicted in Table 7.1.

We utilize a commercial semantic middleware platform (i.e., the iQser GIN Server 2.0) implementing the SIN Facade [171]. Thereby, the latter is realized by a SOAP web service providing methods for querying the SIN (cf. Figure 7.5).

#	Information Demand Examples
E1	A *full-text information retrieval* provides one or more objects from the SIN based on a search query. Thereby, we distinguish between queries regarding the properties of objects (e.g., title) and the contents of objects (e.g., the full text of an office document). A full-text search often constitutes the starting point of a more in-depth search. For example, a process participant may search for objects that include the term "review car control unit". Then, he investigates which objects have been frequently accessed by other process participants in the same context.
E2	A *concept-based information retrieval* provides one or more objects from the SIN dealing with a specific topic (e.g., development, quality management). Thereby, topics are extracted from the objects' contents. Besides the topic itself, each topic has a significance value indicating its importance and a frequency value expressing how often it occurs. Hence, it is possible to distinguish between important and less important topics (cf. Figure 8.9).
E3	A *graph-based information retrieval* provides one or more objects from the SIN based on a given sub-part of the SIN. For example, a process participant may want to identify process information needed in the context of a specific business process. The business process itself is represented in the SIN as a set of related process objects (e.g., tasks, events, gateways, roles, and data objects). Accordingly, the part of the SIN representing the business process constitutes the input of a graph-based information retrieval.
E4	A *context-based information retrieval* provides one or more objects from the SIN based on the specific work context of a process participant. As aforementioned, the work context is represented by the CM. The most important context factor in the CM is *process*. In particular, it also indicates the current process task a process participant is working on (cf. Figure 8.6). In turn, the current process task constitutes the starting point of a context-based search in POIL; i.e., the process object representing the process task. Other context factors may be also used to filter the SIN. For example, the factor *device* ensures that no objects are provided, which can then be not visualized on the screen of the process participant's device.

Table 7.1: Information demand examples.

7.4 Related Work

Besides the SIN Facade, there exist a number of similar approaches. In the context of the semantic web, for example, various languages exist that have been used to query a network. They are ranging from selection languages to fully-fledged reasoning languages. According to Bailey et al., two types of languages can be distinguished [229]: the more specific *restricted* and the more *general languages*. The former are restricted to a certain representation format (e.g., *resource description framework* (RDF), SIN). In turn, the latter support multiple representation formats. Examples of restricted languages are XPath [230], XSLT [231] and Tolog [232]. A general language that supports multiple representation formats (e.g., *web ontology language* (OWL), RDF) is SPARQL [233]. Moreover, frameworks for querying semantic networks exist. Tzompanaki and Doerr, for

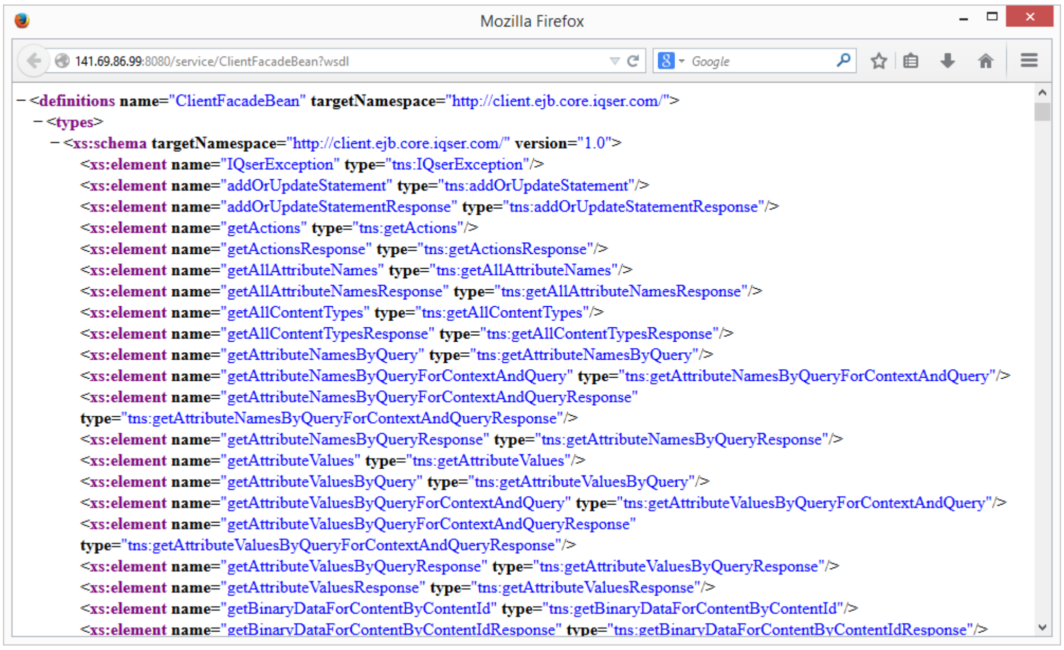

Figure 7.5: Implemented SIN Facade.

example, provide such a framework for formulating queries by a small list of configurable relationships and relevant specializations [234].

However, most of these approaches have been originally developed for specific representation formats (e.g., XML, RDF and OWL) and cannot be directly applied to POIL (and thereby to the SIN) without modification. Particularly, they do not allow dealing with the specific characteristics of a SIN. For example, relationships in a SIN are multiple, weighted and labeled (cf. Section 5.3). Moreover, existing approaches do not consider business processes in a semantic network. Therefore, it is not possible, e.g., to request all related process information of a specific business process. Moreover, existing approaches do not consider the specific work context of a process participant.

7.5 Summary

This chapter presented the application layer of the POIL framework. First and foremost, we introduced the SIN Facade, which allows retrieving both information and process objects from the SIN as well as applying the CM to filter the SIN. More specifically, we introduced different ways to query the SIN, e.g., using full-text, concept-based, graph-based, or context-based information retrieval techniques.

Part III

Validation

8 Applying Process-Oriented Information Logistics to Real-World Scenarios

This chapter[1] deals with real-world scenarios and shows how they can be supported by the POIL framework. Sections 8.1-8.3 present proof-of-concept prototypes demonstrating the applicability of POIL and its benefits. Section 8.4 then presents a sophisticated business process navigation approach based on the POIL framework. Section 8.5 discusses the lessons learned when developing and applying the prototypes. Finally, Section 8.6 summarizes the results of this chapter.

8.1 Healthcare Domain: iCare

The quantity and diversity of medical information emerging during patient treatment makes it a challenging task for medical staff to identify and handle the medical information needed for performing their tasks in the best possible way [2]. During a ward round [168], for example, doctors not only need access to patient records, but also base their decisions on medical orders, medical reports, and medical knowledge. Generally, the effective and efficient delivery of medical information is a prerequisite for evidence-based decisions, orders, treatments, diagnoses, and therapies [235].

As a particular problem, medical staff can only spend very limited time for each patient. Studies (e.g., [168] and [236]) have revealed that doctors can only spend few minutes for searching and handling medical information per patient and ward round.

To demonstrate how the POIL framework could relieve medical staff from this time-consuming task, we implemented the iCare[2] prototype [6]. iCare is a web-based Java application relying on semantic technology. Its goal is the contextualized (i.e., personalized) delivery of medical information to medical staff. In particular, medical staff does not need to search for medical information anymore, but is automatically supplied with

[1]The chapter is based on the following referred papers:
[6] B. Michelberger, A. Reisch, B. Mutschler, J. Wurzer, M. Hipp, and M. Reichert. *iCare: Intelligent Medical Information Logistics.* In: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), ACM Press, pp. 396–399, Vienna, Austria, 2013.
[10] B. Michelberger, B. Mutschler, D. Binder, J. Meurer, and M. Hipp. *iGraph: Intelligent Enterprise Information Logistics.* In: Proc 10th Int'l Conf on Semantic Systems (SEMANTiCS'14), Posters & Demonstrations Track, CEUR Workshop Proc 1224, pp. 27–30, Leipzig, Germany, 2014.
[12] M. Hipp, B. Mutschler, B. Michelberger, and M. Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* In: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE Computer Society Press, pp. 1–12, Marrakesh, Morocco, 2014.
[2]A screencast presenting the iCare prototype is available at http://nipro.hs-weingarten.de/screencast.

it depending on the current work context. Thus, the required time for searching the needed medical information and its handling can be significantly reduced.

8.1.1 Scenario

iCare aims at supporting clinical ward rounds bring this with: First, a ward round is prepared (Task T1); i.e., the doctor scans patient information. Following these initial preparations, the doctor visits the patients and communicates with them, asking for information about their health status (Task T2). In turn, this information is written down by a nurse (Task T3). Afterwards, the patient is examined (Task T4). In the context of this task, blood values are analyzed and decisions about the further diagnostic procedure are made. Then, the doctor creates medical orders (Task T5). Finally, a nurse updates the patient information and triggers additional medical orders (Task T6).

8.1.2 Implementation

iCare is a web-based semantic Java application. Its implementation is based on the semantic middleware iQser GIN Platform 1.6 [171], the build automation tool Maven 2.2.1, the web framework Wicket 1.5.6, the JavaScript library jQuery 1.72, the database MySQL 5, the text search engine library Lucene 2.4, HTML5, and CSS3. Figure 8.1 shows the start screen of iCare.

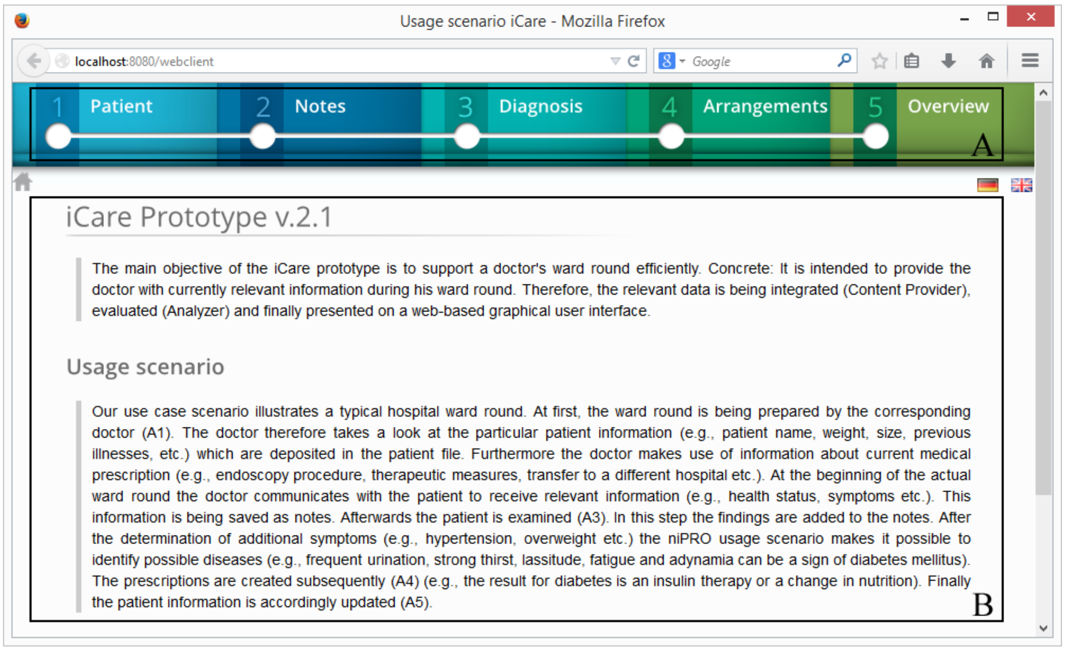

Figure 8.1: iCare: start screen.

The iCare user interface is divided into two parts: (1) *process view* (cf. Figure 8.1A) and (2) *information view* (cf. Figure 8.1B). The former visualizes the currently executed process (i.e., a ward round), whereas the latter shows the medical information needed in the current context (e.g., patient records, lab reports, medical orders, or minutes).

Features. First, iCare allows for the *integration* of structured, semi-structured, and unstructured medical information from different data sources. Second, it enables the syntactic and semantic *analysis* of medical information to automatically discover relationships between information, which have been unknown so far. In turn, from these newly discovered relationships, medical staff can derive medical knowledge. Third, iCare enables the context-aware *delivery* of needed medical information to medical staff. Thus, it provides a central point of access and a homogeneous view on medical information.

The focus of iCare is on *information-* and *context-awareness*; e.g., the discovery of relationships between patient records, medical orders, minutes, diseases, and therapy options as well as their contextualized delivery to medical staff. In turn, process-awareness is only addressed to a limited extent; i.e., through the visualization of executed business processes and corresponding medical information (cf. Figure 8.1).

Architecture. The iCare prototype is based on a 4-tier architecture comprising: (1) *data layer*, (2) *semantic layer*, (3) *context layer*, and (4) *application layer*.

The *data layer* deals with the data sources to be integrated (e.g., *hospital information systems* (HIS), medical databases, digital libraries, and electronic health records). For each data source, a *ContentProvider*[3] is implemented. Its main task is to transform proprietary medical information into a homogeneous information format (i.e., information objects). Note that this is a prerequisite for the subsequent analysis (see below).

The *semantic layer* is responsible for the syntactic and semantic analysis of the medical information. For this purpose, we apply the semantic middleware iQser GIN Platform [171]. For each analysis, an *AnalyzerTask*[4] is implemented. Thereby, syntactic and semantic analyses are performed in several steps: First, basic properties of the integrated information (e.g., authorships) are compared (*syntactic analysis*). For example, this allows us to link information with the same author (e.g., a specific doctor). Second, the content of the available medical information is analyzed (*semantic analysis*). For this purpose, algorithms from the fields of data mining, text mining (e.g., text and linguistic preprocessing, clustering, classification, information extraction), pattern-matching, and machine learning are applied [182]. The goal of the semantic analysis is to further classify and group correlated medical information. Third, the user behavior is investigated, e.g., the frequency of using particular information in the context of a clinical ward round. As result of these analyses, we obtain a corresponding SIN. In particular, the SIN allows discovering inter-linked medical information, e.g., information dealing with the same topic (e.g., flu, diabetes) or needed in the context of a particular process task [7].

The *context layer* is responsible for integrating and analyzing context information (e.g., used device, location, time, and user behavior). Context information is gathered from

[3]A ContentProvider is an interface for the integration of structured or unstructured data. It supports bidirectional data access and can have functions for data processing [196].
[4]An AnalyzerTask is an interface for the analysis of integrated data [196].

data sources called *sensors*. A CM is created based on the available context information and allows characterizing a doctor's work context. In turn, the latter can be used to filter the SIN. Note that the CM is completely independent from the SIN; i.e., context information (i.e., context objects) is only stored in the CM and not in the SIN.

Finally, the *application layer* deals with the personalized delivery of medical information. In particular, the application layer is responsible for the common presentation of executed processes (and process tasks) as well as related medical information.

8.1.3 Scenario Support

We sketch how the scenario from Section 8.1.1 can be addressed with iCare. To support Task T1, a search box is offered to the doctor to select a patient (cf. Figure 8.2A). After completing this selection, iCare provides specific information such as name, gender or previous diseases based on the selected patient record (cf. Figure 8.2B).

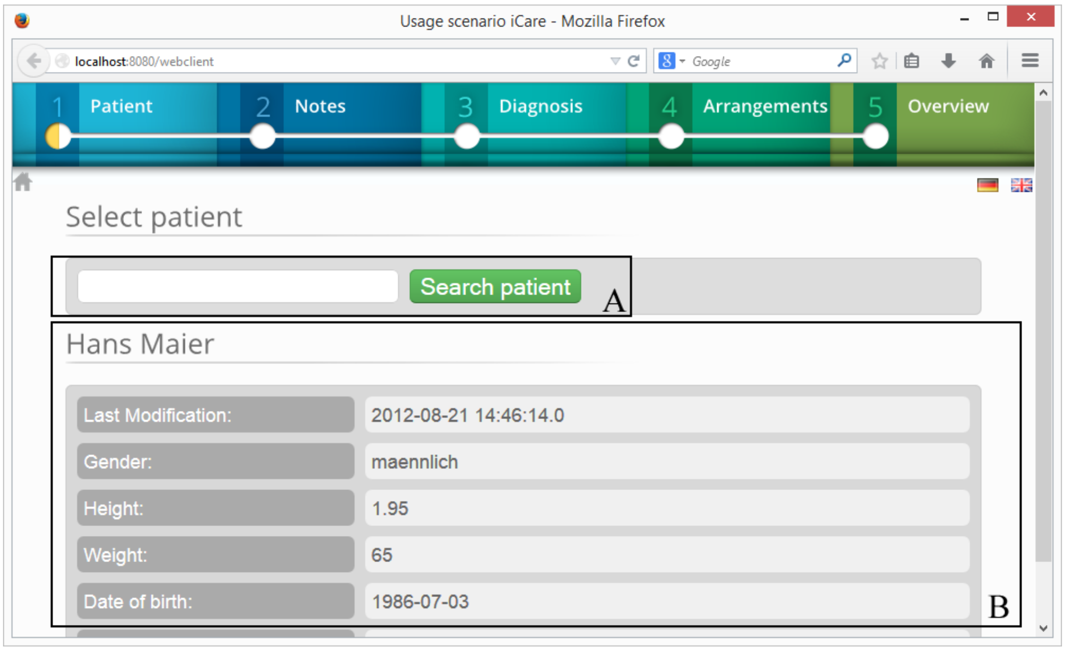

Figure 8.2: iCare: task T1.

When performing Task T2, existing medical notes on the selected patient are displayed. If required, the doctor may add, update or delete medical notes (related to the patient's health status). Medical notes are one of the inputs for the subsequent analysis.

Based on the analysis of medical information, potential diagnoses as well as related treatment options are automatically determined during Task T3. The analysis takes the patient record, medical notes, and information from medical libraries (i.e., Onmeda[5] [237]) into account. For example, iCare might conclude that sore throat, croaki-

[5]Since we have no access to international digital medical libraries we use the German health portal Onmeda (http://www.onmeda.de) instead. Therefore, some screenshots contain German text.

ness, rheumatic pains, and absence of appetite are potentially caused by the disease flu (cf. Figure 8.3A). Note that this conclusion is based on the semantic analysis which automatically determines the text similarity between medical information (cf. Figure 8.3B). The higher the value, the higher is the similarity between information.

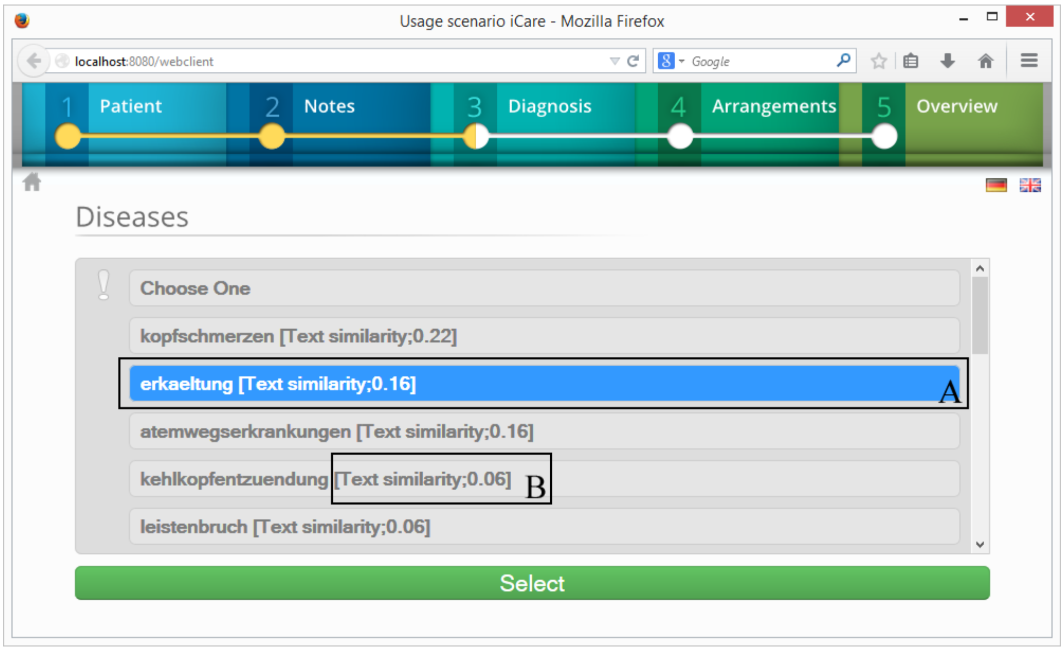

Figure 8.3: iCare: task T3.

As an additional result, the doctor is supplied with treatment options that can be automatically discovered as well. If a treatment option is selected, a more detailed treatment description and respective instructions are displayed by iCare.

In Task T4, the doctor may then add, update or delete medical orders. Finally, the patient record, medical notes, and medical orders are visualized in Task T5.

8.1.4 Related Work

There exist approaches related to iCare. Medical guidelines, for example, have been intensively discussed in recent years as an approach to support medical decision making [20, 238]. Burgers et al. discuss and compare the structure of medical guidelines [239]. In turn, Fervers et al. conduct a survey on the adoption of guidelines in healthcare practice [240]. The exchange and representation of medical guidelines in a machine-interpretable way is addressed by Ohno-Machado et al. [241] and Fox et al. [242]. Their work has been of great importance when developing iCare and its underlying SIN (which also integrates medical guidelines in a machine-interpretable way). Though medical guidelines address medical decision-support, however, they cannot be directly compared with iCare. Instead, medical guidelines may be considered as an additional source of information for iCare (documenting well-established medical procedures).

Moreover, a variety of tools, platforms and solutions have been introduced to support medical decision making. We briefly sketch important approaches. For example, HIS enable the management of administrative as well as medical information [243]. Unlike a HIS, iCare is a more specific application focusing on a specific use case (i.e., the clinical ward round). Other applications deal with the management of electronic medical records, e.g., GNUmed [244], GNUHealth [245], OpenEMR [246], and FreeMED [247]. However, note that these approaches neither provide a semantic analysis of medical information nor its context-aware delivery to medical staff.

Altogether, iCare supports the ward round by reducing the time needed for searching and handling medical information, taking the work context of medical staff into account.

8.2 Automotive Domain: iGraph

The amount of information, engineers in the automotive domain are confronted with, makes it a challenging task for them to identify and handle the exact information needed during daily work [17, 248]. When reviewing product requirements, for example, engineers not only must consider e-mails and office files, but also guidelines and manuals [2]. This information may be accessed through shared drives or enterprise portals.

However, engineers are not only interested in quickly accessing needed information, but additionally require complete, up-to-date and aggregated information, e.g., when conducting a review. iGraph provides an approach that may relieve automotive engineers from this task. Its overall goal is to deliver needed documents (cf. Figure 8.4A) as well as to discover semantically related ones (cf. Figure 8.4B).

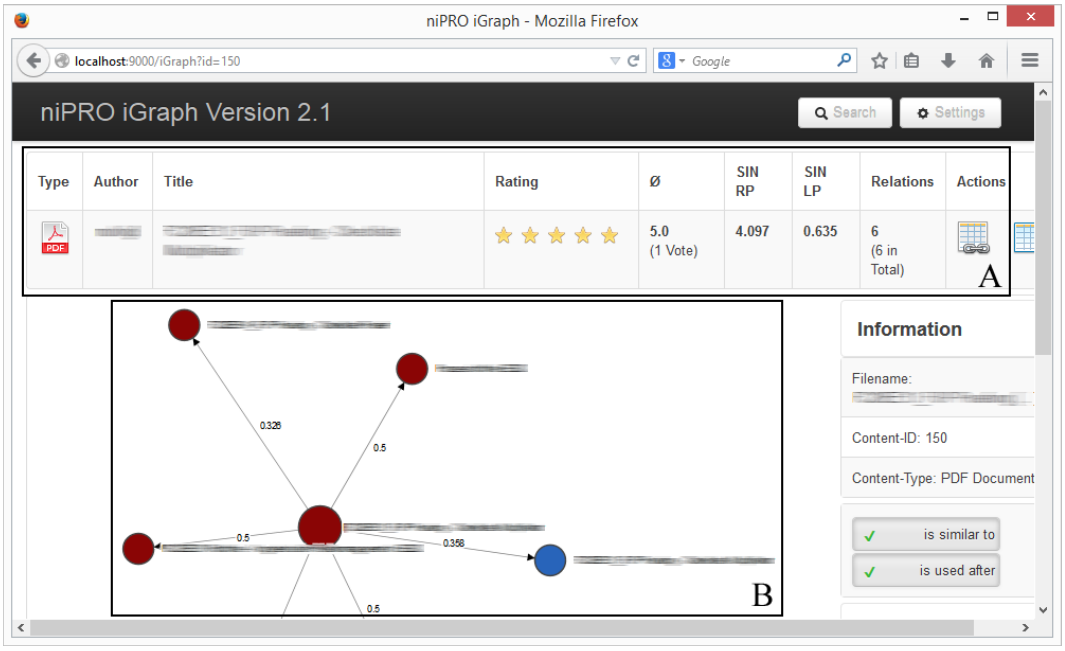

Figure 8.4: iGraph: discovering related documents.

8.2.1 Scenario

We consider a scenario dealing with the review of product requirements, which have been documented as functional specifications. The goal is to both improve and approve the respective specifications. Furthermore, the review process is knowledge-intensive comprising large amounts of process information, user interactions (e.g., "perform review meeting"), and involving decision making (e.g., shall the document be approved or not?). Three roles are involved in the process: The *author* provides the specification to be reviewed. The *review moderator* organizes the review meetings. Finally, the *reviewer* analyzes the provided specification and records errors, ambiguities and uncertainties [3].

8.2.2 Implementation

We realized iGraph[6] as a web-based Java application based on the semantic middleware iQser GIN Server 2.0 [171], the web framework Play 2.1.1, the web engine Bootstrap 2.3.1, the JavaScript library jQuery 1.8.3, the database MySQL 5, the text search engine library Lucene 2.4, the JavaScript library D3 3.1.1, HTML5, and CSS3.

Features. First, iGraph enables the comprehensive *integration* of process information and business processes from heterogeneous data sources. Second, it allows for the syntactic and semantic *analysis* of integrated information and process objects. Finally, it enables the process-oriented *delivery* of needed process information and business processes to knowledge workers and decision makers when conducting a review.

In particular, iGraph addresses *information-* and *process-awareness*; i.e., it integrates processes (e.g., review process) and related process information (e.g., reviews, templates).

Architecture. iGraph implements all architectural layers of POIL except the context layer; i.e., (1) *data layer*, (2) *semantic layer*, and (3) *application layer*.

The *data layer* concerns the set of data sources to be integrated. For each data source, a ContentProvider is implemented. Its main task is to transform proprietary process information or business processes into uniform information and process objects. In turn, the *semantic layer* is responsible for the syntactic and semantic analysis of information and process objects. For this purpose, we apply the semantic middleware iQser GIN Server and implement several AnalyzerTasks. The goal is to classify and group correlated objects (e.g., filled-out review templates). Finally, user behavior is investigated, e.g., the frequency of using particular information in the context of specific process tasks. Finally, the *application layer* deals with the delivery of process information.

8.2.3 Scenario Support

In the following we show how iGraph can be applied to the sketched scenario. More precisely, we consider one process schema[7], three process instances[8], and about 300 doc-

[6] A screencast presenting the iGraph application is available at http://nipro.hs-weingarten.de/screencast.
[7] The process schema is modeled with Signavio Process Editor [249].
[8] The process instances are managed with the Activiti BPM Platform [250].

uments[9] (i.e., process information) such as reviews, review templates, manuals, minutes, presentations, and guidelines. Particularly, we pick up a specific task of the review process: the preparation of a functional specification for a review.

The author of the specification wants to identify relevant information objects supporting the review preparation. For this purpose, iGraph provides a search box. The reviewer enters a query (e.g., "review template") into the search box and executes it. Search results are then listed in a table-based view (cf. Figure 8.5). Thereby, each row corresponds to a search result (i.e., an information object), whereas each column contains detailed metadata about the information objects from the result set, e.g., its type, author, title, rating, or number of semantically related documents.

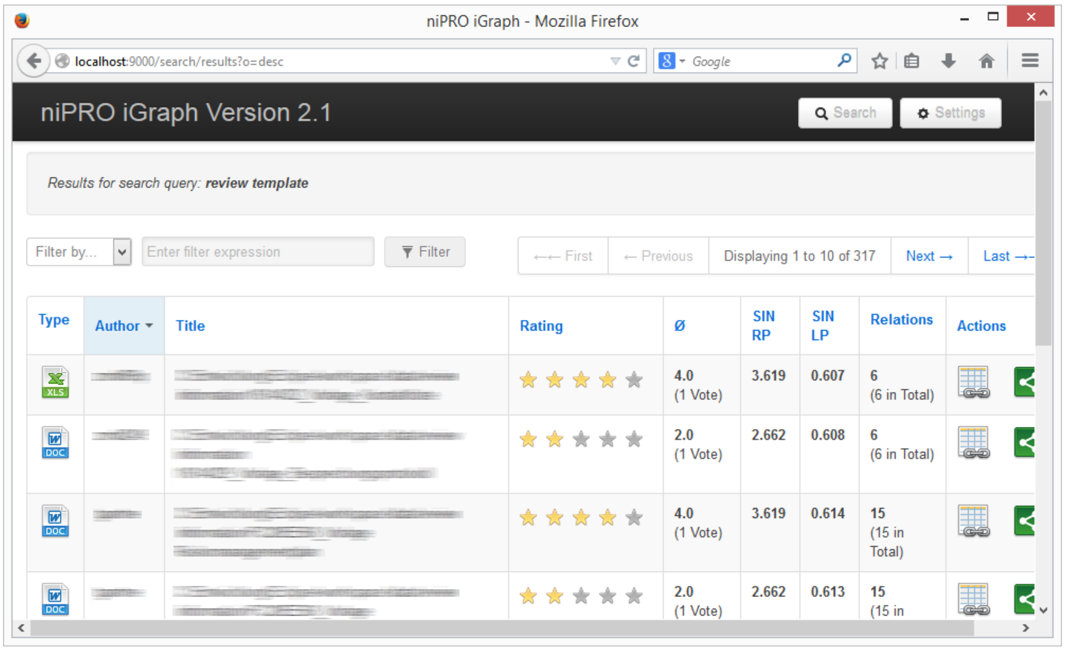

Figure 8.5: iGraph: table-based view.

To identify related information objects (e.g., objects addressing the same topic), iGraph provides a graph-based view (cf. Figure 8.6). The latter displays both related information and process objects starting from a specific information object (e.g., a template). Then, the user can freely navigate through the related documents in the SIN.

In order to quickly identify relevant objects in the SIN, iGraph provides two fundamental key indicators: (1) *link popularity* and (2) *rate popularity* (cf. Section 5.5).

8.2.4 Related Work

Besides iGraph, there exist other applications enabling IL, e.g., in fields like wearable computing [114], weather forecast [115], and healthcare [99]. Moreover, a large num-

[9]These documents are real-world documents and stem from our automotive research partner.

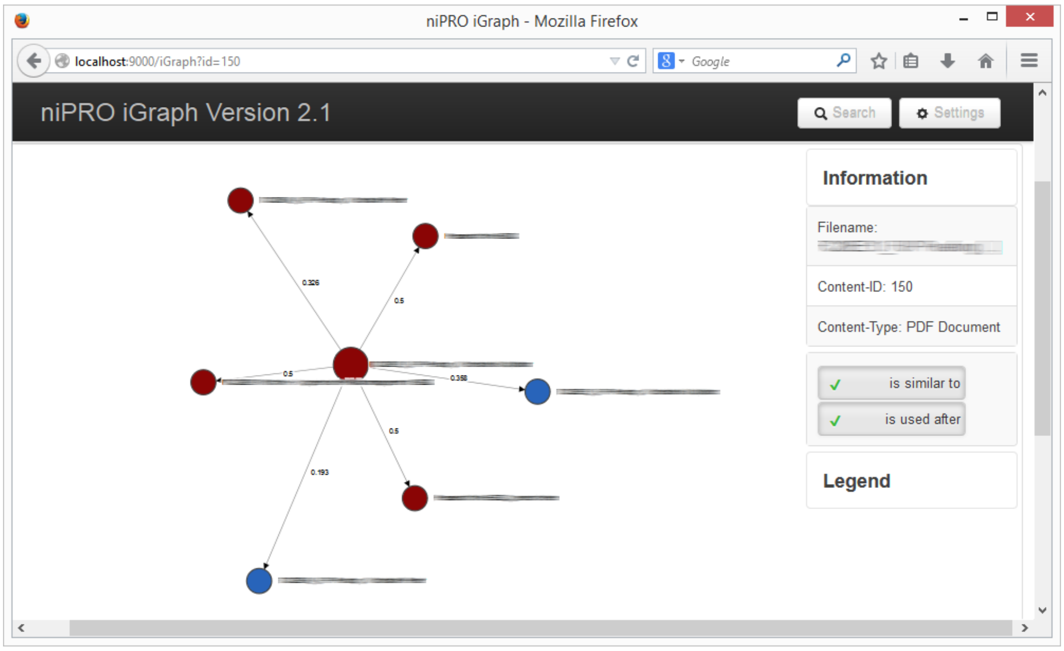

Figure 8.6: iGraph: graph-based view.

ber of tools and solutions have been introduced to discover and visualize related documents. For example, enterprise search engines, such as Solr [251], Elasticsearch [252] and OpenSearchServer [253] allow discovering related documents. In turn, Datameer [254] and Lumify [255] deal with the visualization of relationships between information.[10] Note that none of these approaches includes a syntactic and semantic analysis of business processes and the alignment of the latter with respective process information.

Altogether, iGraph applies semantic technology to enable the integration, analysis and delivery of process information to process participants. The table- and graph-based views as well as the use of key indicators (i.e., *link popularity* and *rate popularity*) make it easy to identify relevant process information during business process execution.

8.3 Scientific Domain: iPub

The quantity of available literature as well as the diversity of applications used in scientific research (e.g., academic search engines, reference management software) make it a challenging task for researchers to handle relevant literature. One of the main problems in this context is to identify related work. To cope with this challenge, academic search engines, such as Google Scholar and Microsoft AS, analyze the co-occurrences of citations within documents to identify related work [257]. However, this approach is not sufficient enough as it lacks the consideration of semantic aspects.

[10] An overview about graph visualization and navigation in information visualization is provided, for example, by Herman et al. [256].

8.3.1 Scenario

The iPub scenario deals with the discovery of related scientific work. Further, it supports the discovery of authors and topics during a literature research. Among others, the following use cases are relevant in this scenario: the user is looking for articles matching a search query (Use Case UC1), the user is looking for articles of a specific author (Use Case UC2), the user is looking for articles dealing with a specific topic (Use Case UC3), the user is looking for authors who published articles dealing with a specific topic (Use Case UC4), the user is looking for co-authors of a specific author (Use Case UC5), and the user is looking for semantically related articles to an existing one (Use Case UC6).

8.3.2 Implementation

We implemented iPub as a web-based Java application based on the iQser GIN Server 2.0, the web framework Play 2.1.0, the web engine Bootstrap 2.3.1, the JavaScript library jQuery 1.9.0, the database MySQL 5, the text search engine library Lucene 2.4, HTML5, and CSS3. Furthermore, we use the website SpringerLink [258] and the computer science bibliography DBLP [259] as data sources for iPub in order to retrieve scientific articles.

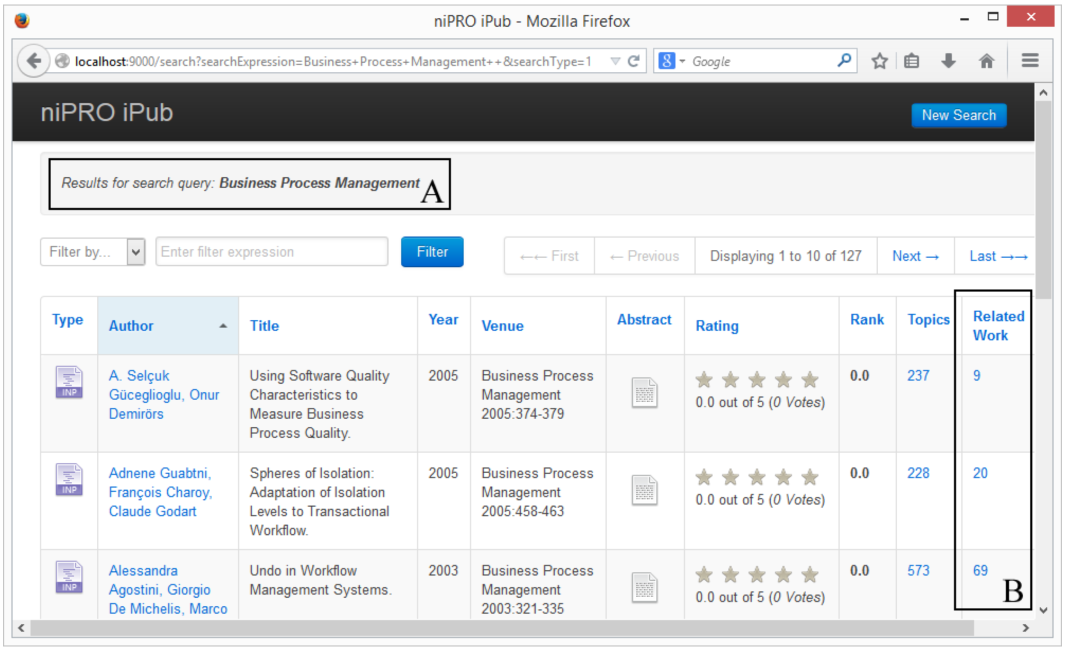

Figure 8.7: iPub: article-based view.

Features. First, iPub enables the *integration* of electronically available scientific articles from different data sources. Second, it allows for the syntactic and semantic *analysis* of articles in order to discover relationships between them as well as between authorships and topics. Third, iPub enables the *delivery* of articles to researchers. Finally, it represents a central point of access and unified views on articles, authors and topics.

iPub focuses *information-awareness*; i.e., the identification of related work (i.e., relevant articles) as well as relevant authors and topics. For example, Figure 8.7A shows the results for the search query "Business Process Management".

Architecture. iPub implements the same architectural layer as iGraph: (1) *data layer*, (2) *semantic layer*, and (3) *application layer*. Additional ContentProviders and AnalyzerTasks are implemented as well [257]. For example, iPub implements ContentProviders crawling the abstracts of scientific papers from SpringerLink [258].

8.3.3 Scenario Support

In the following, we illustrate the use of iPub. Use case UC1 has been already illustrated by Figure 8.7A. To realize UC2, in the author-based view the user can click on the number of articles (cf. Figure 8.8A) to get all published articles of the selected author.[11]

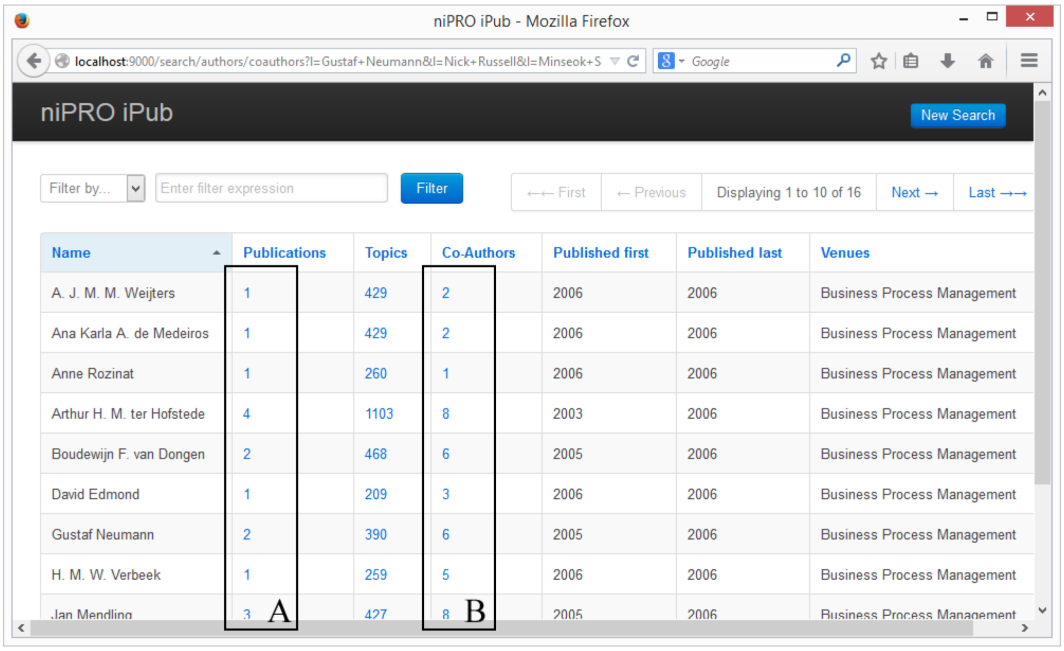

Figure 8.8: iPub: author-based view.

To support use case UC3, in the topic-based view of iPub the user can click on the number of articles (cf. Figure 8.9A) in order to get all articles dealing with a specific topic (e.g., BPM, IM, POIL). Similarly, use case UC4 can be supported by clicking on the number of authors to get all authors who published articles dealing with a specific topic (cf. Figure 8.9B). To realize use case UC4, the user can click on the number of

[11] Note that we only consider articles published in the conference proceedings of BPM'03-BPM'06.

co-authors to get all co-authors of the selected one (cf. Figure 8.8B). Finally, to enable use case UC5, the user can click on related work (cf. Figure 8.7B) to get semantically related work for a specific article. Note that this is possible in the topic-based view as well (in order to get semantically related topics for a specific topic).

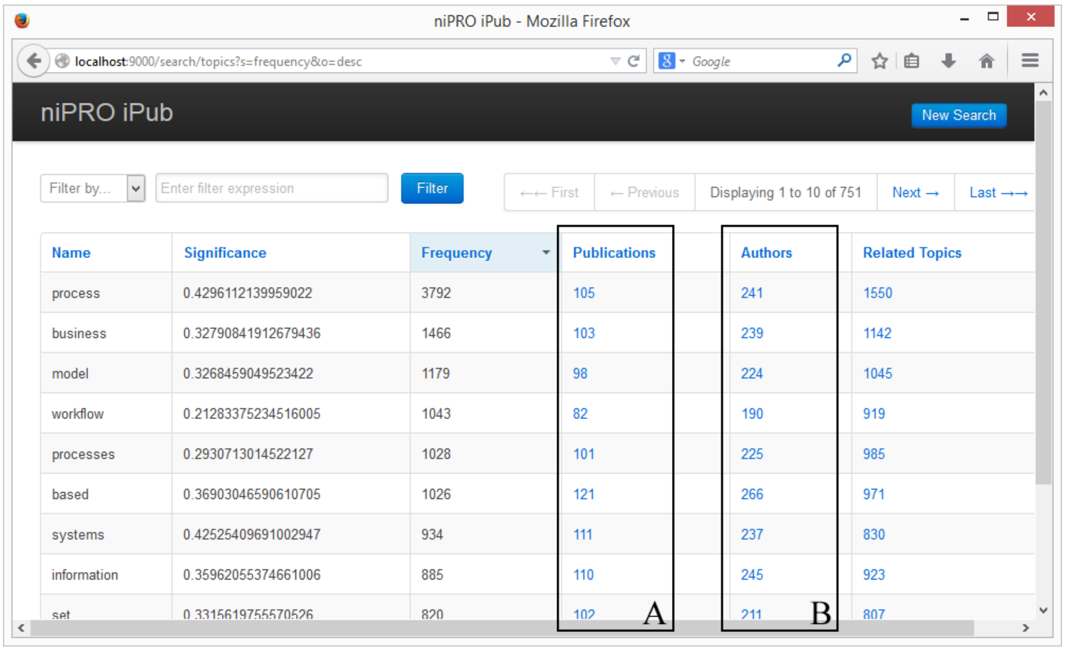

Figure 8.9: iPub: topic-based view.

8.3.4 Related Work

In addition to the iPub prototype, there exist applications that allow researchers to identify related work, e.g., search engines like Google Scholar [260], Microsoft AS [261], SpringerLink [258], CiteSeerX [262], and ScienceDirect [263]. In turn, other solutions such as ArnetMiner [264] deal with the search in academic social networks. Note that none of these approaches provide a classification in articles, authors and topics. Moreover, semantic analyses are only realized up to a certain degree; i.e., existing approaches lack semantic aspects such as the semantics of the content.

Altogether, iPub is a comprehensive application applying semantic technology to enable the identification of related articles, authors and topics [257].

8.4 Enabling Process Navigation based on POIL

Process participants typically have different perspectives on business processes and their tasks. For example, a knowledge worker is mainly interested in detailed information (e.g., process description, duration) about the process task he is currently working on.

In turn, a decision maker needs a more abstract visualization of business processes in order to get an overview of currently running process tasks [265, 266, 267].

Thus, to support process participants when accessing business processes and their tasks in a personalized way, a user-oriented process navigation and visualization approach for business processes is needed. In particular, process navigation shall allow users to switch between different levels of detail (e.g., detailed or abstract view) as well as to select different visualizations of business processes (e.g., time- or logic-based view).

This section sketches the *process visualization and navigation* (ProNaVis) framework developed by Hipp et al. [12, 167]. It allows process participants to flexibly navigate within process model collections as well as single process models on different levels of detail and forms of visualization [9]. In order to enable both flexible navigation and visualization, ProNaVis relies on POIL concepts in general and on the SIN in particular. The latter not only enables a generic representation of business processes (cf. Figure 8.10), but allows determining relationships between the elements of a business process as well. These relationships are then used by ProNaVis to enable users to navigate in process model collections (e.g., show all roles of a business process).

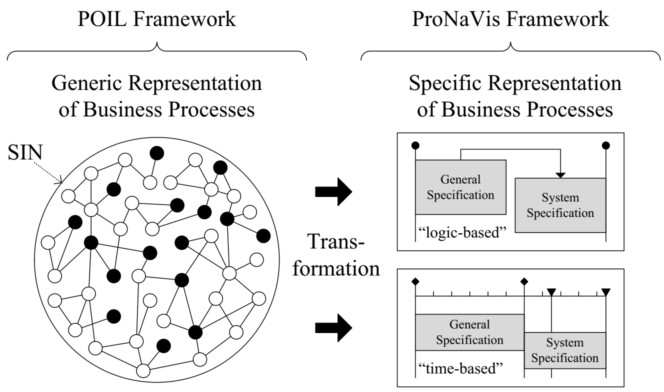

Figure 8.10: Generic and specific representation of business processes.

8.4.1 Scenario

We consider business processes dealing with the electrical/electronic development of car control units.[12] In contemporary process portals only little process navigation support is available. Business processes are documented in terms of process diagrams, which, in turn, are captured in documents (e.g., PDF documents). Furthermore, business processes are categorized into process areas according to their topics (e.g., development, quality). Finally, each process area is depicted as a static image map [167].

[12]Car control units are embedded systems that controls one or more electrical systems in a car [248].

8.4.2 The ProNaVis Framework

This section sketches the ProNaVis framework, which allows process participants to access business processes on different levels of detail, visualizing business processes in different forms, and focusing on specific areas of interest. In order to enable such a flexible process navigation both on different levels of detail and forms of visualization, providing a simple zooming function for business processes is not sufficient [12, 268].

In detail, the ProNaVis framework comprises three dimensions: (1) *semantic dimension*, (2) *geographic dimension*, and (3) *visualization dimension.* The semantic dimension allows visualizing business processes on different levels of detail, whereas the geographic dimension only allows for visually focusing on a certain area of the screen. Finally, the visualization dimension allows changing the way a business process shall be visualized (e.g., as logic- or a time-based view). Most existing research has only considered one single or the combination of two of the aforementioned three dimensions. In turn, the ProNaVis framework supports three independent navigation dimensions, enabling a user-triggered navigation in complex business processes and thus representing a new generation of process navigation and visualization support [12].

In the *semantic dimension*, business processes may be displayed in different levels of detail (cf. Figure 8.11). On a high semantic level, for example, only the names (e.g., "general specification") of processes are displayed. In turn, on a low semantic level, the names and relationships of sub-processes in a process are displayed as well.

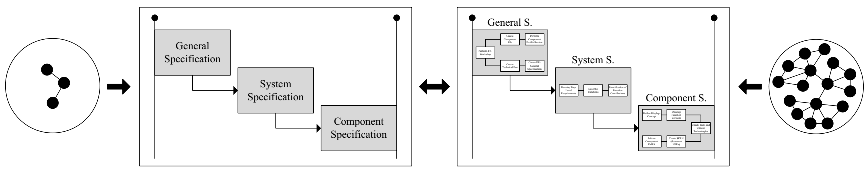

Figure 8.11: Semantic dimension of ProNaVis.

In turn, the *geographic dimension* allows for visually zooming without need to change the level of detail (cf. Figure 8.12). Think of a magnifier while reading a newspaper.

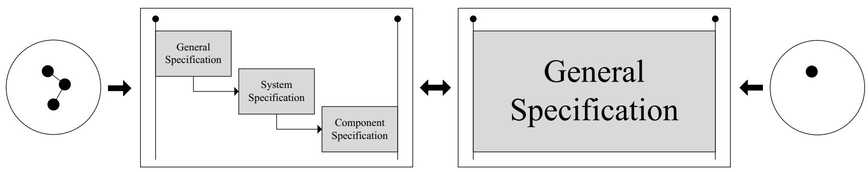

Figure 8.12: Geographic dimension of ProNaVis.

Finally, the *visualization dimension* enables process participants to select among different forms of visualization (cf. Figure 8.13). For example, a logic-based visualization emphasizes logic relationships (e.g., sequence flows) between process tasks, whereas a time-based visualization, deals with time aspects and uses a time line for visualization.

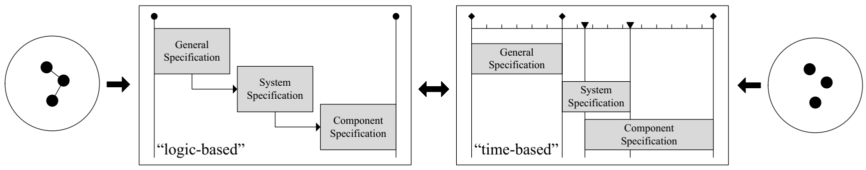

Figure 8.13: Visualization dimension of ProNaVis.

Altogether, ProNaVis is a flexible navigation framework allowing users to flexibly navigate within business processes on different levels of detail, zooming levels and visualizations. In the following, we describe how the POIL framework supports ProNaVis.

8.4.3 Scenario Support

ProNaVis provides an advanced *navigation element* to switch between different levels of detail as well as to select different visualizations of business processes and their tasks. The navigation element is created based on available process objects and relationships from the SIN. Figure 8.14 depicts a schematic navigation element which comprises a slider to change the geographic dimension G, check boxes to choose different levels of the semantic dimension S and radio buttons to switch between different forms of visualization in the visualization dimension V. Note that, for example, the number of available levels of detail in the semantic dimension S depends on the SIN; i.e., if process objects in different granularity levels are available in the SIN then the user may select different levels in the semantic dimension S. In turn, if process objects are only available on a coarse-grained level, the user cannot select between different levels. Thus, ProNaVis depends on the core component of POIL; i.e., the SIN.

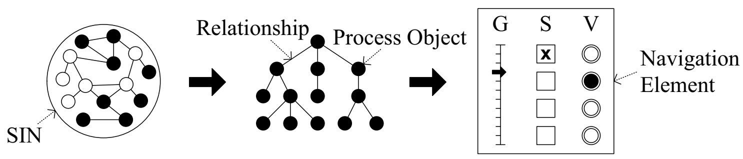

Figure 8.14: Creation of the ProNaVis navigation element.

The following example shows the results of user interactions with the navigation element as shown in Figure 8.14. Each change of a dimension (e.g., G, S and V) in ProNaVis results in a specific navigation state.

Navigation State 1. Process navigation starts with creating a structured representation for a collection of business processes based on the SIN (cf. Figure 8.15A). In this context, different business processes are visualized as grey boxes. The geographic level corresponds to level 1; i.e., all processes are shown. The semantic dimension visualizes business processes as abstract grey boxes (semantic level 3). In turn, this visualization is a time-based one; i.e., business process duration is represented through the lengths of each box. Note that the duration can be derived from the SIN as well; i.e., by using properties "start" and "end" of each process object.

Navigation State 2. Assume that an engineer is solely interested in his current process task. Then, he may select semantic level 4 to see all included process tasks. This interaction results in navigation state 2 (cf. Figure 8.15B), which displays all process tasks (semantic level 4) in combination with the associated processes (semantic level 3).

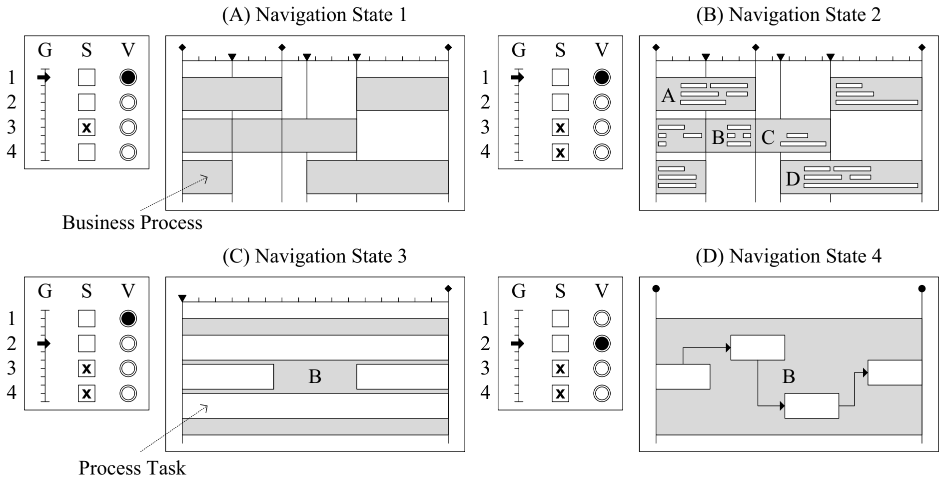

Figure 8.15: Process navigation in three navigation dimensions.

Navigation State 3. As the requirements engineer is interested in a specific process task of process B, he applies the geographic dimension to process B and reaches navigation state 3 (cf. Figure 8.15C). Note that all mentioned interactions are user-driven; i.e., triggered by a user interaction with the navigation element.

Navigation State 4. Finally, assume that the engineer is less interested in time aspects, but wants to know what he has to do next after finishing his current task. Therefore, he switches to the logic-based visualization. In turn, this interaction results in navigation state 4 (cf. Figure 8.15D). Using this visualization, he may quickly identify predecessor and successor relationships of the tasks involved. In particular, these relationships are provided by the SIN as well; i.e., relationships of type "sequence" (cf. Figure 8.16).

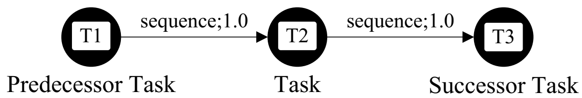

Figure 8.16: Predecessor and successor of a task.

The presented example sketches the ProNaVis navigation concept and shows how process participants may flexibly navigate in complex business processes and associated tasks based on three independent dimensions created by the SIN.

8.4.4 Related Work

Research on the flexible visualization of business processes (i.e., on what we call view dimension) is presented by Bobrik et al. [265] and Smirnov [269]. More specifically, these approaches introduce aggregation and reduction techniques to create flexible views on complex business processes. Kolb et al. present a view framework allowing for updatable and user-centered process views as well as the user-centered modeling and visualization of business processes [270, 271]. In turn, Kabicher-Fuchs et al. focus on timeline visualizations for documenting and visualizing continuously changing business processes [272]. Note that all these approaches solely deal with business processes, while the combination of business processes and related process information as well as the navigation in respective process spaces are neglected.

Challenges related to zooming functionality in user interfaces are addressed by Reiterer and Büring, who present zoomable user interfaces for navigating in complex information spaces [273]. The JAZZ framework [274] applies these concepts. Corresponding user interface concepts include Squidy [275], ZOIL [276] and ZEUS [277]. Zooming and moving in a 3D environment is realized by the Flight Navigator tool [278], which supports numerous interaction paradigms enabling users to present, inspect and analyze business process models in a 3D environment. Similarly, Brown et al. use 3D technology to realize a collaborative approach for modeling business processes [279, 280]. An approach for efficient zooming is presented by Wijk and Nuij [41].

Finally, there exists research on the provision of information on different levels of detail (i.e., on what we call semantic dimension). Both Seyfang et al. [281] and Shneiderman [282] make use of process hierarchies in order to efficiently visualize complex process models on a small canvas. Respective approaches allow displaying large process hierarchies in their entirety in a compact manner and thus facilitate the presentation of information on different semantic levels. Furthermore, Misue and Yazaki discuss the representation of detailed information about a single activity without losing the overview of the global structure of an organization [283]. Finally, this approach provides a representation technique embedding charts into cells of a tree map.

8.5 Lessons Learned

Based on the implementations of our prototypes, we gathered knowledge when realizing POIL. Based on these practical experiences, we derived the following lessons learned:

- When creating the SIN, business processes should be split up into their constituent elements (e.g., tasks, events). Each process element should be treated as a single process object. Therefore it is possible to align process information with business processes on a fine-grained granularity level (e.g., for a particular process task) and not only on a coarse-grained one (e.g., for an entire business process).

- After creating the SIN's first stage of expansion, process information should be integrated in different granularity levels, ranging from fine-granular information (e.g., database tuple, single-page office document) to coarse-granular information (e.g., database table, multi-page office document). Reason is that aggregation, interpolation or reduction of objects in the SIN is cost-intensive in terms of time.

- When analyzing integrated process and information objects in the SIN, usually, a large number of relationships are created by the semantic analysis. Therefore, a threshold for relation weights should be defined in order to avoid creating weak relationships; i.e., objects are similar only to a small extent. The less relationships exist in a SIN, the better is the performance of the SIN.

- When analyzing the metadata of objects in the SIN, it is essential that not all metadata is considered in the syntactic analysis. For example, it does not make sense to relate information objects with the same file format. Depending on the use case, metadata to be analyzed must be well chosen.

- When querying the SIN, the key indicators link and rate popularity are fundamental when determining the relevance of objects. The empirical validation confirmed that most of the objects returned by our algorithms are indeed relevant.

8.6 Summary

This chapter presented various proof-of-concept prototypes demonstrating the applicability of the POIL framework. First, we described iCare, an application focusing on the contextualized delivery of medical information to medical staff. Using iCare, medical staff needs not search for medical information anymore, but is automatically supplied with relevant medical information depending on the current work context. In addition, we presented iGraph, an application aiming at the delivery of currently needed documents as well as the identification of semantically related documents during the preparation of a functional specification for a review. Finally, iPub was presented, an application dealing with the identification of scientific work as well as supporting the identification of authors and topics during a literature research. In summary, Figure 8.17 shows a classification of the applications along the problem dimensions of this thesis.

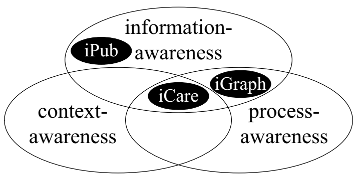

Figure 8.17: Applications along problem dimensions.

Moreover, this chapter introduced the ProNaVis framework, a navigation and visualization framework allowing process participants to flexibly navigate within business processes on different levels of detail, zooming levels, and forms of visualization. In order to enable such a navigation and visualization, ProNaVis uses POIL concepts, particularly on the business processes and relationships represented by the SIN.

9 Empirical Validation

This chapter empirically validates the POIL framework. Section 9.1 describes the research design. Section 9.2 then presents a survey verifying the applicability of the SIN LP and SIN RP algorithms. Section 9.3, in turn, presents a case study on the applicability of the SIN maintenance algorithms. Section 9.4 verifies the applicability and feasibility of the POIL framework in a controlled experiment. Finally, Section 9.5 summarizes the results of the chapter.

9.1 Research Design

The empirical validation comprises three parts (cf. Figure 9.1). First, we performed a survey in the automotive domain (Part 1). Second, we conducted a qualitative exploratory case study in the same domain based on face-to-face interviews and paper-based questionnaires (Part 2). Third, we conducted a controlled experiment in the agricultural domain based on a proof-of-concept prototype (Part 3).

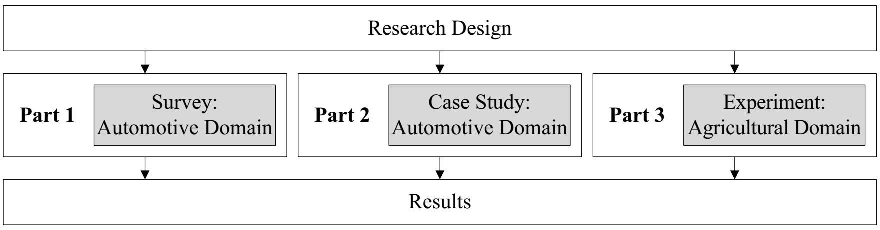

Figure 9.1: Parts of the empirical validation.

Part 1: Survey. The survey was performed in April 2013. It was conducted with a web questionnaire comprising 18 questions. Overall, 20 experts from an automotive *original equipment manufacturer* (OEM) participated. Most of the experts work at departments dealing with the engineering of electric/electronic car components. However, staff from other departments participated as well. All participants were selected according to their expert knowledge regarding the considered use case (cf. Figure 9.2). No participant was a member of the research team.

Part 2: Case Study. The case study was performed in July 2014. It was conducted based on a questionnaire with 60 questions. Overall, eleven employees from an automotive OEM were interviewed. The interviewees work in different areas (e.g., development, management) of their organization. Both knowledge workers and decision makers were involved. In the case study, data was gathered through face-to-face interviews following a semi-structured interview guideline. After each interview, an additional questionnaire had to be filled out. Each interview lasted about 90 minutes.

Part 3: Experiment. The controlled experiment was performed in February 2015. A part of the experiment constituted of face-to-face interviews. Furthermore, an additional questionnaire had to be filled out to collect other data. In the experiment, participants had to solve tasks using a prototype (cf. Section 9.4.1). Overall, twelve employees from an enterprise operating in the agriculture, building materials, and energy sector participated. None of the interviewed participants was a member of the research team.

9.2 Survey: Automotive Domain

In order to prove that the SIN LP and SIN RP algorithms (cf. Sections 5.5.1 and 5.5.2) actually support process participants involved in knowledge-intensive business processes, we applied the algorithms to a real-world use case from the automotive domain (cf. Section 9.2.1). Specifically, we implemented the algorithms (cf. Section 9.2.2) and then compared their outcome with the results of a survey among experienced automotive engineers. The latter were asked to manually rate the relevance of process information related to the considered use case based on their own experiences (cf. Section 9.2.3). Results indicate that the algorithms can indeed replace the costly and time-intensive human determination of relevant process information (cf. Section 9.2.4).

In particular, the survey has been guided by two research questions (cf. Table 9.1):

#	Research Questions
RQ1.1	How do the results of the SIN LP algorithm match with user-generated evaluations on the relevance of process information?
RQ1.2	How good is the ranking of process information based on the SIN RP algorithm compared to other ranking approaches?

Table 9.1: Research questions underlying the survey.

9.2.1 Use Case

The considered use case deals with the review of product requirements as documented in functional specifications at a large automotive OEM. The goal is to improve as well as to approve such specifications. The corresponding review process is knowledge-intensive since it requires large amounts of process information (e.g., protocols, checklists, guidelines, manuals, and review results), user interactions (e.g., "perform review meeting", "send review comments"), and decision-making (e.g., shall the document be approved or not?). Three roles are involved: (1) the *author* provides the specification to be reviewed, (2) the *review moderator* organizes the review meetings, and (3) the *reviewer* analyzes the provided specification and records errors, ambiguities and uncertainties.

The review process (cf. Figure 9.2) starts with the preparation of the document to be reviewed (Task T1). This task is performed by the author of the document. Based on this initial preparation, the author decides whether or not a preliminary review meeting becomes necessary (Task T2). Afterwards, the document may be reviewed (Task T3). Based on the outcome of the review, the reviewer decides whether an additional review meeting is needed (Task T5) or whether it is sufficient to directly send findings and comments to the author (Task T4). The latter then evaluates review results (Task T6) and updates the document accordingly (Task T7). If the overall review status of the document is "rejected", it will not be approved. In turn, if the overall review status is "accepted", the author may finally approve the document (Task T8). For each of these tasks, a variety of process information is needed, e.g., guidelines and templates.

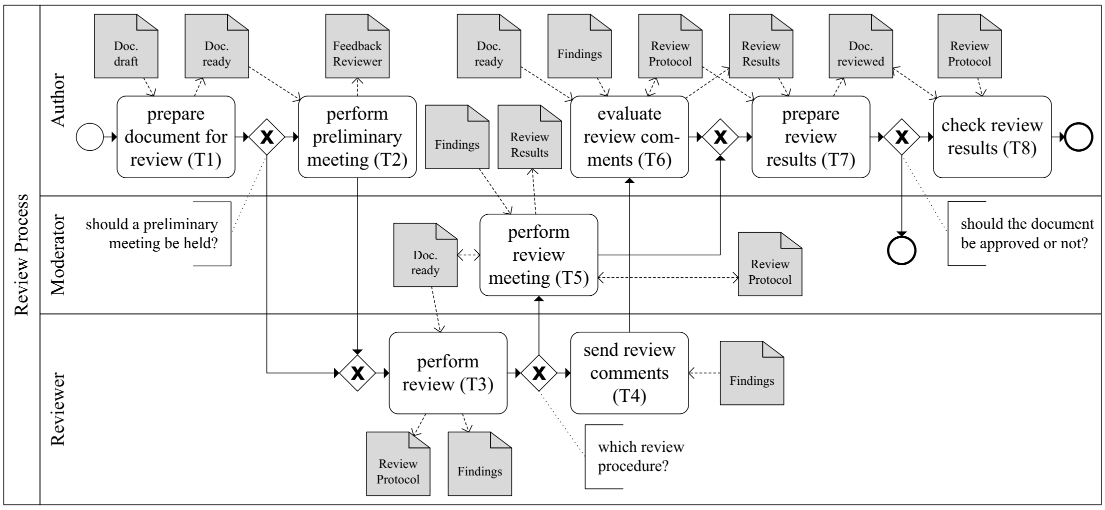

Figure 9.2: Use case: review of product requirements.

9.2.2 Implementation

Based on the discussed use case, we first implemented the corresponding SIN. Altogether, it comprised a process schema, three process instances, and about 300 documents (i.e., process information) such as review protocols, guidelines, and review results. For creating the SIN, we used the semantic middleware iQser GIN Server as well as several Java open-source plugins we had developed in this context. The implemented SIN includes 348 objects (45 process objects, 303 information objects) and 65,991 relationships (77 process object relationships, 65,319 information object relationships, and 595 cross-object relationships). While Figure 9.3 shows the entire SIN of the use case, Figure 9.4 only depicts objects (i.e., information and process objects) directly related to Task T3. For privacy reasons, the document names have been blurred in the following screenshots.

We then implemented the algorithms in a prototype called iGraph (cf. Section 8.2), a web-based Java application. iGraph uses the web framework Play 2.1.1, the web engine

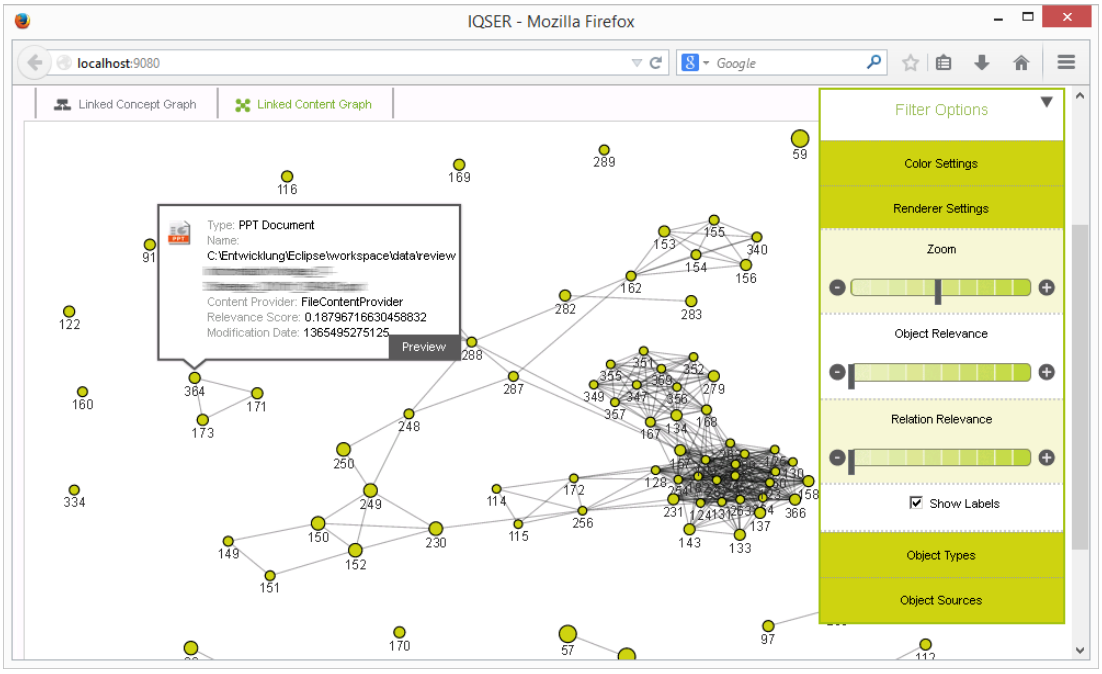

Figure 9.3: Entire SIN of the use case.

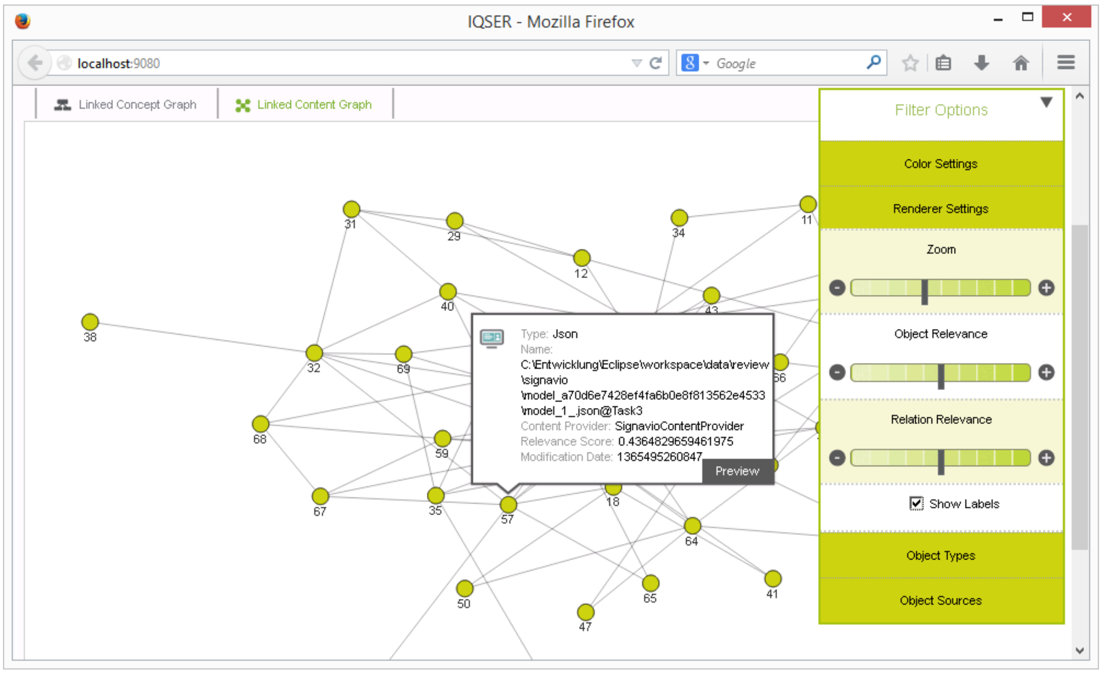

Figure 9.4: Task T3 and related information and process objects.

Bootstrap 2.3.1, the JavaScript library jQuery 1.8.3, the database MySQL 5, the text search engine library Lucene 2.4, the JavaScript library D3 3.1.1, HTML5, and CSS3.

9.2.3 Empirical Validation

In order to validate the SIN LP and SIN RP algorithms, we conducted a survey in the automotive domain. In this survey, automotive engineers evaluated previously calculated results of the two algorithms. With this survey, we want to show that the algorithmic results can indeed replace the costly and time-intensive human determination of relevant process information. More specifically, the goal is to prove the accuracy of the SIN LP and SIN RP algorithms.

RQ1.1 (Investigating the SIN LP algorithm). To investigate RQ1.1, we used iGraph to calculate two link popularity result lists. As input values, we set $init = 0.45$, $i = 12$, and $d = 0.5$. Moreover, we double weighted "is similar to" relationships since these relationships are usually more important than others in a SIN (cf. Section 5.3.1).

Overall, we received two result lists: The first list constituted the top eight documents according to the SIN LP algorithm for Task T1 ("prepare document for review"). In turn, the second list constituted the top eight documents according to the SIN LP algorithm for Task T3 ("perform review"). Table 9.2 shows the documents (i.e., process information) the SIN LP algorithm returns for Tasks T1 and T3. We then asked survey participants to evaluate the relevance of the calculated documents for both process tasks.

#	ID	Type	SIN LP	Marked	Ratio	Relevant?
T1	1231	Review Template	0.443	12	60.0 %	✓
	1210	Process Overview	0.442	20	100.0 %	✓
	439	Review Template	0.441	4	20.0 %	
	432	Specific Review	0.439	17	85.0 %	✓
	811	Guideline	0.435	4	20.0 %	
	439	Protocol	0.434	2	10.0 %	
	578	Checklist	0.434	19	95.0 %	✓
	777	Guideline	0.432	19	95.0 %	✓
T3	1210	Process Overview	0.443	17	85.0 %	✓
	879	Protocol	0.442	19	95.0 %	✓
	431	Specific Review	0.441	10	50.0 %	
	432	Specific Review	0.439	9	45.0 %	
	741	Review Template	0.435	7	35.0 %	
	439	Review Template	0.434	6	30.0 %	
	578	Checklist	0.434	18	90.0 %	✓
	729	Review Template	0.432	19	95.0 %	✓

Table 9.2: SIN LP algorithm validation results.

As can be seen from Table 9.2, the survey participants confirmed the relevance for the majority of the 16 documents identified by the SIN LP algorithm. Note that we consider a document as being relevant if more than half of the survey participants confirm relevance. Moreover, the results show that the SIN LP algorithm is indeed well working, especially since its overall accuracy can be further improved, for example, by combining it with other algorithms (e.g., the SIN RP algorithm).

RQ1.2 (Investigating the SIN RP algorithm). To investigate RQ1.2, we first calculated a ranking of review templates with the SIN RP algorithm based on real-world ratings we obtained from the automotive OEM supporting the survey. Additionally, we calculated three rate-based rankings on Formula 5.8 (ranking based on the total number of ratings), Formula 5.9 (ranking based on the average rating), and a random ranking. For example, Figure 9.5 shows a ranking of PDF documents (mainly guidelines) according to the SIN RP algorithm.

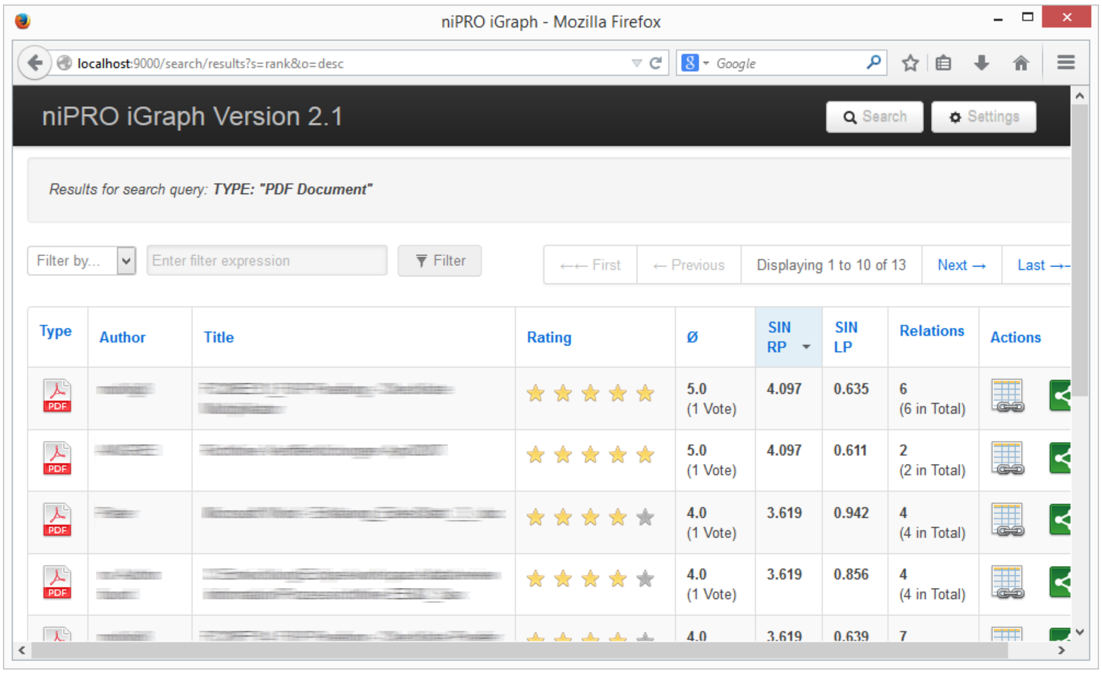

Figure 9.5: Ranking of PDF documents according to the SIN RP algorithm.

We then asked survey participants to evaluate both the plausibility and the usefulness of the four rankings. Figure 9.6A shows that 16 out of 20 participants consider the ranking created with the SIN RP algorithm as the most plausible one. The ranking based on the total number of ratings is considered as the second most plausible one with three votes. The ranking based on the average rating only received one vote by the participants. The random ranking received no votes. Moreover, as aforementioned, we asked the participants to evaluate the usefulness of the rankings based on a 5-Likert scale [284] ranging from “not at all useful” to “very useful”. Figure 9.6B shows that 87.5% of the participants stated that the ranking created with the SIN RP algorithm is "useful" or "very useful". Again, survey results confirm that the SIN RP algorithm is indeed performing well and can support participants during daily work.

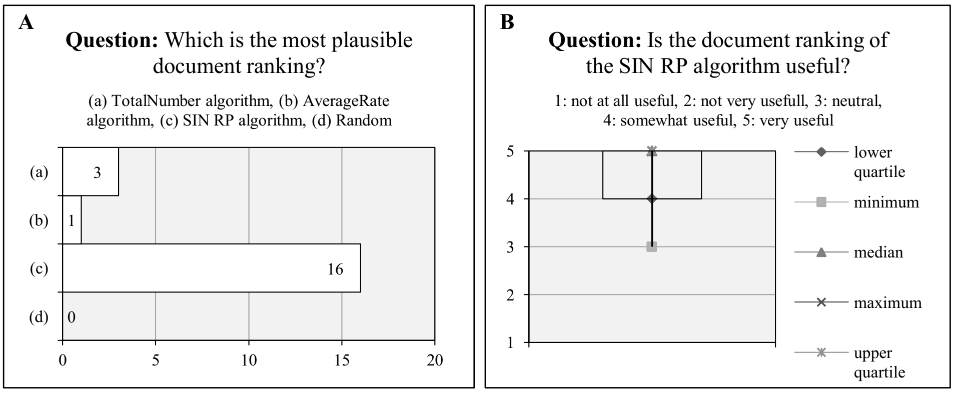

Figure 9.6: SIN RP algorithm validation results.

9.2.4 Conclusion

The considered automotive scenario confirms that most of the documents returned by the SIN LP algorithm are indeed relevant (RQ1.1). Moreover, the empirical research shows that the link popularity constitutes a good indicator for identifying relevant process information, especially since results of the SIN LP algorithm can be further refined for specific tasks by applying the SIN LP algorithm to only specific parts of the SIN (e.g., to a particular process task, corresponding task instances, or related information objects).

The results of the SIN RP algorithm are considered as being very useful by the participants as well (RQ1.2). In fact, most participants state that the ranking of documents as suggested by the SIN RP algorithm is both plausible and useful. Additionally, the algorithm avoids the problematic situation that process information with only few good user ratings is directly ranked on the first position. Finally, note that the results of the algorithm can be further improved, for example, by taking the expertise of users into account; i.e., ratings of experienced users might be weighted higher.

In summary, the popularity values of the algorithms (cf. Sections 5.5.1 and 5.5.2) clearly help to determine the relevance of process information. However, as it is difficult to determine the overall relevance of process information based on a single algorithm, we will combine the algorithms when extending the POIL framework at a later stage.

9.3 Case Study: Automotive Domain

In order to prove that the SIN maintenance algorithms presented in Section 5.4.3 are able to maintain SINs, we need to validate them. Specifically, we first implemented the algorithms and evaluated their performance. Thereby, we took both *depth* and *breadth* into account (cf. Section 9.3.1). This means that we measured the time needed to add, update, and delete SIN objects and relationships. In a second step, we then evaluated the applicability of the algorithms based on a real-world automotive use case (cf. Section 9.3.2). A part of this empirical validation were interviews with employees about the usefulness of SIN maintenance. Results show that the provided algorithms can indeed replace the costly and time-intensive human maintenance of SINs (cf. Section 9.3.3).

Overall, the validation has been guided by three research questions (cf. Table 9.3):

#	Research Questions
RQ2.1	Is SIN maintenance feasible taking depth and breadth into account?
RQ2.2	How do depth and breadth affect the runtime of the SIN maintenance algorithms?
RQ2.3	How useful is the automatic maintenance of objects and relationships in a SIN?

Table 9.3: Research questions underlying the case study.

9.3.1 Implementation

We implemented the SIN maintenance algorithms as presented in Section 5.4.3 in a prototype. The prototype was realized as a web-based Java application based on the semantic middleware iQser GIN Server 2.0, the web framework Play 2.1.1, the web engine Bootstrap 2.3.1, the JavaScript library jQuery 1.8.3, the database MySQL 5, the text search engine library Lucene 2.4, the JavaScript library D3 3.1.1, HTML5, and CSS3.

As introduced in Section 5.3.1, when maintaining a SIN, not only the object-relationship level must be considered, but the properties of objects and relationships as well. For example, if the title of an information object has changed, it is not necessary to overwrite the entire object. Instead, only changing parts need to be updated.

Based on the property classification from Section 5.4.2, we then created several SINs for technical analysis and performance evaluation. These SINs allow us to answer RQ2.1 and RQ2.2. Note that the objects and relationships share the same properties (cf. Figure 9.7). This becomes necessary to be able to compare the outcome of the validation.

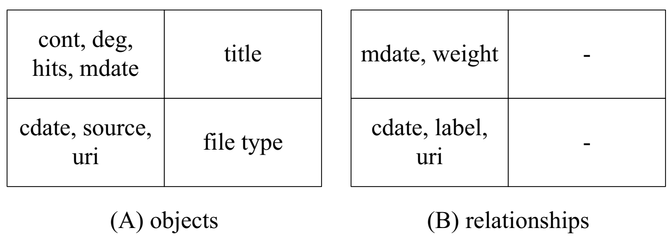

Figure 9.7: Properties of objects and relationships in the SINs.

Specifically, when considering objects, we chose “cdate” (creation date of the object), “source” (data source of the object), and “uri” (unique identifier of the object) as *mandatory* and *static* properties. Moreover, *optional* properties were “file type” and “title”.

While property “file type” does not change over time (it is a *static* property), the “title” may change (it is a *dynamic* property). Then, *mandatory* and *dynamic* properties were “cont” (content of the object), “deg” (number of relationships of the object), “hits” (number of hits of the object), and “mdate” (modification date of the object).

Analogous to object properties, “cdate” and “uri” were *mandatory* for relationships. However, relationships had additional *mandatory* properties such as the “label” (reason of the relationship) or “weight” (relevance of the relationship). Property “weight” can vary; i.e., changing “cont” may affect the “weight” of “is similar to” relationships.

Overall, we created six SINs containing 5, 50 and 500 objects, either with smaller files (1 KB) or larger files (100 KB) (cf. Figure 9.8). We investigated the performance of add, update and delete operations for both the *pull* and *push algorithm* (cf. Section 5.4.3). This results in a total of 36 (= 36∗3∗2) cases (six SINs, three operations, two algorithms). An exemplary case may be “adding an object with a size of 1 KB to a SIN containing 50 objects using the pull algorithm”. To ensure comparability, all objects within a SIN were equal to each other; i.e., all objects had same properties “author” and “content”.

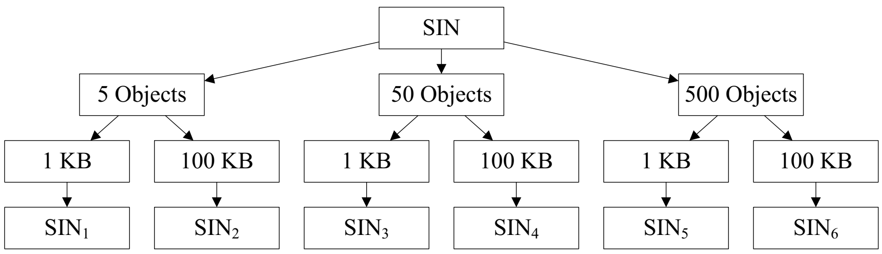

Figure 9.8: Correlation between the SINs.

Note that we simulated worst-case scenarios for the performance tests. This means that each SIN object is related with every other object in the SIN. This results in 40 (considering 5 objects), 4,900 (considering 50 objects), and 499,000 relationships (considering 500 objects). Note that only “is similar to” and “is author of” relationships were discovered since the text files we used do not have the property “title” and therefore no “has same title as” relationships were discovered during the technical analysis.

For each of the 36 cases, the performance results shown in the diagrams represent averages over three warm runs (cf. Figures 9.9-9.11). Warm runs were chosen to ensure comparability of the measured values since the iQser GIN Server [182] performs several initial background tasks after start-up. The performance tests were performed on a laptop with an Intel quad-core CPU Intel Core i7 2670Q with 3.1 GHz, 16 GB RAM, 512 GB solid-state drive (SATA 6 Gbit/s), and a Windows 7 64-bit operating system.

Add-Operations. Figure 9.9 shows that objects can be added to a SIN in *linear* time. More specifically, an object can be added in 169 ms on average. New relationships are discovered in 7446 ms on average. Therefore, *breadth* has a bigger influence on the runtime of the SIN maintenance algorithms than *depth*. Further note that discovering relationships between the added and existing objects is *polynomial* in the number of objects. The overall performance of the algorithms depends on the properties of objects and relationships as well. For example, the larger an object is (in terms of content), the longer will it take to add it to the SIN. The discovery of relationships will also take longer then since the underlying analysis needs more time to analyze the content.

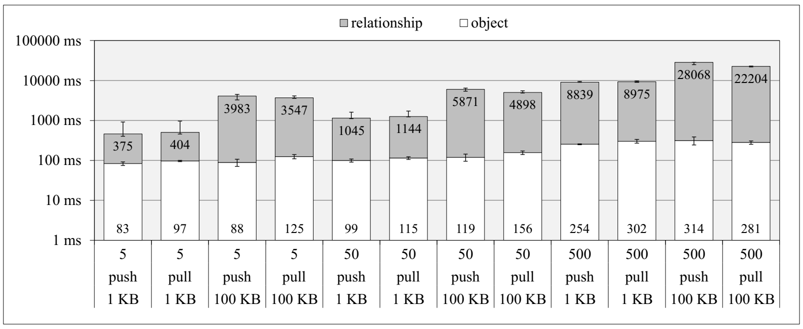

Figure 9.9: Effect of depth and breadth (add operations).

Update-Operations. Figure 9.10 shows that update operations perform similarly compared to add operations. An object can be updated in the SIN in 176 ms on average. In turn, corresponding relationships are discovered in 7982 ms on average. Therefore, again, *breadth* has a bigger influence on the runtime of the algorithms. As opposed to add operations, update operations need not update mandatory and static properties (e.g., "uri"). Instead, comparisons between properties need to be performed (e.g., to check if a property has changed). However, we could not detect any significant differences concerning the cost of add and update operations in identical situations (i.e., cases).

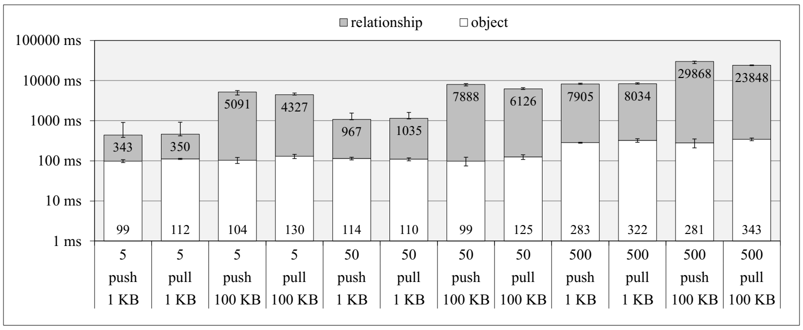

Figure 9.10: Effect of depth and breadth (update operations).

Delete-Operations. As opposed to add and update operations, delete operations perform differently (cf. Figure 9.11). While objects can be deleted in *linear* time, the time for deleting relationships significantly depends on the size of involved objects. On average, an object can be deleted in the SIN in 143 ms. In turn, corresponding relationships are deleted in 1451 ms on average. The cost for deleting an object with a size of 1 KB out of 5 objects (i.e., 73 ms) was higher than deleting an object with a size of 100 KB out of 5 objects (i.e., 57 ms) (cf. Figure 9.11). This might be explained with the specific implementation characteristics of the iQser GIN Server or measurements with Java which can be less accurate compared to native programming languages (e.g., the Java garbage collector cannot be disabled during the performance tests).

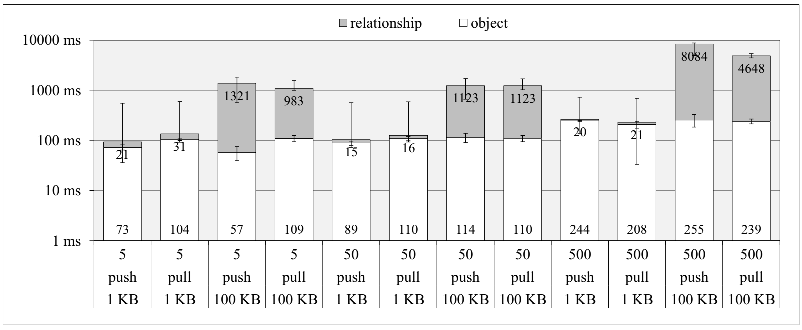

Figure 9.11: Effect of depth and breadth (delete operations).

Despite the fact, that the used technology might cause inaccuracies, we were still able to verify that maintenance cost with the SIN maintenance algorithms highly depends on both *depth* and *breadth* (RQ2.2). However, external components (e.g., for a syntactic or semantic analysis) can significantly influence the performance of SIN maintenance operations. The consideration of the property classification (cf. Section 5.4.2), in turn, can have positive effects on maintenance performance if only necessary operations (e.g., only updating properties which are not *mandatory* and *static*) are executed.

Altogether, the push and pull algorithms ensure a continuously synchronized SIN. The technical validation has shown that automatic maintenance of SINs is applicable as well as feasible (RQ2.1). We showed that the runtime of the SIN maintenance algorithms is mainly influenced by depth and breadth (RQ2.2). In the following, we evaluate the necessity and practicability of automatic maintenance of SINs in the automotive domain.

9.3.2 Empirical Validation

We performed an empirical case study to evaluate RQ2.3 ("How useful is the automatic maintenance of objects and relationships in a SIN?") with knowledge workers and decision makers from several innovation departments in the automotive domain. Thus, all users are involved in knowledge-intensive business processes and tasks. The participants were selected based on their expert knowledge regarding the considered use case.

The use case deals with the identification of upcoming technologies. More specifically, process participants investigate technologies according to strengths, weaknesses, opportunities, and threats [285]. The process requires a lot of process information, expertise, and decision making. Three basic roles are involved: (1) the *decision maker* is responsible for the prioritization of technologies as well as for the assignment of a *technology responsible* with the investigation of a technology, (2) the *technology responsible* identifies experts for a particular technology and supervises their work, and (3) the *expert* works in pre-development projects and continuously develops and improves various technologies.

First, the participants had to answer general questions about their current work context and on how they handle process information when performing business processes. Second, they had to perform tasks with our prototype and answer questions regarding automatic SIN maintenance. The SIN we used for the case study contained 333 documents from the participants' field of interest (e.g., innovation reports, technology fact sheets). Therefore, users were familiar with information represented in the SIN and able to evaluate the information quality. Third, in order to generalize results and to gain further insights, we asked the users to evaluate some concluding questions.

The case study results show that process information is mostly handled and accessed in an electronic form. Process information is mostly unstructured and often distributed across different data sources, which makes it difficult to find. Figures 9.12A and 9.12B show that users cannot easily identify needed process information. Besides this, process information needed during daily work changes frequently. Therefore, SINs must ensure a maintenance even though process information frequently changes.

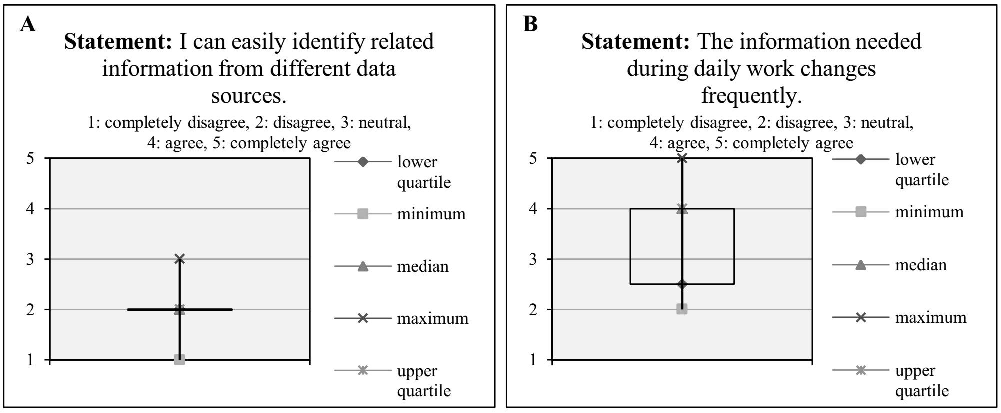

Figure 9.12: Information characteristics.

In order to evaluate the usefulness of SINs, we introduced the proof-of-concept prototype and asked the participants to perform selected tasks (e.g., "check if a property value was updated" or "check the correctness of relationships"). After completing the tasks, we asked the participants whether the SIN can be easily maintained by the algorithms, which the majority of the participants confirmed (cf. Figure 9.13A). Moreover, we asked for the effort needed for maintaining SINs. The majority confirmed that the effort for maintaining SINs is low (cf. Figure 9.13B). However, these results are not surprising since the algorithms work completely automatically and were already tested as well.

Additionally, we asked the users whether they prefer the push or pull algorithm. The majority prefers the push algorithm since the effects of their tasks are reflected immediately in the SIN. By contrast, the pull algorithm always has a short delay since it is triggered at a certain point in time. As an optimization, the partial-pull algorithm addresses this issue. The partial-pull algorithm received positive feedback from the par-

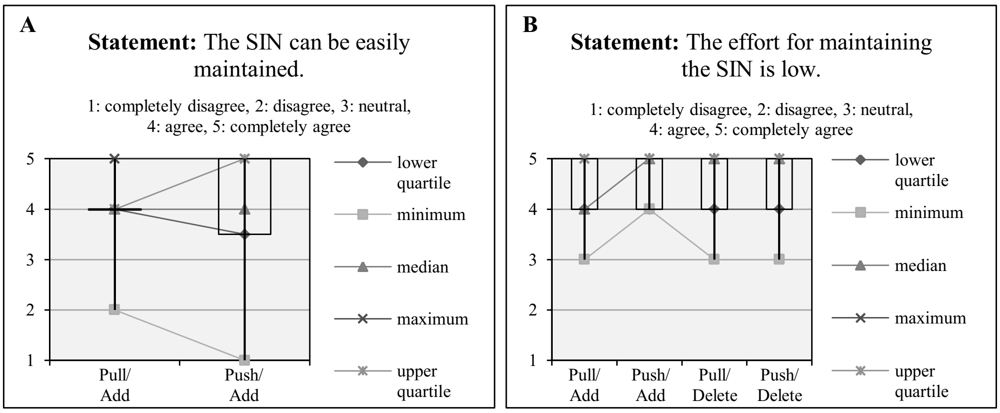

Figure 9.13: SIN maintenance: simplicity and effort.

ticipants. Note that the partial-pull algorithm only considers objects currently existing in a SIN. Concluding, we asked the users about their impression concerning the benefits for their daily work in consideration of automated SIN maintenance. Participants stated that integrating distributed process information is easy when using the prototype (cf. Figure 9.13A) and that it can be also done with reasonable effort (cf. Figure 9.13B). Additionally, they confirmed the benefit of a SIN for their daily work (cf. Figure 9.14).

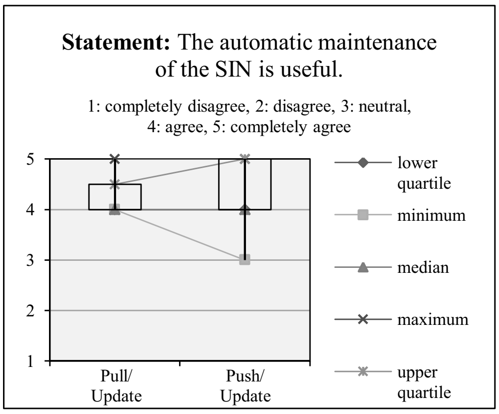

Figure 9.14: Automatic maintenance of the SIN.

Asking about use cases for a maintained SIN, both knowledge workers and decision makers recognized further potentials to support their daily work. For example, participants stated that expert search and decision support scenarios are of great relevance.

More than 90% of the participants stated that a SIN provides an extended overview on process information and business processes. Therefore, a maintained SIN is desired by process participants. In particular, users benefit from a homogeneous view on process information and business processes integrated from large, distributed and heterogeneous data sources. A participant of the study summarized: "maintenance and corresponding updates should be automatically done; no user interaction for maintenance should be required; process participants should focus on working with the SIN."

9.3.3 Conclusion

We have shown that SIN maintenance is feasible with acceptable costs (RQ2.1). The SIN maintenance algorithms performed satisfactorily when adding, updating and deleting objects and relationships in a SIN. The costs for discovering relationships, however, vary significantly; i.e., depending on the algorithms used for this task (RQ2.2). In the case study, most of the automotive participants stated that a maintained SIN constitutes a prerequisite for highly needed use cases, e.g., for finding experts within and across enterprises or for handling inter-linked or similar objects. Therefore, in practice, knowledge workers and decision makers can benefit from maintained SINs.

The validation further shows that it is indeed helpful to provide up-to-date and homogeneous objects. Moreover, the validation confirms that there is a demand for a central point of access to process information when performing knowledge-intensive processes. We can conclude that a maintained SIN is a prerequisite for such processes (RQ2.3).

9.4 Experiment: Agricultural Domain

In order to validate the POIL framework, we conducted a controlled experiment at a large enterprise from the agricultural domain. Overall, twelve employees participated. They mainly stemmed from the sales department, but also from the department responsible for project management. These departments were selected because of the knowledge-intensive business processes they perform. The main goal of the experiment was to investigate the benefits of a process information portal implementing POIL compared to a conventional enterprise portal implementing hard-wired IL [3].

Specifically, we implemented the architectural layers of the POIL framework except for the context layer in a process information portal called iProcess (cf. Section 9.4.1). Then, we compared the experimental results of POIL with the ones of a hard-wired IL solution as used by the involved enterprise (cf. Section 9.4.2). The results show that the POIL framework can indeed close the gap between process information and business processes. The framework can decrease the time needed for handling and managing process information during business process execution (cf. Section 9.4.3).

Overall, the experiment has been guided by two research questions (cf. Table 9.4):

#	Research Questions
RQ3.1	Is the POIL framework suitable to close the gap between process information and business processes?
RQ3.2	Is the POIL framework more suitable for providing relevant process information than hard-wired IL?

Table 9.4: Research questions underlying the experiment.

In the experiment, participants had to perform tasks in the context of a newsletter process. For example, participants had to find the last created newsletter, current newsletter templates, or current newsletter guidelines (cf. Section 9.4.2). The specific use case underlying the experiment was the creation and review of newsletters. Process participants create newsletters and send them to customers in order to inform them about upcoming events and special offers. Two basic roles are involved: (1) the *case worker* gathers required information and creates the newsletter and (2) the *decision maker* initiates the creation of the newsletter and finally reviews the newsletter.

When designing the experiment, the following criteria were considered: (1) the design of the experiment should allow gathering as much empirical data as possible with respect to the goals of the experiment, (2) the gathered data should be unambiguous, and (3) the experiment had to be applicable and feasible for a given setting [286, 287].

Considering these criteria, we designed the experiment as a *controlled single factor experiment* [288]. This means that participants are randomly divided into two groups with six subjects each (cf. Figure 9.15). There is one *experimental group* (i.e., group A) and one *control group* (i.e., group B). The experimental group worked with the iProcess prototype, whereas the control group worked with an existing enterprise portal provided by the experiment partner. This portal is implemented as hard-wired IL solution.

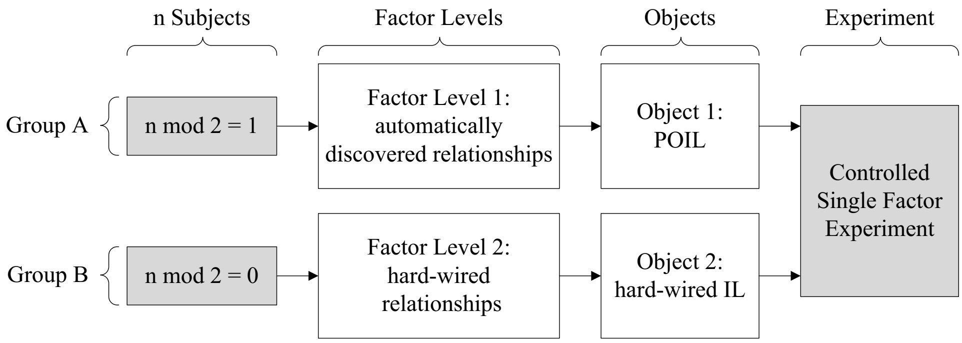

Figure 9.15: Design of the experiment.

The subjects, objects and response variables of the experiment are as follows:

- **Subjects:** Subjects are twelve[1] employees working at a large enterprise in the agricultural domain. All subjects are familiar with the considered use case of the experiment; i.e., creation and review of newsletters. The subjects are divided into two groups. Each group comprises six subjects. Subjects are randomly assigned to the groups according to the above mentioned procedure.

- **Objects:** The objects to be evaluated are (1) a prototype implementing POIL and (2) a conventional enterprise portal implementing hard-wired IL.

[1]We only chose subjects from the agricultural domain who were familiar with the newsletter process; i.e., participants had to perform the newsletter process at least six times in the last twelve months. This fact explains the small number of subjects since such employees are hard to find. Note that similar experience levels are necessary to ensure the internal validity of the experiment.

- **Factor and Factor Levels:** The factor POIL is applied to the prototype. The factor levels are (1) automatically discovered relationships between process information and business processes and (2) hard-wired relationships.
- **Response Variables:** Response variables are considered to describe the participants' impressions of the objects used: *intuitiveness* and *speed*. We also consider the *relevance* of process information. Finally, execution times of particular tasks have been measured during the experiment.
- **Instrumentation:** To collect data, we interview participants and use a paper-based questionnaire. For logging execution times, we use the log file of the iQser GIN Server [182]. This log file provides detailed information about performed tasks, executed search queries, and accessed process information. For the conventional enterprise portal, we stop execution times manually.
- **Data Collection Procedure:** The same questionnaire is used for the entire data collection. It includes questions and, if necessary, pre-defined answers (based on Likert scales), and tasks to be executed by the subjects.
- **Data Analysis Procedure:** For data analysis well-established statistical methods, such as factor analysis and analysis of variance, are applied [289].

9.4.1 Implementation

We implemented the POIL framework in a prototype called iProcess[2]. We realized the latter as a web-based Java application based on the semantic middleware iQser GIN Server 2.0 [171], the web framework Play 2.1.1, the web engine Bootstrap 2.3.1, the JavaScript library jQuery 1.8.3, the database MySQL 5, the text search engine library Lucene 2.4, the JavaScript library D3 3.1.1, HTML5, and CSS3. Figure 9.16 shows the start screen of iProcess for a newsletter process and four running process instances.

Features. iProcess enables the comprehensive *integration* of process information and business processes from heterogeneous data sources. Both process schemas and instances are supported. iProcess allows for the syntactic, semantic and conceptual *analysis* of integrated information and process objects. Additionally, it enables the process-oriented *delivery* of needed process information and business processes to knowledge workers and decision makers when creating a particular newsletter.

Architecture. iProcess implements all architectural layers of the POIL framework except for the context layer; i.e., (1) the *data layer*, (2) the *semantic layer*, and (3) the *application layer*. However, at least some features of the context layer are implemented as well. For example, the users can select roles of the use case (e.g., case worker, decision maker) and are then provided with needed process information depending on their role.

[2] A screencast presenting iProcess is available at http://nipro.hs-weingarten.de/screencast.

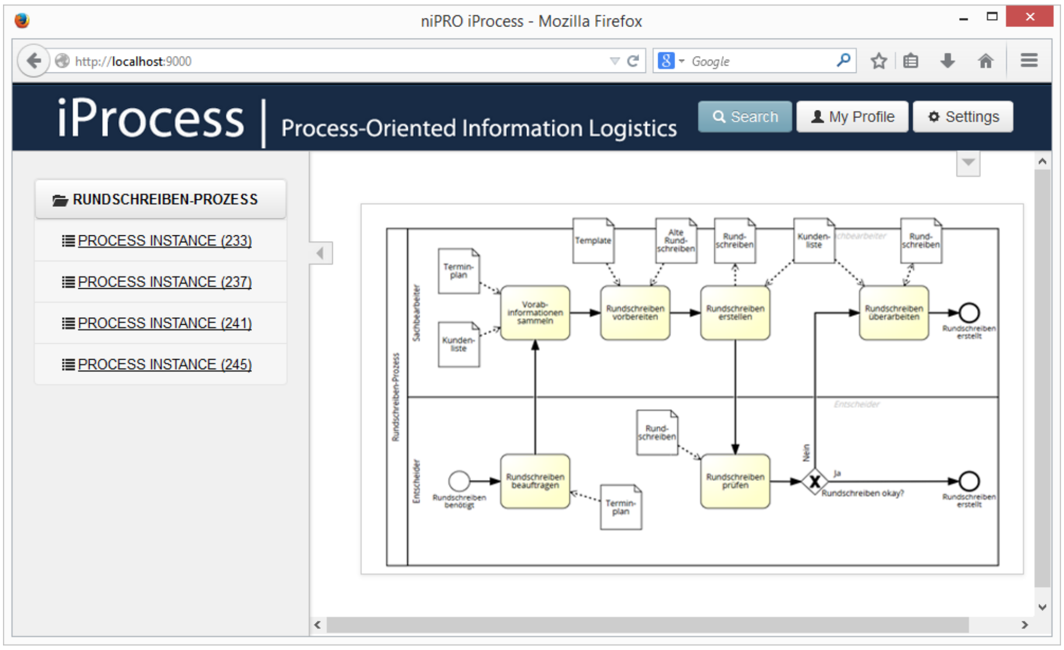

Figure 9.16: iProcess: start screen for the newsletter process.

The *data layer* concerns the set of data sources to be integrated. For each data source, a ContentProvider is implemented. Its main task is to transform proprietary process information (e.g., newsletter guidelines) or business processes (e.g., newsletter process) into uniform information and process objects. In turn, the *semantic layer* is responsible for the syntactic and semantic analysis of information and process objects. For this purpose, we use the semantic middleware iQser GIN Server and implement several AnalyzerTasks. The goal is to classify and group correlated objects (e.g., filled-out newsletter templates). Furthermore, the user behavior during the newsletter process is investigated, e.g., the frequency of using particular templates when preparing the newsletter. Finally, the *application layer* deals with the delivery of process information.

In the following, we show how iProcess can be applied to the use case. More precisely, we consider one process schema[3], four process instances[4], and about 1,500 documents[5] (i.e., process information) such as newsletters, newsletter templates, customer lists, event documents, presentations, and guidelines. For illustration purposes, we pick up a specific process task of the newsletter process; i.e., the preparation of the newsletter.

Let us assume that a case worker prepares a particular newsletter. For this purpose, iProcess provides him with currently needed information about the process task (cf. Figure 9.17). For example, the start and end times of the task are provided. Additionally,

[3]The process schema is modeled with Signavio Process Editor [249].
[4]The process instances are managed with the Activiti BPM Platform [250].
[5]These documents are real-world documents and stem from our industry partner.

the case worker is provided with the name and a detailed description of the task. This helps, for example, inexperienced process participants to perform their task successfully.

Besides information about the process task, related process information (e.g., newsletter guidelines, newsletter templates, and existing newsletters) is provided as well (cf. Figure 9.18). iProcess automatically identifies the top five process information items for a task based on the SIN LP algorithm. Additionally, if provided process information is not sufficient, the case worker may manually search or explore the SIN. When the user clicks on button “Process Search”, iProcess displays an advanced search form to query the SIN. In turn, when the user clicks on buttons “Graph” or “Table”, iProcess provides a graph- or table-based view. Figure 9.19 shows, for example, the graph-based view on the SIN. Figure 9.19A shows process information related to the process schema, whereas Figure 9.19B shows related process information related to a process instance.

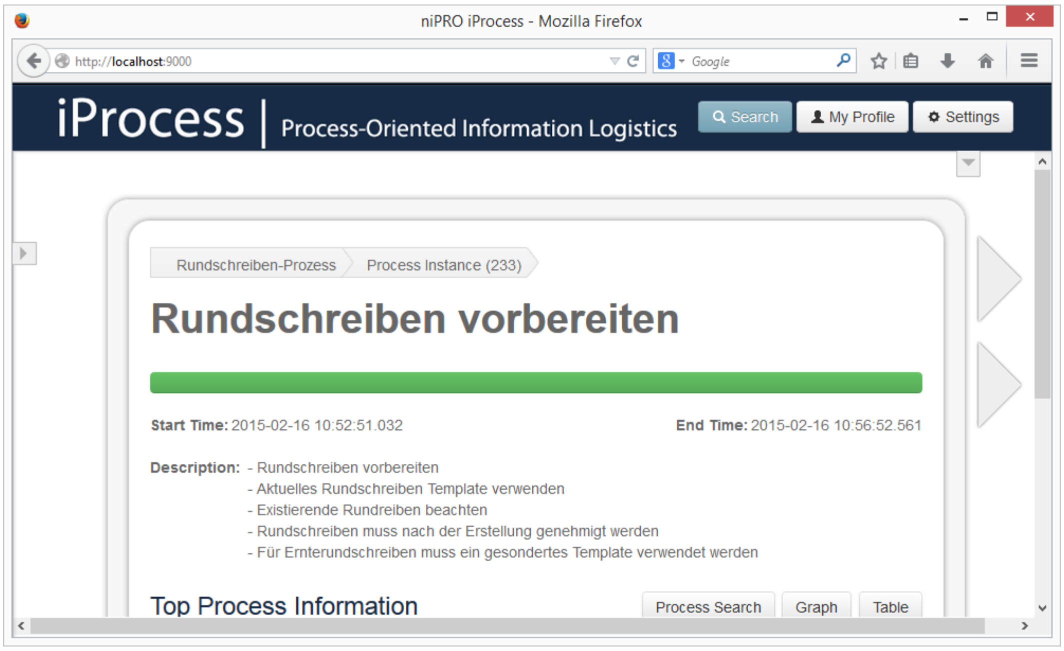

Figure 9.17: iProcess: preparation of a newsletter.

9.4.2 Empirical Validation

With the controlled single factor experiment, we are able to evaluate RQ3.1 (“Is the POIL framework suitable to close the gap between process information and business processes?”) and RQ3.2 (“Is the POIL framework more suitable for providing relevant process information than hard-wired IL?”). As aforementioned, twelve employees were involved, ten knowledge workers and two decision makers. All employees were familiar with the newsletter process. All employees were selected based on their knowledge regarding the considered use case as presented in Section 9.4.

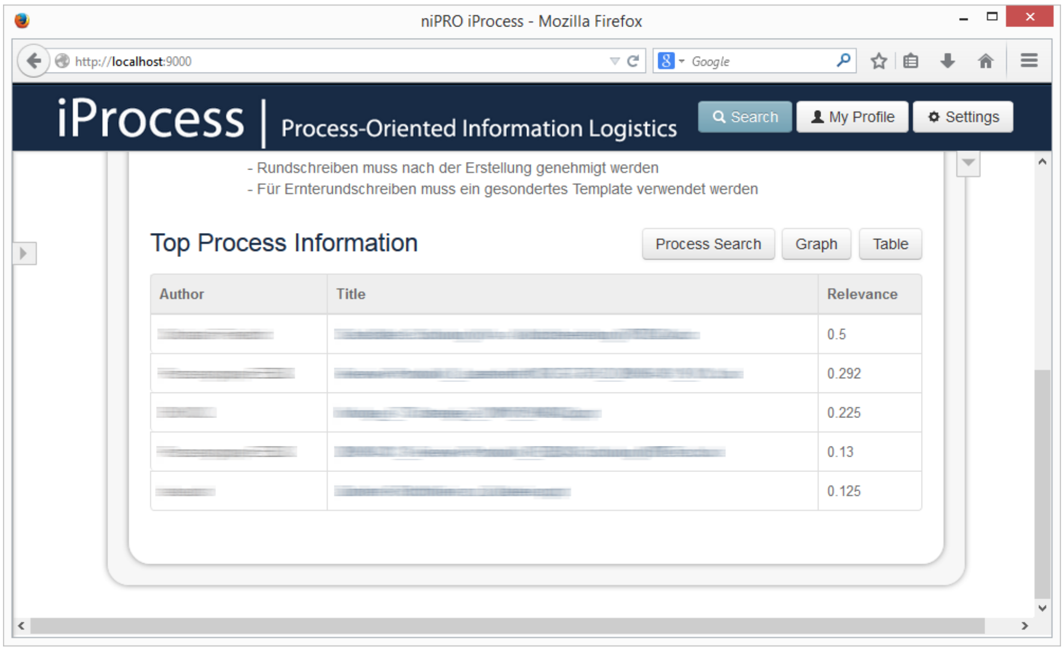

Figure 9.18: iProcess: related process information for preparing a newsletter.

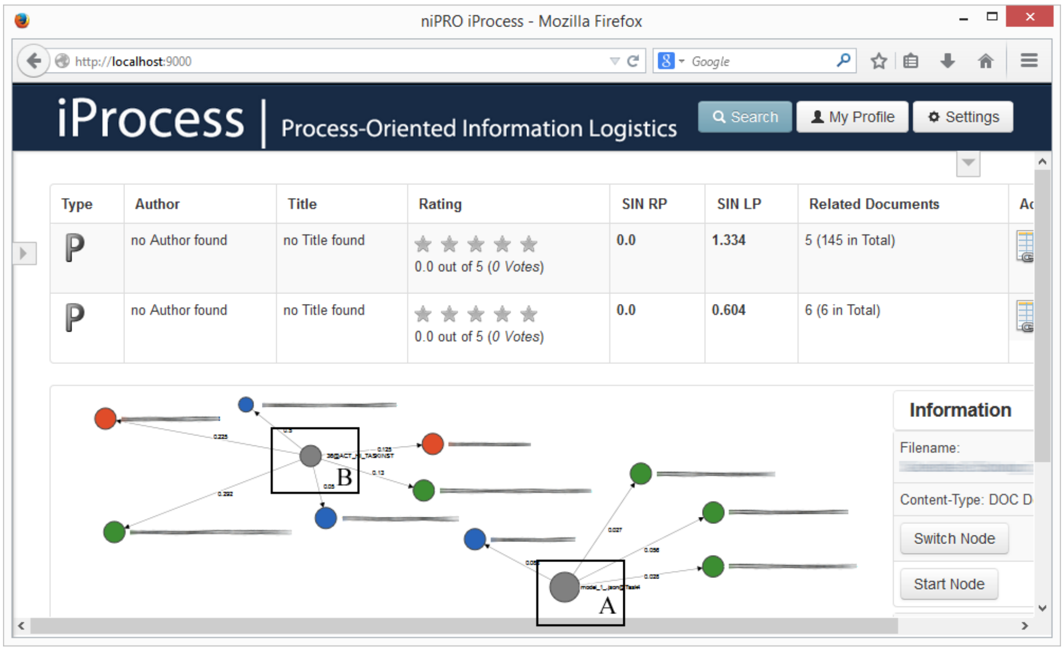

Figure 9.19: iProcess: closed gap between process information and business processes.

To investigate RQ3.1 and RQ3.2 (cf. Table 9.4), we define three hypotheses (H1-H3) clustering the experiment's response variables: (H1) *intuitiveness of the approach*, (H2) *relevance of process information*, and (H3) *speed of the approach*.

Intuitiveness of the Approach (H1). We investigate whether or not the POIL framework is more intuitive during process execution compared to hard-wired IL.

- **0-Hypothesis $H_{0,1}$:** There is no significant difference in how intuitive the POIL framework is during process execution compared to hard-wired IL.
- **Altern.-Hypothesis $H_{1,1}$:** There is a significant difference in how intuitive the POIL framework is during process execution compared to hard-wired IL.

Relevance of Process Information (H2). We investigate whether or not the POIL framework better identifies relevant process information compared to hard-wired IL.

- **0-Hypothesis $H_{0,2}$:** There is no significant difference in how good the POIL framework identifies relevant process information compared to hard-wired IL.
- **Altern.-Hypothesis $H_{1,2}$:** There is a significant difference in how good the POIL framework identifies relevant process information compared to hard-wired IL.

Speed of the Approach (H3). We investigate whether or not a task can be performed faster using the POIL framework compared to hard-wired IL.

- **0-Hypothesis $H_{0,3}$:** There is no significant difference in how fast a task can be performed with the POIL framework compared to hard-wired IL.
- **Altern.-Hypothesis $H_{1,3}$:** There is a significant difference in how fast a task can be performed with the POIL framework compared to hard-wired IL.

In order to confirm or reject the hypotheses, the following may applied (cf. Figure 9.20): (1) we introduced the procedure of the experiment to the participants, (2) we asked questions about demographic issues, business process management, and handling of process information, (3) we introduced iProcess and the enterprise portal respectively, (4) we introduced the experiment's tasks the users had to perform, and (5) we asked concluding questions.

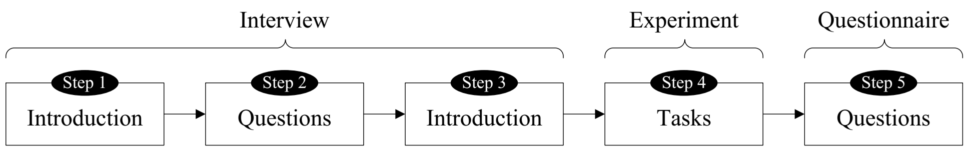

Figure 9.20: Steps of the experiment.

When performing the tasks, all subjects gave direct feedback (think-aloud protocol[6]) that was documented by the research team. For particular tasks (e.g., "find the last created newsletter"), execution times were additionally captured. After finishing the tasks, all subjects had to fill out a questionnaire.

Specifically, to answer the hypotheses, we confronted the experiment participants with ten statements (cf. Table 9.5) describing the experiences made during the experiment. Each participant had to evaluate all statements based on a 5-Likert scale [284] ranging from "completely disagree" to "completely agree".

#	**Statements**	**H1**	**H2**	**H3**	$\mathbf{p_1}$
S1	The approach is interesting.	×			0.005
S2	The approach is easy to use.	×			0.153
S3	The approach is easy to learn.	×			0.000
S4	The approach is fun to use.	×			0.010
S5	Process information is up-to-date.		×		0.028
S6	Process information is complete.		×		0.001
S7	Process information is consistent.		×		0.001
S8	Process information is needed for a particular task.		×		0.023
S9	Business processes can be quickly performed.			×	0.120
S10	Needed process information is quickly available.			×	0.008

p_1 = p_1-value according to Lilliefors [291, 292].

Table 9.5: Statements of the experiment.

We first checked whether the questionnaire results are *normally distributed*. Based on the Lilliefors test[7] [291, 292], we calculated the p_1-values for each statement S1-S10 (cf. Table 9.5). A p_1-value smaller than 0.05 indicates a statistically significant deviation from the normal distribution. Statements S1, S3-S8, and S10 have a p_1-value smaller than 0.05, whereas statements S2 and S9 have a p_1-value greater than 0.05.

In a second step, for S1, S3-S8, and S10, we applied the Mann-Whitney u-test[8] [293] and, for S2 and S9, we applied the t-test[9]. These tests are necessary to determine the final p-values (cf. Table 9.6). Moreover, we calculated the d-value of each statement, which indicates a trend if the p-value is not statistically significant enough. Thereby, a p-value is significantly enough if it is smaller than 0.05 [268].

[6]A *think-aloud protocol* involves participants thinking aloud as they are performing a set of tasks [290].

[7]The *Lilliefors test* is a test based on the KolmogorovSmirnov test. It is used to test the frequency distribution of data in order to identify deviations from the normal distribution [291, 292].

[8]The *Mann-Whitney u-test* is a test of the 0-hypothesis that two populations are the same against an alternative-hypothesis [293].

[9]The *t-test* is a test in which the test statistic follows a Student's t distribution if the 0-hypothesis is supported. It can be used to determine if two sets of data are significantly different from each other.

Intuitiveness of the Approach (H1). To investigate the first hypothesis, we used the following statements: (S1) the approach is interesting, (S2) the approach is easy to use, (S3) the approach is easy to learn, and (S4) the approach is fun to use.

Results show that iProcess is more interesting than the conventional enterprise portal (group A: 4.00, group B: 3.00, p: 0.093, d: 1.310, cf. Figure 9.21A). The reason may be that iProcess is somewhat new for the employees. Moreover, the enterprise portal of the industry partner is not state-of-the-art. Subjects stated that both approaches are easy to use (group A: 3.00, group B: 3.17, cf. Figure 9.21B) and easy to learn (group A: 4.00, group B: 3.83, cf. Figure 9.21C). Results show no significant differences (p: 0.734, p: 0.937). Concerning the fun to use, the participants stated that iProcess is more fun to use than the conventional portal (group A: 4.17, group B: 3.33, p: 0.132, d: 1.062, cf. Figure 9.21D). The reason may be that participants have an overview about business processes and their tasks as well. Moreover, iProcess provides different views for process participants; i.e., a table- and graph-based view.

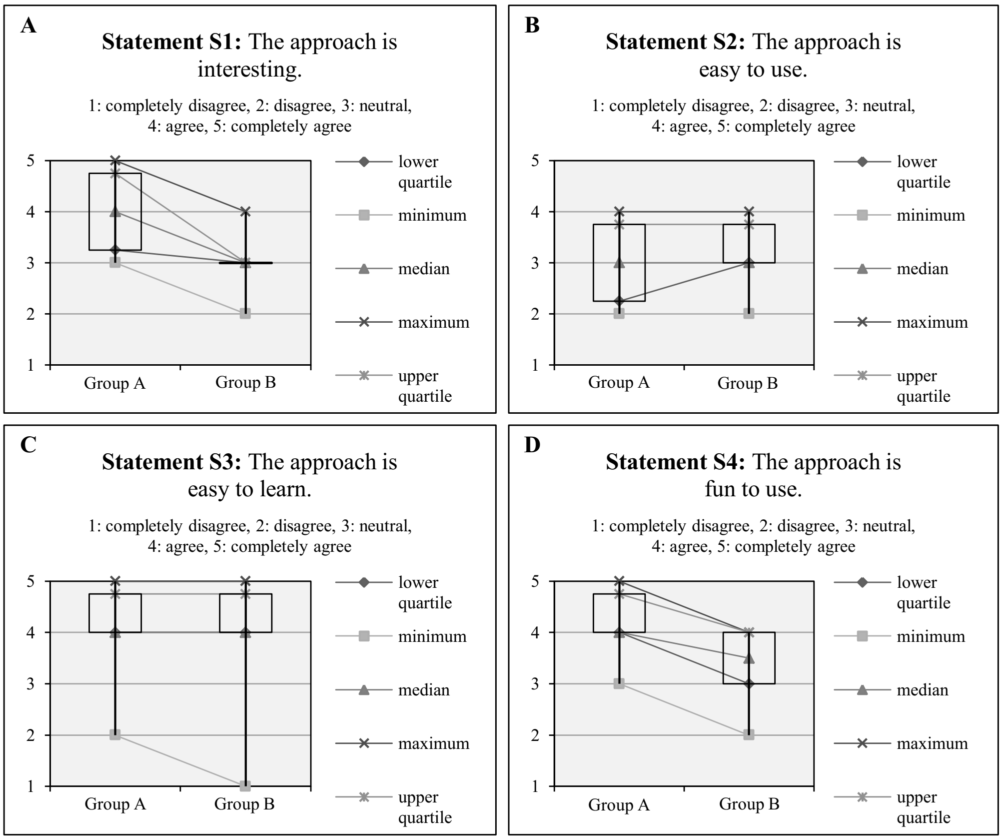

Figure 9.21: Results of hypothesis H1.

Based on these results, we can reject H1; i.e., there is a significant difference in how intuitive the POIL framework is during process execution compared to hard-wired IL. Especially, the users confirmed that POIL is more interesting and fun to use than IL.

Relevance of Process Information (H2). In order to investigate the second hypothesis, we used the following statements: (S5) process information is up-to-date, (S6) process information is complete, (S7) process information is consistent, and (S8) process information is needed for performing a particular task.

Results show that both POIL and hard-wired IL perform similarly. Subjects stated that both approaches deliver up-to-date process information (group A: 4.17, group B: 4.33, cf. Figure 9.22A), complete process information (group A: 4.33, group B: 4.17, cf. Figure 9.22B), and consistent process information (group A: 3.67, group B: 3.33, cf. Figure 9.22C). Results show no significant difference (p: 0.699, p: 0.818, p: 0.394). In turn, a significant difference exists when considering statement S8 (group A: 4.00, group B: 2.50, p: 0.015, d: 2.042, cf. Figure 9.22D). Subjects confirmed that POIL provides process information in a more user-adequate way; i.e., only few, but relevant process information is provided. Hard-wired IL, in turn, provides a large amount of process information without considering business processes and their tasks.

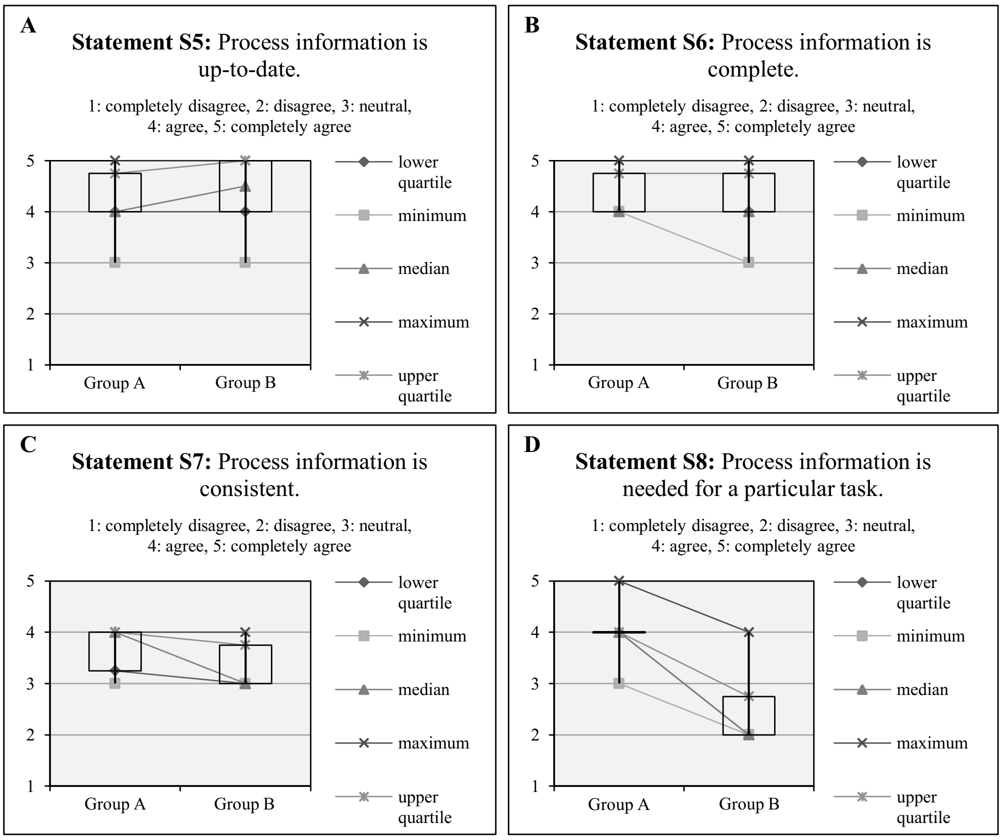

Figure 9.22: Results of hypothesis H2.

Based on this, we can reject H2; i.e., there is a significant difference in how good the POIL framework identifies relevant process information compared to hard-wired IL.

More specifically, POIL provides process information in a more user-adequate way. This means that an appropriate amount of process information is delivered.

Speed of the Approach (H3). To investigate the third hypothesis, we used the following statements: (S9) business processes can be quickly performed and (S10) needed process information is quickly available.

Results show that business processes can be quicker performed using the iProcess prototype (group A: 4.00, group B: 3.33, cf. Figure 9.22A) since the latter provides a homogeneous view on process information and business processes. Thus, employees have a good overview about the process tasks to be performed. However, results are not statistically significant (p: 0.341, d: 0.320). In turn, when considering the delivery of relevant process information, hard-wired IL performs better than POIL (group A: 3.33, group B: 3.83, cf. Figure 9.22B). A reason may be that the POIL framework automatically determines needed process information on demand and thus needs a specific time interval to identify relevant process information from the SIN. Therefore, performance issues constitute a crucial factor. However, results are not statistically significant as well (p: 0.394, d: 1,041).

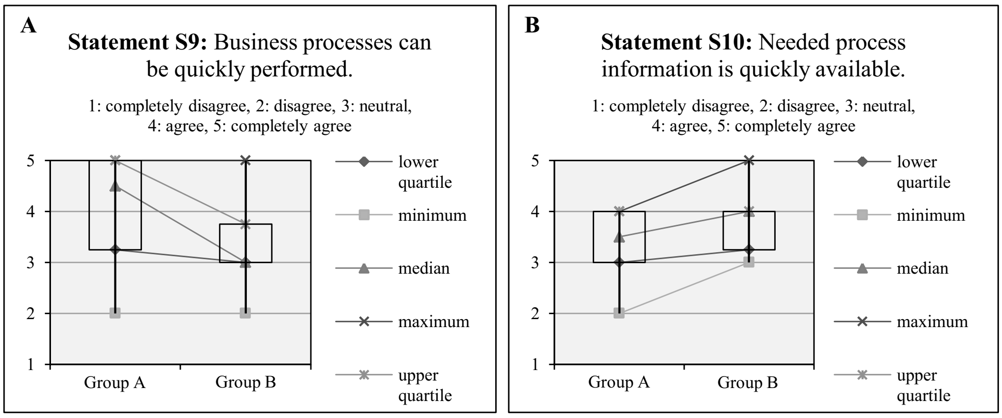

Figure 9.23: Results of hypothesis H3.

Based on the results, we cannot reject H3. In order to gain further insights, participants must perform different tasks using the iProcess prototype or the conventional enterprise portal; i.e., depending on the group the participants belong to.

However, we cannot identify significant results during the execution of the tasks. Figure 9.24 shows the measured execution times for the three investigated tasks: (T1) find the last created newsletter, (T2) find the current newsletter template, and (T3) find the current newsletter guideline. When considering the results, we can notice that users improve their execution times from task to task when using the iProcess prototype.

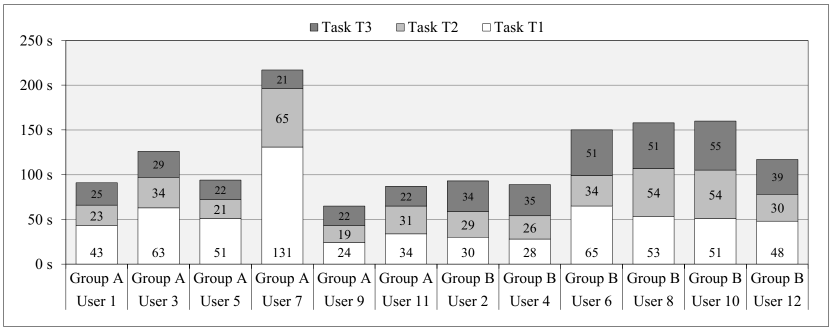

Figure 9.24: Measured execution times of the tasks.

Table 9.6 shows the raw results presented in the box plot diagrams from Figures 9.21-9.23. Moreover, we present the mean-, p- and d-value of each statement. These values are calculated based on aforementioned procedures (cf. Section 9.4).

#	Group A						Group B						All	
	lq	min	m_1	m_2	max	uq	lq	min	m_1	m_2	max	uq	p	d
S1	3.25	3.00	4.00	4.00	5.00	4.75	3.00	2.00	3.00	3.00	4.00	3.00	0.093	1.310
S2	2.25	2.00	3.00	3.00	4.00	3.75	3.00	2.00	3.00	3.17	4.00	3.75	0.734	0.202
S3	4.00	2.00	4.00	4.00	5.00	4.75	4.00	1.00	4.00	3.83	5.00	4.75	0.937	0.130
S4	4.00	3.00	4.00	4.17	5.00	4.75	3.00	2.00	3.50	3.33	4.00	4.00	0.132	1.062
S5	4.00	3.00	4.00	4.17	5.00	4.75	4.00	3.00	4.50	4.33	5.00	5.00	0.699	0.212
S6	4.00	4.00	4.00	4.33	5.00	4.75	4.00	3.00	4.00	4.17	5.00	4.75	0.818	0.263
S7	3.25	3.00	4.00	3.67	4.00	4.00	3.00	3.00	3.00	3.33	4.00	3.75	0.394	0.645
S8	4.00	3.00	4.00	4.00	5.00	4.00	2.00	2.00	2.00	2.50	4.00	2.75	0.015	2.042
S9	3.25	2.00	4.50	4.00	5.00	5.00	3.00	2.00	3.50	3.33	5.00	3.75	0.341	0.320
S10	3.00	2.00	3.50	3.33	4.00	4.00	3.25	3.00	4.00	3.83	5.00	4.00	0.394	1.041

lq = lower quartile, min = minimum, m_1 = median, m_2 = mean, max = maximum, uq = upper quartile, p = p-value, d = d-value

Table 9.6: Results of the statements S1-S10.

9.4.3 Conclusion

The results of the experiment confirm that the POIL framework is suitable to close the gap between process information and business processes (RQ3.1). This is demonstrated by the iProcess prototype, which provides related process information for particular process tasks. Participants of the experiment stated that a homogeneous view on process information and business processes can help to perform process tasks in a more effective and efficient way (cf. Figure 9.23A).

Results further shows that the POIL framework is more suitable for providing relevant process information than hard-wired IL (RQ3.2). Participants confirmed that provided process information is indeed needed during the execution of business processes (cf. Figure 9.22B). Based on these results, two of the three hypotheses can be rejected as shown in Table 9.7. The last hypothesis cannot be rejected since the results are not significant (cf. Table 9.6). However, we may assume that we can reject this hypothesis as well when we further improve POIL regarding performance issues (e.g., revision of existing SIN queries, improvement of query processing by the SIN Facade).

#	Hypotheses	Rejected?
$H_{0,1}$	There is no significant difference in how intuitive the POIL framework is compared to hard-wired IL.	✓
$H_{0,2}$	There is no significant difference in how good the POIL framework identifies relevant process information compared to hard-wired IL.	✓
$H_{0,3}$	There is no significant difference in how fast a task can be performed with the POIL framework compared to hard-wired IL.	

Table 9.7: 0-hypotheses underlying the experiment.

In summary, the research questions from Table 9.4 can be answered with reasonable certainty: The POIL framework is suitable to close the gap between process information and business processes (RQ3.1). Moreover, it is more suitable for providing relevant process information than hard-wired IL (RQ3.2).

Generally, there are risks threatening the experiment results. Regarding the experiment, the threats of internal and external validity are as follows:

- **Subjects:** Different experience levels of subjects are a threat. To limit this, we only chose subjects from one industry domain. All participants were familiar with the considered use case. However, this fact explains the small number of subjects (i.e., twelve) in our experiment.

- **Objects:** Objects should not differ in more than one factor to make results comparable. We used the same business processes (e.g., the newsletter process) for the two objects (e.g., iProcess and the existing portal of our industry partner).

- **Experience:** All subjects were familiar with the existing portal of our partner. Therefore, a comparison between iProcess and the portal is difficult. We tried to take this issue into account when creating the statements for the experiment.

- **Business Process:** The newsletter process needs a lot of process information and creativity. However, it is not as knowledge-intensive as an engineering process in the automotive domain. Since the process is performed many times a month, we were quickly able to gather data about the frequency of used process information. Therefore, we were able to increase the precision of the POIL framework.

9.5 Summary

First, we presented the results of a survey conducted in the automotive domain. Results showed that most of the process information returned by the SIN LP algorithm is indeed relevant. Moreover, we showed that the link popularity values are good indicators for identifying relevant process information. The results of the SIN RP algorithm were considered as useful by the participants. In fact, most of them stated that the ranking of process information as suggested by the SIN RP algorithm is both plausible and useful.

Second, we presented the results of a case study in the automotive domain. Results showed that automated SIN maintenance is feasible with regard to both depth and breadth with acceptable costs. The push, pull and partial-pull algorithm performed satisfactorily in terms of adding, updating and deleting SIN objects and relationships.

Finally, we presented the results of an experiment conducted in the agricultural domain. Results showed that the POIL framework can indeed close the gap between process information and business processes. Moreover, the experiment showed that the POIL framework can better support process participants with needed process information compared to traditional enterprise portals. The experiment revealed that SN performance is a crucial factor for performing business processes in a more efficient way.

10 Performance Tests

This chapter presents results of POIL performance tests in the automotive domain. Section 10.1 introduces their design. Section 10.2 presents the results, which are then discussed in Section 10.3. Section 10.4 concludes the chapter with a summary.

10.1 Test Design

In practice, the technical performance of the POIL framework constitutes a critical success factor. Only if POIL is able to integrate and analyze process information and business processes in reasonable time, process participants can be provided with the right process information at the right point in time. In this context, criteria like *execution time*, *CPU utilization*, and *memory utilization* play an essential role.

We conducted performance tests to determine potential weaknesses of the POIL framework as well as options for its improvement. In this context, we investigated the core component of the POIL framework, i.e., the SIN. In particular, we measured the time needed for performing the SIN creation phases as introduced in Section 5.3.2. More specifically, we investigated the effects of the number of files as well as different file formats (e.g., xlsx, docx, pptx, pdf) and file sizes on the creation of the SIN.

Overall, the tests have been guided by three research questions (cf. Table 10.1):

#	**Research Questions**
RQ1	How does the format of files affect SIN creation in terms of time?
RQ2	How does the number of files affect SIN creation in terms of time?
RQ3	How does the size of files affect SIN creation in terms of time?

Table 10.1: Research questions underlying the performance tests.

The files used for the tests stem from an automotive OEM. Specifically, we had access to a project shared drive. In total, we considered 3,473 files with an overall size of 2.47 GB. 36% of these files were PowerPoint files, 31% were Excel files, 14% were Word files, and 19% were PDF files. The tests were performed on a laptop with an Intel quad-core CPU Intel Core i7 2670Q with 3.1 GHz, 16 GB RAM, 512 GB solid-state drive (SATA 6 Gbit/s), and a Windows 7 64-bit operating system.

For the performance tests, we used the open-source tool VisualVM [294]. The latter enables the creation of different reports about system properties, CPU load, thread-dump, heap-dump, classes, and core-dump. Moreover, VisualVM provides analysis features such as a CPU profiler, a thread analyzer, and a memory profiler.

The tests were accomplished in four steps (cf. Figure 10.1): (1) identification of relevant performance criteria, (2) selection of an appropriate test tool, (3) performing the tests based on the iQser GIN Server, and (4) investigation of the results.

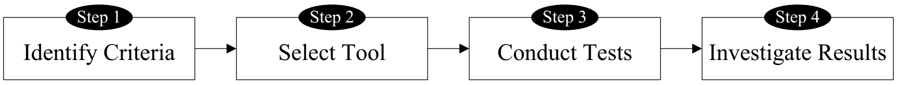

Figure 10.1: Steps of the performance tests.

The performance criteria of our tests are as follows: (1) *execution time*, (2) *CPU usage*, and (3) *memory usage*. The execution time indicates how long a test runs. Thereby, we distinguish between the time needed for the integration and the analysis. CPU usage, in turn, indicates how strongly the CPU is used during the execution of a particular test and what the average CPU usage is. Finally, memory usage indicates the seizure of memory. Thereby, we distinguish between the heap size and used size of the memory.

Overall, we conducted three test series. Each one comprises a set of test cases (cf. Tables 10.2-10.4). The latter are used to address the research questions (cf. Table 10.1).

Test Series I. In the first test series we wanted to know how long it takes to integrate and analyze different file formats in the SIN (RQ1). We tested different file formats such as Excel (xls, xlsx), Word (doc, docx), PowerPoint (ppt, pptx), and PDF, and investigated execution time, CPU usage, and memory usage. Moreover, the amount of files (i.e., 125, 250) and overall file sizes (i.e., 100 MB, 200 MB) were considered. Each file was bigger than 20 KB, but smaller than 5 MB (cf. Table 10.2).

#	Series	Formats	Sizes	Files	Smallest File	Biggest File
1.1	I	[xls][xlsx]	100 MB	125	>20 KB	<5 MB
1.2	I	[doc][docx]	100 MB	125	>20 KB	<5 MB
1.3	I	[ppt][pptx]	100 MB	125	>20 KB	<5 MB
1.4	I	[pdf]	100 MB	125	>20 KB	<5 MB
1.5	I	[xls][xlsx]	200 MB	250	>20 KB	<5 MB
1.6	I	[doc][docx]	200 MB	250	>20 KB	<5 MB
1.7	I	[ppt][pptx]	200 MB	250	>20 KB	<5 MB
1.8	I	[pdf]	200 MB	250	>20 KB	<5 MB

Table 10.2: Test cases of series I.

Test Series II. In the second test series, we wanted to know how long it takes to integrate and analyze varying amounts of files in the SIN (RQ2). Therefore, we created four test cases (i.e., 5.1-5.4) and considered execution time, CPU usage, and memory usage. We considered mixed file formats, different number of files (i.e., 125, 250, 500, 1,000), and different overall sizes (i.e., 100 MB, 200 MB, 400 MB, 800 MB). Like in series I, each file was bigger than 20 KB, but smaller than 5 MB (cf. Table 10.3)

#	Series	Formats	Sizes	Files	Smallest File	Biggest File
2.1	II	[all]	100 MB	125	>20 KB	<5 MB
2.2	II	[all]	200 MB	250	>20 KB	<5 MB
2.3	II	[all]	400 MB	500	>20 KB	<5 MB
2.4	II	[all]	800 MB	1,000	>20 KB	<5 MB

Table 10.3: Test cases of series II.

Test Series III. In the third test series, we wanted to know how long it takes to integrate and analyze different file sizes (RQ3). Each file was bigger than 0 KB, but smaller than 16 MB (depending on the test case). We used mixed file formats and 100 files for each of our six test cases (cf. Table 10.4). Like before, all criteria were considered.

#	Series	Formats	Sizes	Files	Smallest File	Biggest File
3.1	III	[all]	–	100	>=4 MB	<16 MB
3.2	III	[all]	–	100	>=2 MB	<4 MB
3.3	III	[all]	–	100	>=1 MB	<2 MB
3.4	III	[all]	–	100	>=500 KB	<1 MB
3.5	III	[all]	–	100	>=100 KB	<500 KB
3.6	III	[all]	–	100	>0 KB	<100 KB

Table 10.4: Test cases of series III.

10.2 Results

Test Series I. When considering the file format (Tests 1.1-1.4), the execution time is almost identical. More specifically, the time is ranging from 01:50 mm:ss (PowerPoint) to 02:52 mm:ss (Excel) (cf. Figures 10.2-10.5). A reason might be that PowerPoint files contain little text and can therefore be quickly processed. When considering the file format together with a larger number of files (cf. Figures 10.6-10.9), results can be confirmed. However, when considering Tests 1.5 and 1.9, we can see that PDF files can be integrated and analyzed much faster than Excel files; i.e., 11:08 mm:ss for Excel files and 06:31 mm:ss for PDF files. During the tests with a lower number of files, CPU usage is ranging from 59.96% to 78.41%, whereas memory usage is ranging from 284 MB to 331 MB. When considering a larger number of files, CPU usage is ranging from 69.32% to 84.51%, whereas memory usage is ranging from 317 MB to 356 MB.

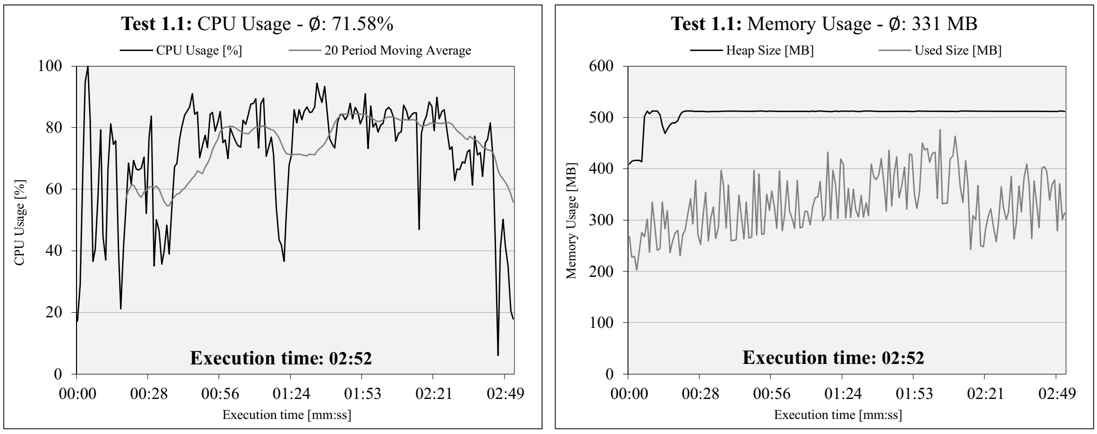

Figure 10.2: Execution time, CPU usage, and memory usage of Test 1.1.

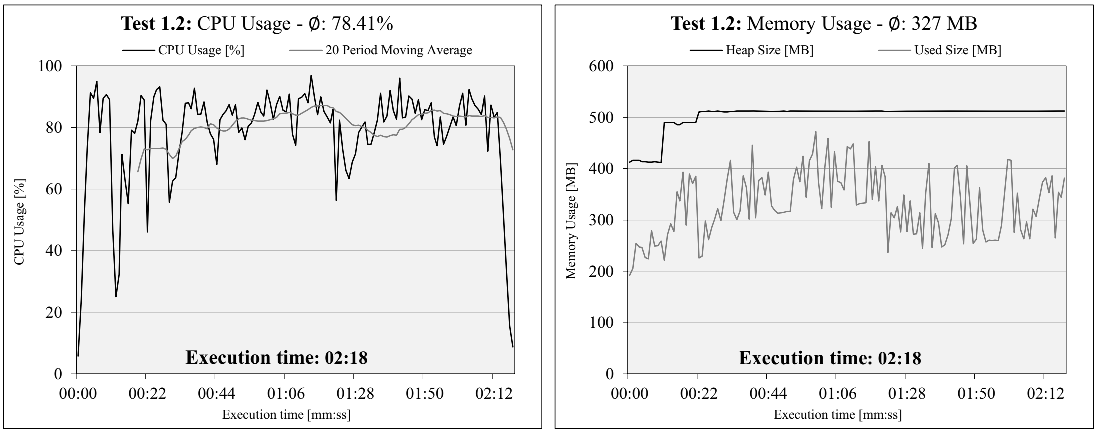

Figure 10.3: Execution time, CPU usage, and memory usage of Test 1.2.

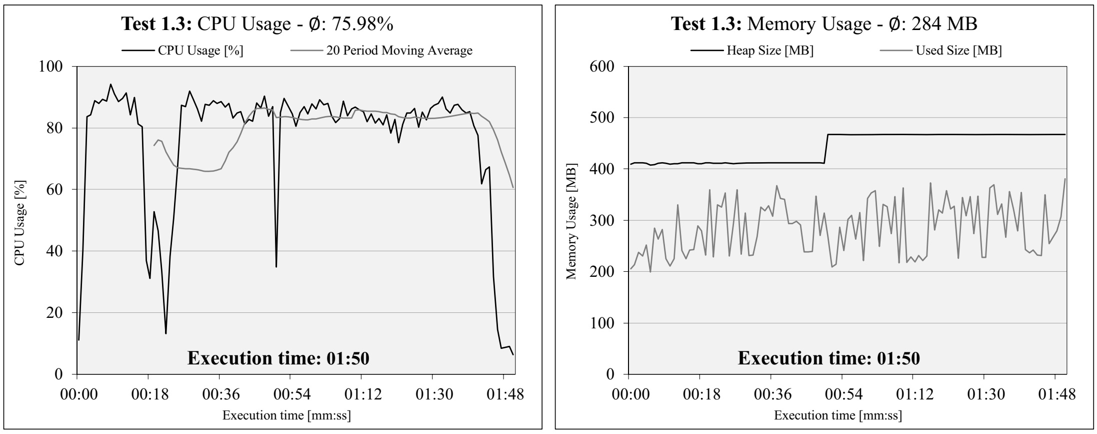

Figure 10.4: Execution time, CPU usage, and memory usage of Test 1.3.

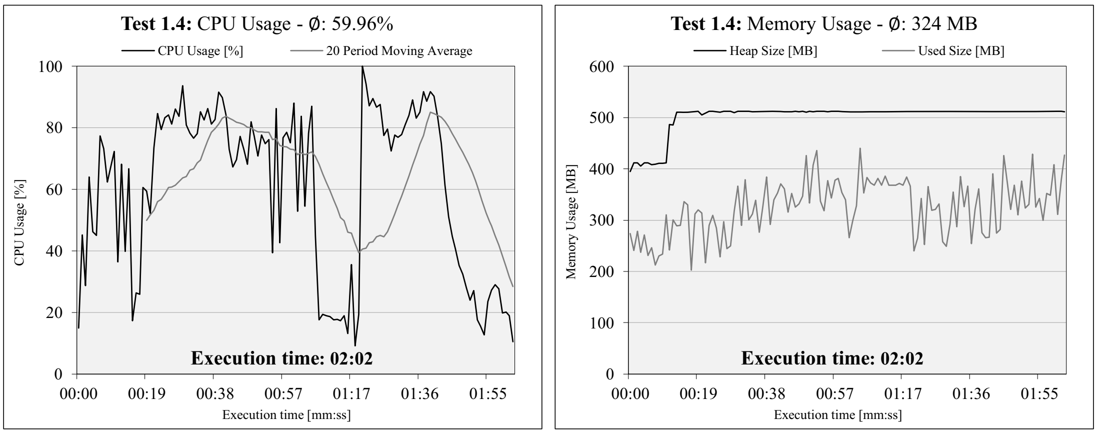

Figure 10.5: Execution time, CPU usage, and memory usage of Test 1.4.

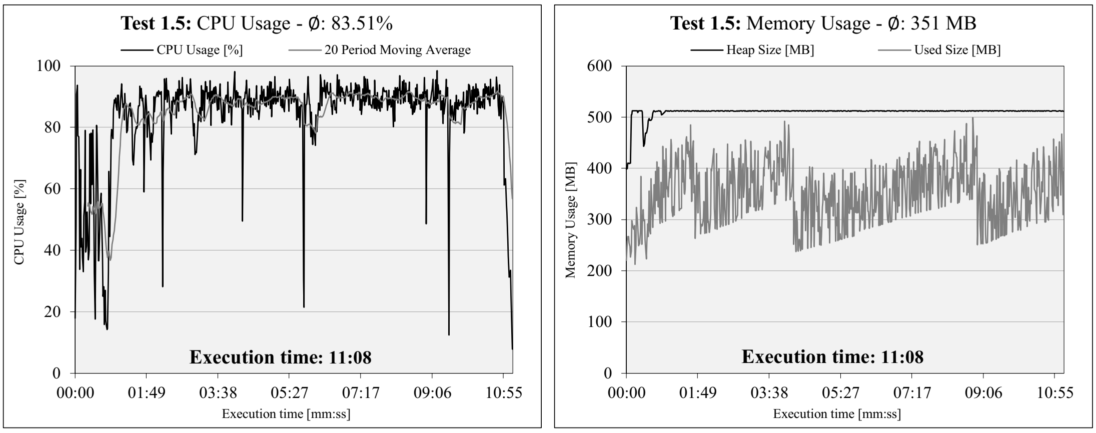

Figure 10.6: Execution time, CPU usage, and memory usage of Test 1.5.

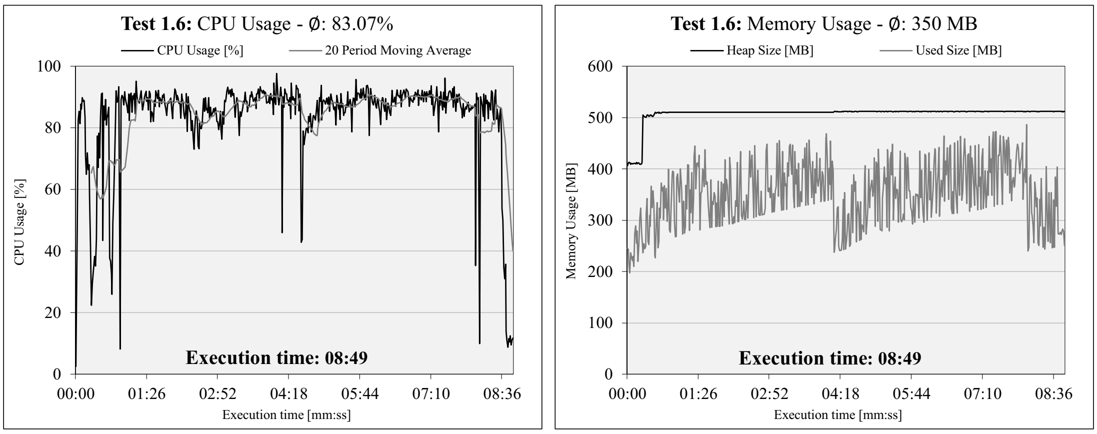

Figure 10.7: Execution time, CPU usage, and memory usage of Test 1.6.

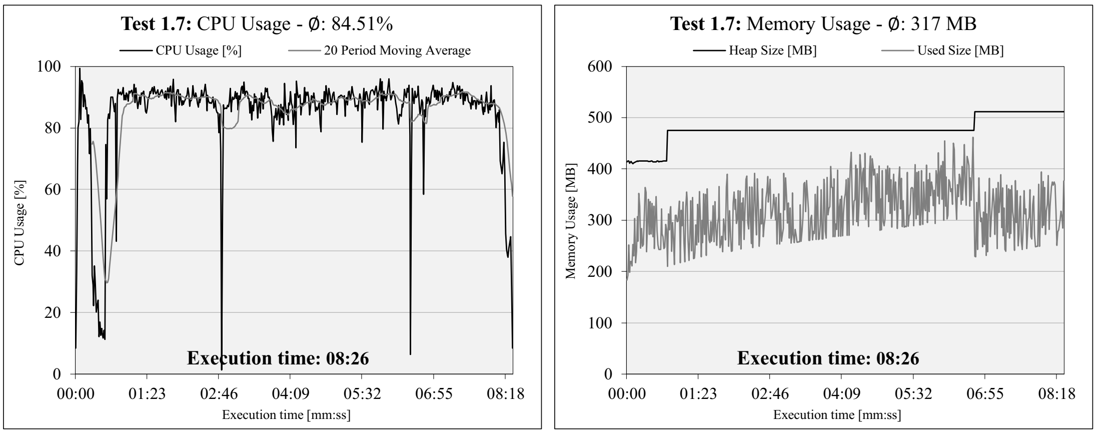

Figure 10.8: Execution time, CPU usage, and memory usage of Test 1.7.

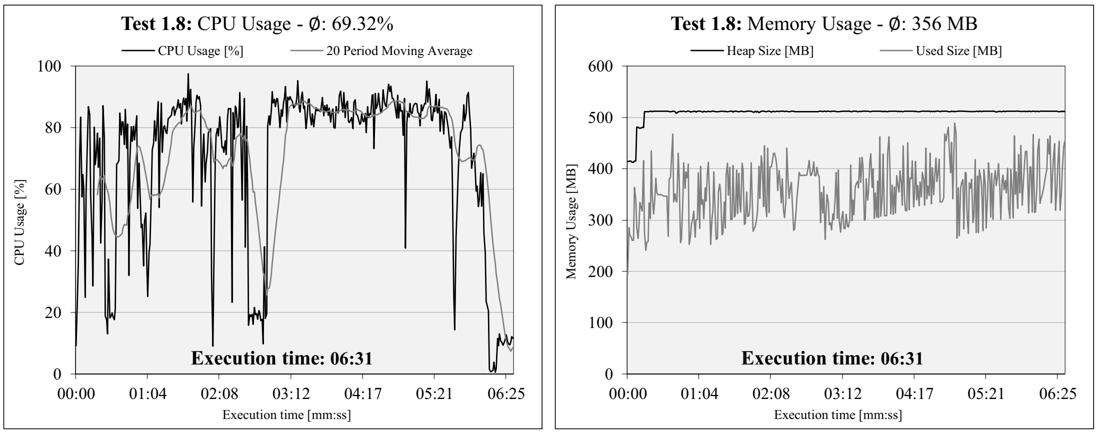

Figure 10.9: Execution time, CPU usage, and memory usage of Test 1.8.

Test Series II. When considering the number of files (Tests 2.1-2.4), the execution time varies (cf. Figures 10.10-10.13). The smaller the number of files is, the smaller the execution time will be. For example, the integration and analysis of 250 files is done in 07:58 mm:ss, whereas 500 files are integrated and analyzed in 46:16 mm:ss (cf. Figures 10.11 and 10.12). Therefore, the time needed for one file is significantly higher (1.912 seconds vs. 5.552 seconds per file). The reason for this issue is the analysis since each integrated file must be compared to each and every other file in the SIN. During the tests, CPU usage is ranging from 78,62% to 92.41% on average. The higher the number of files is, the higher the average CPU usage will be. In turn, when considering memory usage, we can notice that the usage is ranging from 327 MB to 376 MB on average. Again, the higher the number of files is, the higher the average memory usage will be.

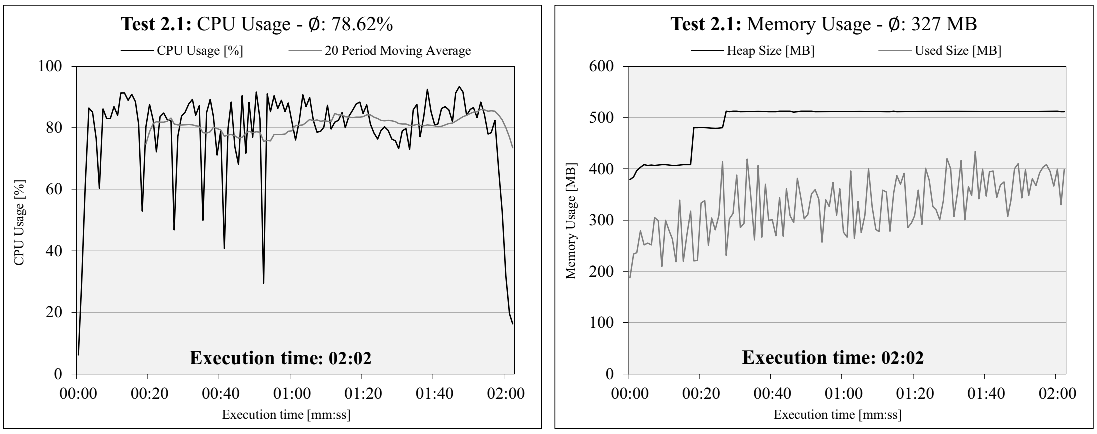

Figure 10.10: Execution time, CPU usage, and memory usage of Test 2.1.

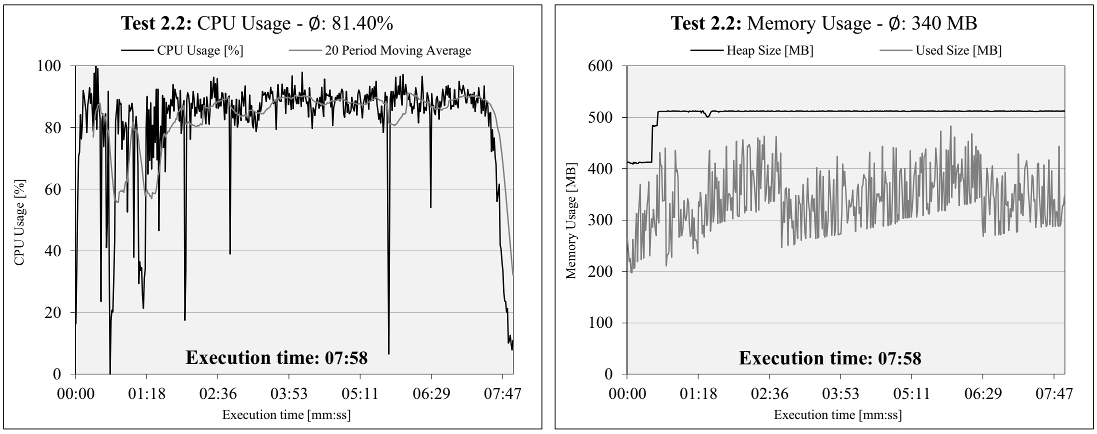

Figure 10.11: Execution time, CPU usage, and memory usage of Test 2.2.

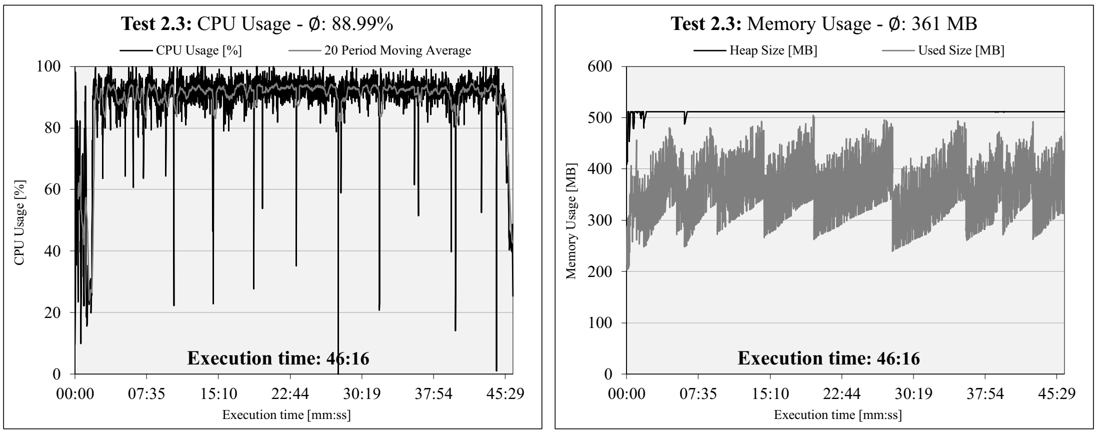

Figure 10.12: Execution time, CPU usage, and memory usage of Test 2.3.

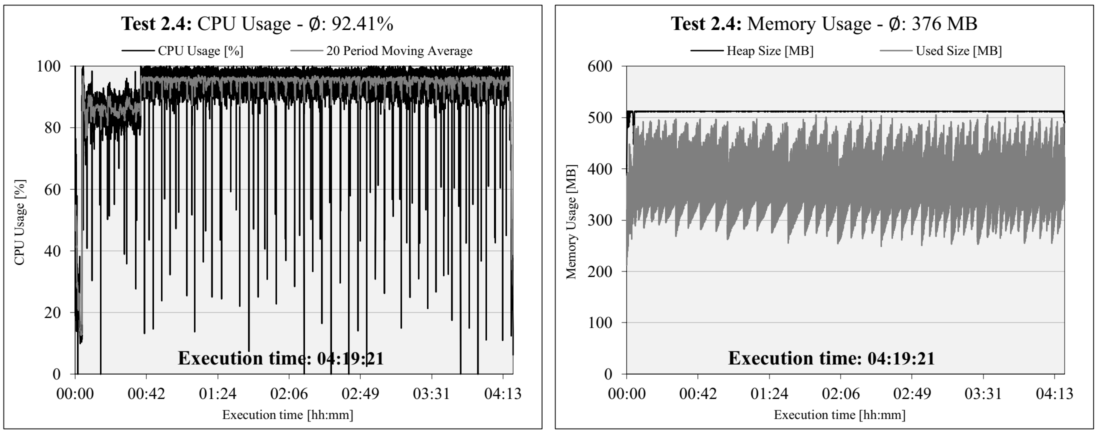

Figure 10.13: Execution time, CPU usage, and memory usage of Test 2.4.

Test Series III. When considering file size (Tests 3.1-3.6), the execution time is varying again. The smaller the file size is, the smaller the execution time will be. For example, for 100 files ranging from 4 MB to 16 MB 02:41 mm:ss are needed. In turn, for 100 files ranging from 2 MB to 4 MB only 02:03 mm:ss are needed (cf. Figures 10.14-10.19). This result might be explained with the fact that smaller files contain little text and can therefore be faster processed. When considering CPU usage, the latter is ranging from 54.01% to 70.79% on average. Thereby, no dependencies between file size and CPU usage can be identified in the performance tests. In turn, when considering the memory usage, we can notice that the usage is ranging from 303 MB to 340 MB on average.

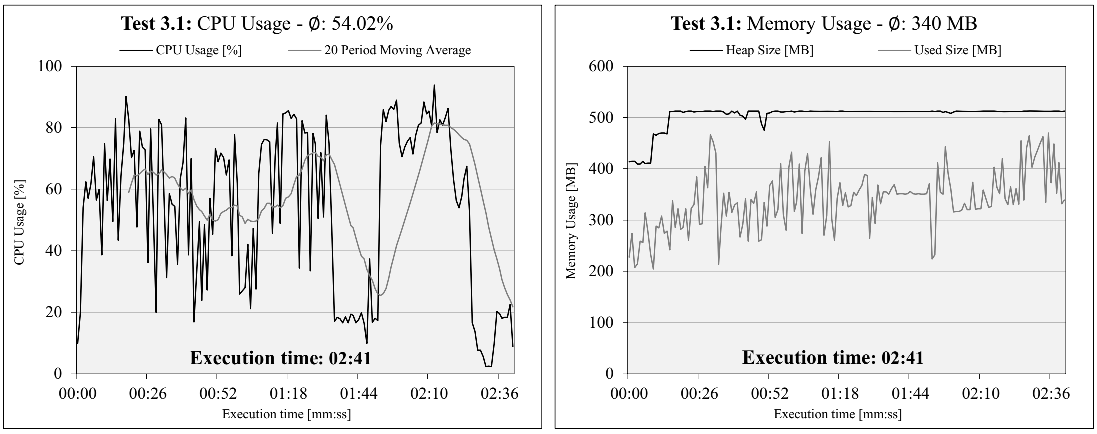

Figure 10.14: Execution time, CPU usage, and memory usage of Test 3.1.

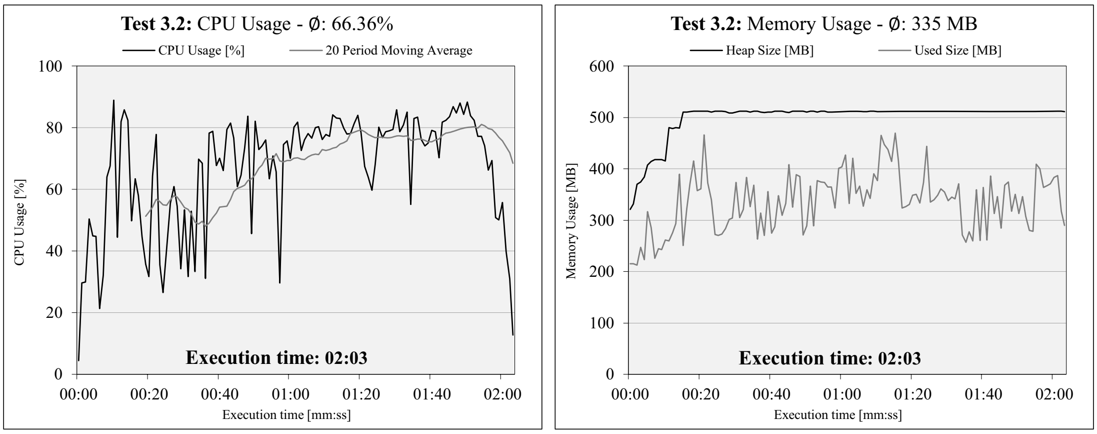

Figure 10.15: Execution time, CPU usage, and memory usage of Test 3.2.

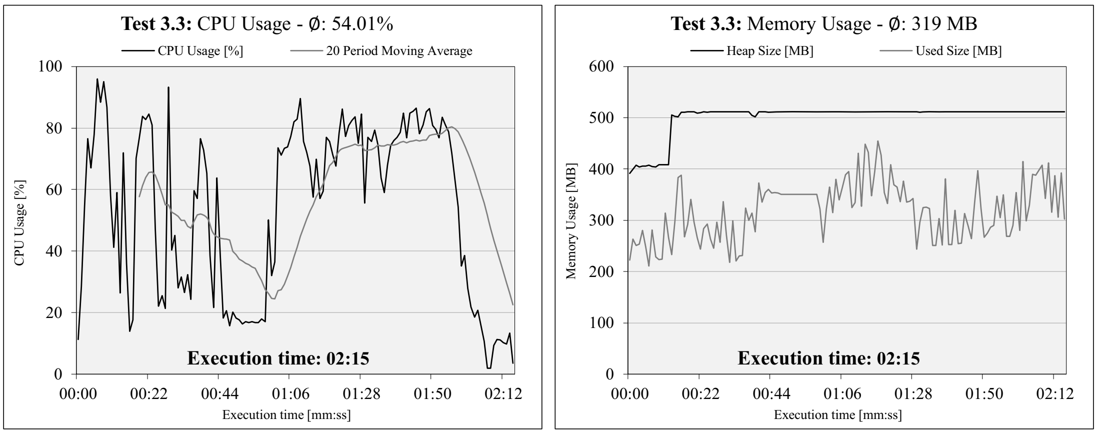

Figure 10.16: Execution time, CPU usage, and memory usage of Test 3.3.

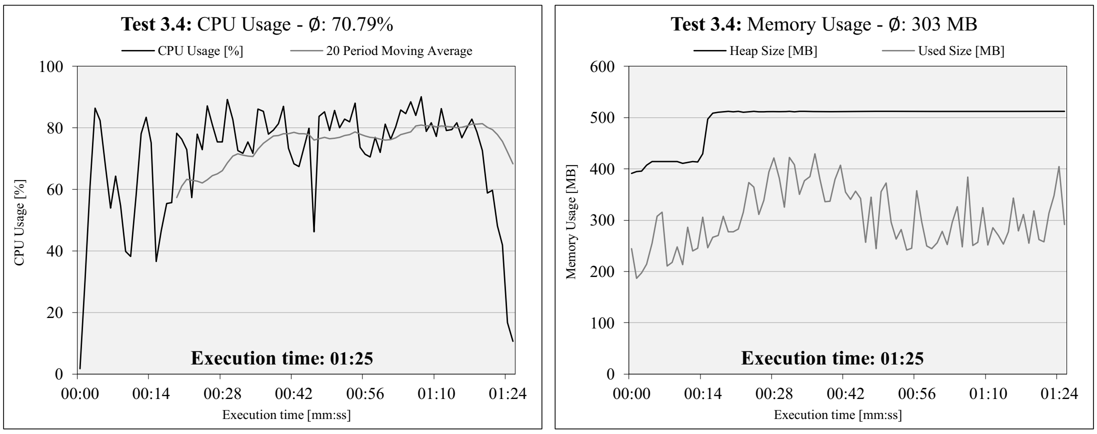

Figure 10.17: Execution time, CPU usage, and memory usage of Test 3.4.

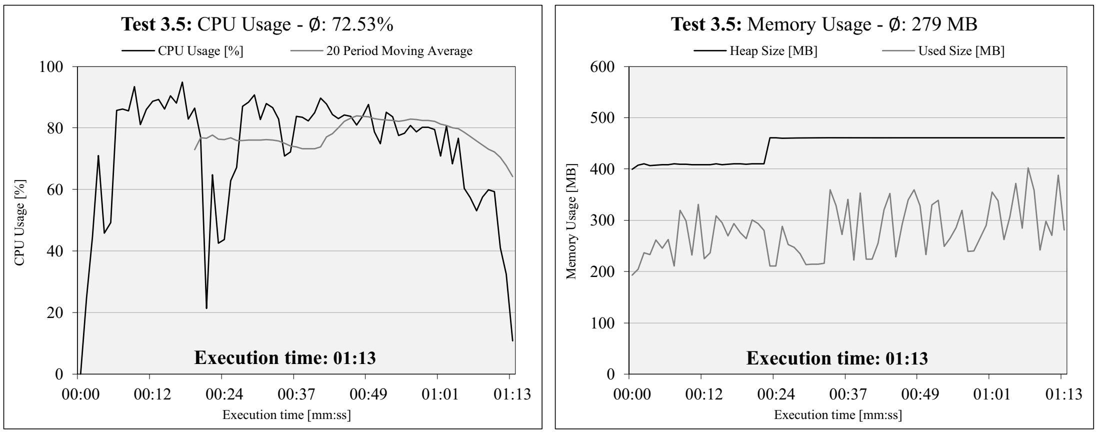

Figure 10.18: Execution time, CPU usage, and memory usage of Test 3.5.

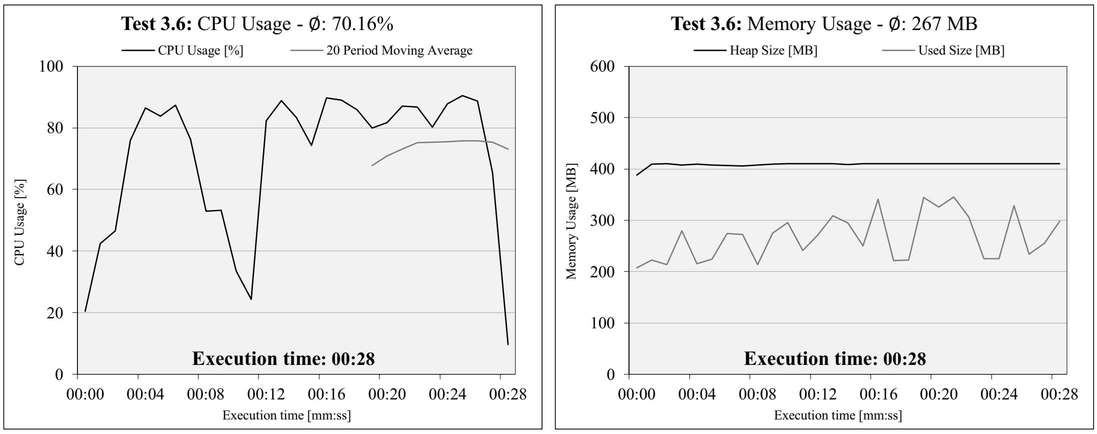

Figure 10.19: Execution time, CPU usage, and memory usage of Test 3.6.

Further Tests. When considering all test cases (i.e., Tests 1.1-3.6), we can observe that the size of memory is a limited factor. Therefore, we increased the heap size from 512 MB to 4096 MB and executed the Tests 2.1-2.4 again. Goal was to investigate whether the execution time can be significantly reduced. Figures 10.20-10.23 show that we can indeed decrease execution time. More specifically, the execution time of Test 2.1 can be decreased by 18.85%, of Test 2.2 by 26.78%, of Test 2.3 by 46.58%, and of Test 2.4 by 52.61%. These results confirm that we can increase the performance by upgrading the hardware (e.g., CPU, internal memory, hard disk drive).

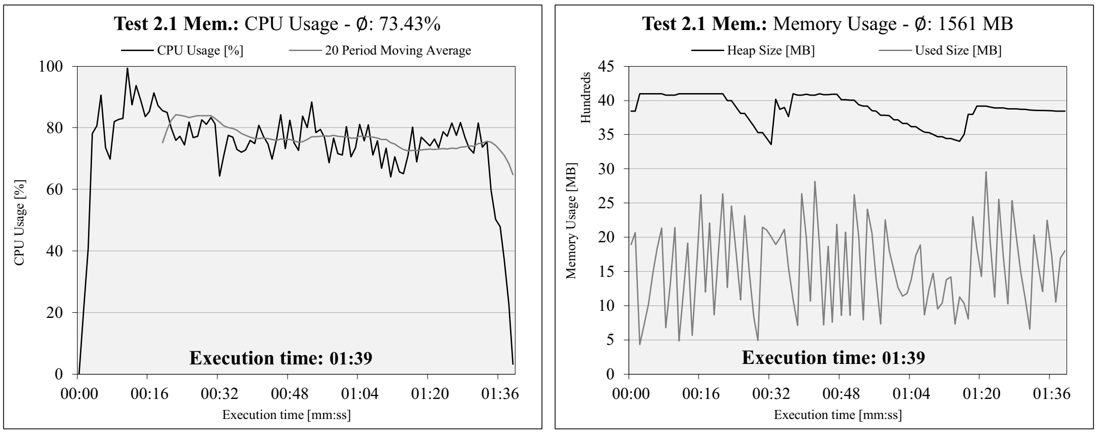

Figure 10.20: Decreasing the execution time of Test 2.1 (memory case).

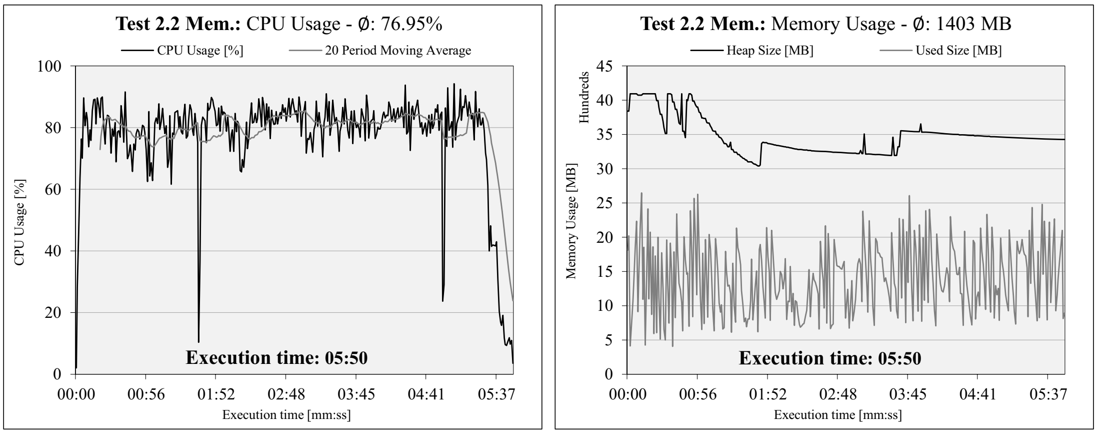

Figure 10.21: Decreasing the execution time of Test 2.2 (memory case).

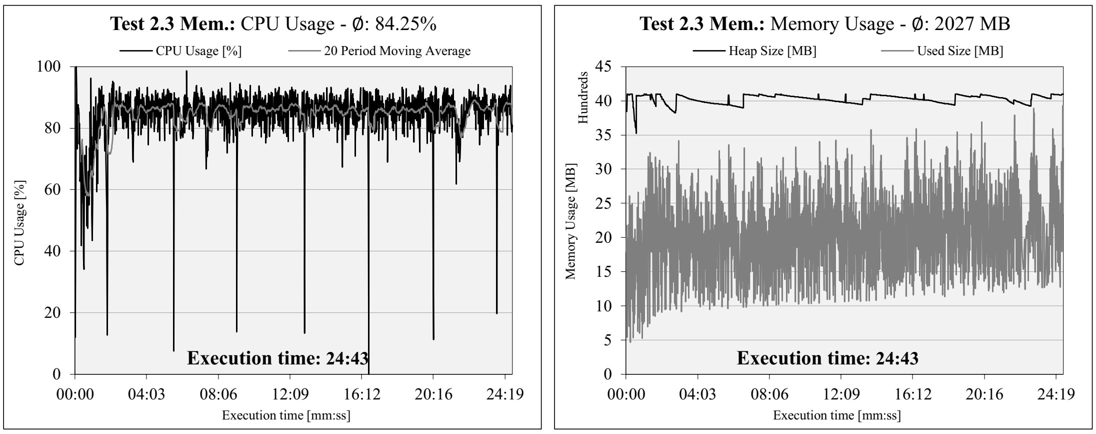

Figure 10.22: Decreasing the execution time of Test 2.3 (memory case).

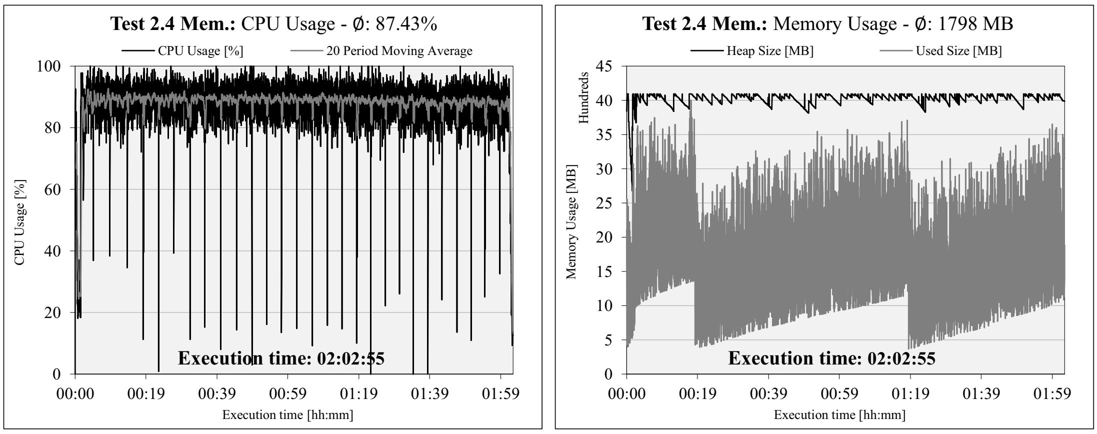

Figure 10.23: Decreasing the execution time of Test 2.4 (memory case).

10.3 Discussion

The results of the performance tests confirm that POIL is able to integrate and analyze process information in reasonable time. The tests have shown that different file formats affect the SIN creation (RQ1). PowerPoint and PDF files can be processed faster than other file formats. Moreover, the tests have shown that the number of files influences the SIN as well. The larger the number of files is, the longer the execution time will be (RQ2). Finally, different file sizes affect the SIN as well. The larger the files are, the longer is the execution time (RQ3). Figure 10.24 summarizes the execution times.

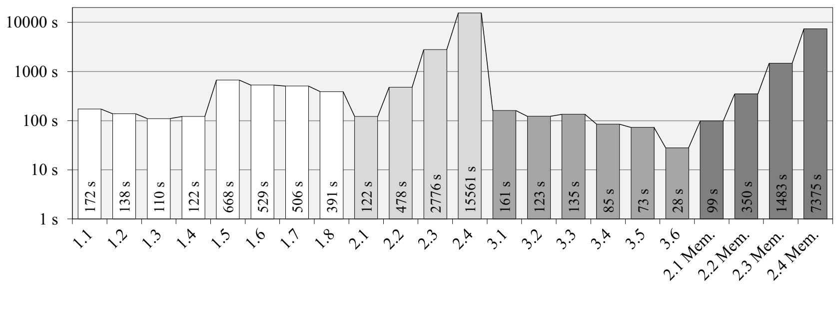

Figure 10.24: Execution times of the tests.

To increase the overall SIN performance, CPU and memory can be upgraded. When increasing the size of memory, we are able to decrease execution time up to 52.61%.

In summary, the POIL framework performs well in practice. POIL is able to create a SIN in reasonable time and can therefore provide the right process participant with the right process information in the right format at the right point in time.

10.4 Summary

This chapter presented the results of POIL performance tests in the automotive domain. First, we investigated how file formats affect SIN creation. Second, we investigated the impact the number of files has on SIN creation. Third, we analyzed how file sizes affect SIN creation. Based on the results, we showed that SINs can be created in reasonable time and that POIL components are performing well.

Part IV

Discussion and Summary

11 Discussion

This chapter discusses the major achievements of the POIL framework as well as strengths and limitations. Section 11.1 presents a SWOT analysis. Section 11.2 then presents the research activities performed in the context of the thesis. Section 11.3 discusses results in relation to the research questions addressed by the thesis.

11.1 SWOT Analysis

Evaluating the practical benefits of POIL is a complex task to accomplish. We perform a SWOT analysis [295] to discuss the POIL framework (cf. Figure 11.1). In particular, we address *strengths*, *weaknesses*, *opportunities*, and *threats*. Strengths correspond to characteristics of the POIL framework giving it advantage over other approaches. Weaknesses, in turn, are characteristics that place the POIL framework disadvantageously compared to other approaches. Furthermore, opportunities are characteristics the POIL framework could exploit to its advantage. Finally, threats are characteristics of the environment that might trouble POIL. Note that the characteristics identified in the SWOT analysis are based on practical experiences gathered in a follow-up research project of the niPRO project.

	Helpful for achieving the goals	Harmful for achieving the goals
Internal origin	**Strengths** • process-aware delivery of process information • automatically identified relationships • processes can be performed more efficiently • independent from the use of ICT • existing data sources can be further used	**Weaknesses** • formalization of business processes needed • connector for each data source needed • no common standards are available • security issues • performance issues
External origin	**Opportunities** • added value to existing applications • discovery of unknown relationships	**Threats** • other concepts (e.g., semantic web or big data) • complicated laws and regulations • complicated privacy policies in enterprises

Figure 11.1: SWOT analysis of the POIL framework.

Strengths. POIL supports process-awareness that is missing in contemporary IL approaches. POIL automatically discovers unknown relationships between process information and business processes based on semantic technology. Therefore, process information and business processes need not be manually linked by dedicated administrators.

As a consequence, maintenance efforts and costs can be significantly reduced. Even more important, through the alignment of process information and business processes, process participants can perform their tasks more efficiently. They are provided with the right process information at the right point in time. Further note that POIL is independent from the use of ICT. Existing data sources, such as shared drives or databases, can be easily used since POIL is realized as a middleware infrastructure.

Weaknesses. The business processes to be supported by POIL must be explicitly specified, e.g., using a process modeling language such as BPMN or EPC. Only such an explicit process description allows us to automatically transform a process schema and the corresponding process instances into SIN process objects. Additionally, for each data source, an interface must be implemented. Its main task is to transform proprietary process information into SIN information objects. Moreover, no common standards exist for representing business processes and process information in a SIN or querying a SIN for more details. Finally, the current development state of POIL lacks security and performance features (see the last two paragraphs of this chapter for more details).

Opportunities. POIL increases the value of existing enterprise applications. For example, an existing *enterprise search engine* [296, 297] might be combined with POIL to further improve the overall accuracy of search results. Moreover, unknown relationships between process information can be determined by POIL. This enables new perspectives on existing enterprise (process) information.

Threats. POIL overlaps with other concepts such as *semantic web* [298] or *big data* [299, 300]. Other threats are complicated laws, regulations and privacy policies. For example, in enterprises it is often not allowed tracking information (e.g., times, clickstreams and queries) of knowledge workers and decision makers.

11.2 Research Activities

During the development of POIL, we conducted both empirical and non-empirical research activities in the research phases as introduced in Section 1.4; i.e., (1) problem analysis, (2) requirements analysis, (3) solution design, and (4) solution validation.

Design Science Research Framework. We mainly used the research principles of design science as introduced in Section 1.4, since the goal of this thesis is to provide a solution (i.e., the POIL framework) for a real-world problem (i.e., the gap between process information and business processes) by creating innovative artifacts. We used several methods such as case studies, experiments, prototypes, and use cases in order to evaluate the POIL framework. We provided contributions in the area of the design artifact. In addition, we applied rigorous techniques such as graph theory or linear algebra. Moreover, we studied prototypes that instantiate posed or newly learned design prescriptions from our research. We provided comprehensive information to technical as well as managerial audience. Finally, we introduced technical implementations but also risks and benefits when applying the POIL framework.

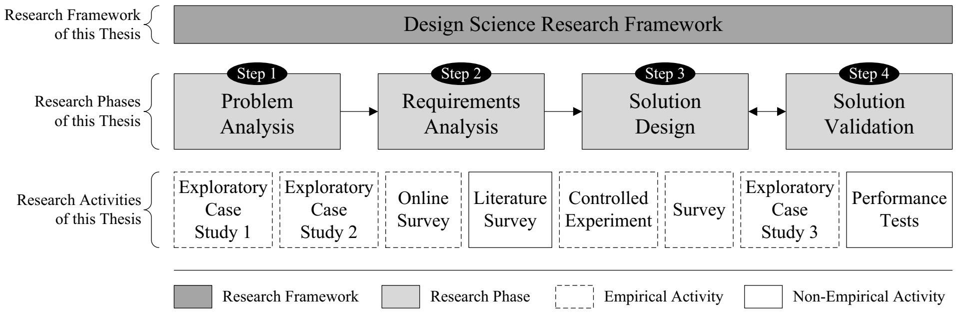

Figure 11.2: Research activities along research phases.

Step 1: Problem Analysis. During the problem analysis, the gap between process information and business processes was investigated. In practice, process information is not only stored in large, distributed and heterogeneous data sources, but also managed separately from business processes. Hence, in practice, process information and business processes are only linked manually, statically and partially, e.g., in enterprise portals connecting specific process information with business processes and associated process tasks. To cope with this problem, the POIL framework automatically aligns process information with business processes and their tasks in a context-aware manner. In this case, the problem analysis was supported through the following research activities:

- **Exploratory Case Study 1 (cf. Section 3.3):** We conducted an exploratory case study in the automotive domain. We analyzed business processes such as the review of product requirements or the identification of system specifications. We performed eight interviews and received other questionnaires with additional data. In particular, this case study allowed us to identify problems related to the gap that exists between process information and business processes.

- **Exploratory Case Study 2 (cf. Section 3.4):** Moreover, we conducted an exploratory case study in the clinical domain. Again, we analyzed business processes such as the admission of patients to a surgical clinic. We performed eight interviews and received questionnaires with further data. The clinical case study allowed us to compare results with the automotive case study and to gain further insights helping us to generalize our problem investigation.

Step 2: Requirements Analysis. During this step, requirements enabling POIL were elicited. The requirements reflect wishes and needs of process participants such as knowledge workers and decision makers. They further concern technical issues enabling the delivery of relevant process information to process participants. In this case, the requirements analysis was supported through the following research activities:

- **Exploratory Case Studies 1 and 2 (cf. Sections 3.3 and 3.4):** From the aforementioned case studies, we also identified requirements. These are mainly driven from a practical perspective since the POIL requirements were derived from the interviews and filled-out questionnaires.

- **Online Survey (cf. Section 3.5):** We conducted an online survey with 219 employees from more than 100 enterprises. It allowed us to increase the validity of our case study results.

- **Literature Survey (cf. Section 3.6):** We conducted a comprehensive literature survey. Its goal was to approve the requirements identified during the empirical studies. Additionally, the literature survey allowed us to derive further POIL requirements.

Step 3: Solution Design. During the solution design, we created the POIL framework to meet the identified requirements. POIL aligns process information with business processes, both at the process schema and instance level [53]. In turn, this enables a process-oriented, context-aware delivery of process information to process participants. The main idea of POIL is to split up business processes into their constituent process elements and to integrate the latter with comprehensive process information [7]. As mentioned in Chapter 4, enabling the POIL framework requires four architectural layers: *data layer*, *semantic layer*, *context layer*, and *application layer* (cf. Figure 11.3).

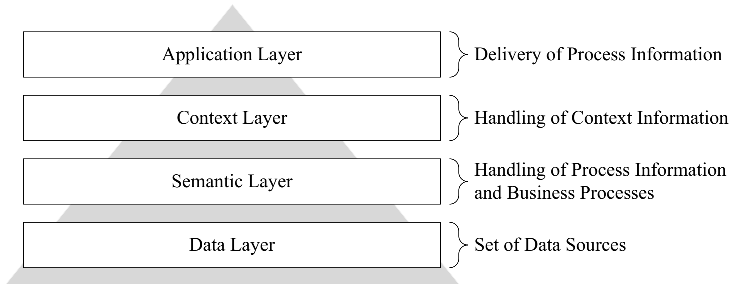

Figure 11.3: POIL architecture levels.

Step 4: Solution Validation. During this step, the POIL framework was thoroughly validated. Unlike the other research activities, the solution validation investigated the concepts of the POIL framework. The solution validation analyzed whether the POIL framework works as desired and fulfills the expectations and requirements. For this purpose, prototypes, statistical methods, and test were used. The goal of the solution validation was to demonstrate the applicability and feasibility of the POIL framework. Moreover, lessons learned from the development and application of the prototypes were taken into account as well. In this case, the solution validation was supported through the following research activities:

- **Survey (cf. Section 9.2):** We conducted a survey in the automotive domain to prove that the SIN LP and SIN RP algorithm (cf. Sections 5.5.1 and 5.5.2) actually support process participants when performing knowledge-intensive business processes. We applied the algorithms to a real-world use case from the automotive domain and then compared their outcome with the results of a survey among experienced automotive engineers. Overall, 20 experts from an automotive OEM participated. Particularly, results have shown that the algorithms can indeed replace the costly and time-intensive human determination of relevant process information.
- **Exploratory Case Study 3 (cf. Section 9.3):** We conducted a case study in the automotive domain. Unlike the first two case studies, this case study proved the validation of the POIL framework. More specifically, we proved that the SIN maintenance algorithms presented in Section 5.4.3 are able to maintain SINs. Parts of this empirical validation were interviews with employees regarding the usefulness of SIN maintenance. Overall, eleven employees from an automotive manufacturer were interviewed. Results confirm that the algorithms can indeed replace the costly and time-intensive human maintenance of SINs (cf. Section 9.3.3).
- **Controlled Experiment (cf. Section 9.4):** We conducted a controlled single factor experiment in the agricultural domain. Its main goal was to investigate the benefits of a process information portal implementing POIL compared to a conventional enterprise portal implementing hard-wired IL. Overall, twelve employees participated. They mainly stemmed from the sales department, but also from the department responsible for project management. During the development of the process information portal, we gained a lot of know-how and experience important for the design of POIL. Specifically, we implemented the architectural layers of the POIL framework (except for the context layer) in a process information portal called iProcess (cf. Section 9.4.1). We then compared the experimental results of POIL with the ones of a hard-wired IL solution, which has been used by the involved enterprise (cf. Section 9.4.2). The results show that the POIL framework can indeed close the gap between process information and business processes. The framework can decrease the time needed for handling and managing process information during business process execution (cf. Section 9.4.3).
- **Performance Tests (cf. Section 10.2):** We conducted 22 performance tests to determine potential weaknesses of the POIL framework as well as options for its improvement. In particular, we investigated the core component of POIL; i.e., the SIN. We measured the time needed for performing the SIN creation phases as introduced in Section 5.3.2.

11.3 Research Questions

Based on the discussed results, we can summarize the answers to our research questions as depicted in Section 1.2. Note that the detailed results are described in the respective chapters. Figure 11.4 relates each research question to one or more chapters.

	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11	Chapter 12
Research Question 1	o	+			o	o	o	o			o	o
Research Question 2			o	o	+			o	o	o	o	o
Research Question 3			o	o		+		o	o	o	o	o
Research Question 4			o	o	+			o	o	o	o	o
Research Question 5				o			+	o	o		o	o

+ = main results for the research question, o = further results for the research question

Figure 11.4: Research questions along chapters.

- **Research Question 1:** *What are existing approaches that may be used to provide relevant process information to process participants?* In recent years, various approaches were proposed, including data warehousing [43], business intelligence [44], decision support systems [45], and enterprise content management [46]. However, these approaches have not primarily been designed with POIL in mind. Data warehousing, for example, rather focuses on the creation of an integrated database [47]. Opposed to this, POIL deals with the delivery of process information to support the effective and efficient execution of business processes. Traditional business intelligence, in turn, enables data analysis and is usually completely isolated from business process execution [48]. Moreover, information supply is often restricted to decision makers on the management level [49, 50]. Conversely, POIL focuses on the integration and analysis of process information as well as its delivery to both knowledge workers and decision makers. By contrast, decision support systems support decision making; i.e., they serve the management level [51]. Enterprise content management, in turn, deals with the management of information across enterprises referring to strategies, methods and tools [52].

- **Research Question 2:** *How can the gap between process information and business processes be bridged?* The semantic layer of POIL allows bridging the gap between process information and business processes. More specifically, the semantic layer integrates process information and business processes. In particular, it discovers relationships between process information and business processes that have not been known so far. The core component of the semantic layer is the SIN. It is used to represent process information and business processes in a meaningful machine- and user-interpretable form. A SIN can be created following a bottom-up approach; i.e., starting with the integration of process information and business processes from different data sources. Thus, integrated process information and business processes are analyzed. The resulting SIN comprises *information objects*, *process objects*, and inter-object *relationships*, and thus closes the gap.

- **Research Question 3:** *Which context information is needed to characterize the work context of process participants?* Context information allows us to characterize the process participant's situation. Process-related context information (e.g., temporal process constraints, milestones), user-related context information (e.g., user name, experience level), device-related context information (e.g., display size, bandwidth), location-based context information (e.g., position), time-based context information (e.g., current date), and environment-related context information (e.g., temperature, humidity) are considered in the POIL framework.

- **Research Question 4:** *How can the relevance of process information for a specific business process and its process tasks be determined?* The relevance of process information can be identified based on the proposed algorithms. In practice, for example, there exist many specific review templates for review processes. Depending on the concrete process, therefore, specific review templates are relevant and hence need to be delivered to the process participants. In POIL, the SIN provides the basis for this task. However, specific techniques and algorithms are needed to determine relevant process information; i.e., currently needed information objects in a SIN dependent on the work context. The reason for this is that the SIN just identifies objects linked to each other for some reasons, but does not consider additional influence factors. We introduced algorithms for identifying relevant information objects in a SIN. The first one (i.e., the SIN LP) determines the link popularity of information objects based on the relationship structure of a SIN (cf. Section 5.5.1). The second one (i.e., the SIN RP) calculates the rate popularity of information objects based on user ratings (cf. Section 5.5.2). Note that these algorithms may be used independently, but also in combination with each other.

- **Research Question 5:** *How can process information, business processes, and context information be combined to enable process-oriented and context-aware process information delivery to process participants?* Generally, the application layer of the POIL framework allows for this combination. The layer is realized by the SIN Facade as introduced in Section 7.3. The SIN Facade aims at the context-aware delivery of process information to process participants. Thereby, it constitutes an interface to retrieve both information and process objects from the SIN taking context objects from the CM into account. Overall, the application layer allows applications to query the SIN as well as to filter the latter by the use of the CM. Moreover, the application layer is responsible for security issues. For example, if a process participant queries the SIN, the layer should filter the results and remove the objects for which a process participant has no access rights.

Besides the benefits mentioned when answering the research questions, the implemented POIL framework has revealed limitations that are discussed in the following.

It is not possible to grant or deny access permission to process participants when querying the SIN. As a consequence, each process participant has access to all information and process objects stored in the SIN. Furthermore, supporting short-time business processes by POIL constitutes a non-trivial task. Reason is that integrating and analyzing process information and business processes in the SIN will take some time. In the meantime, a particular process instance may have already been processed before the POIL framework can offer support for process participants. Therefore, the performance of the creation and maintenance phases of the SIN must be further improved.

Another limitation we have not considered is SIN evolution in an enterprise over a longer period of time (i.e., more than two years). Therefore, it is difficult to estimate how a SIN evolves over time, when, for example, more than 100 process participants work with the SIN during daily work over a longer period of time. However, the evaluation of a SIN in such a situation is a non-trivial task since its operative use is a prerequisite. In addition, we must investigate the combination of the proposed algorithms; i.e., we must answer the question which algorithms shall be used for which use case. For this purpose, a retrieval function for the SIN algorithms must be developed.

12 Summary and Outlook

This thesis introduced *process-oriented information logistics* (POIL) as a new paradigm for delivering the right process information, in the right format and quality, at the right place and the right point in time, to the right people. Missing process-awareness in contemporary *information logistics* (IL) has guided the development of POIL. In particular, POIL allows for the process-oriented and context-aware delivery of process information to process participants. The overall goal is to no longer manually link business processes with required process information, but to automatically identify and deliver relevant process information to knowledge workers and decision makers involved in various processes. Accordingly, POIL not only tackles the problem dimensions *information-* and *context-awareness*, but takes *process-awareness* into account as well (cf. Figure 12.1).

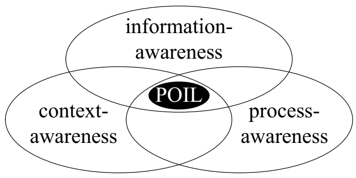

Figure 12.1: Problem dimensions: POIL.

The core component of POIL is a *semantic information network* (SIN) comprising *information objects* (e.g., e-mails, office files, guidelines, best practices), *process objects* (e.g., tasks, events, roles), and *relationships* between them. In particular, the SIN serves as basis for discovering objects linked with each other in different ways, e.g., objects addressing the same topic or objects needed when performing a particular process task.

The SIN not only enables an integrated formal representation of process information and business processes, but also allows determining the relevance of process information for a given work context based on novel techniques, methods and algorithms. Note that this was crucial to achieve the aforementioned overall goal of this thesis.

Our research has started with an analysis of a non-trivial problem in practice; i.e., the gap between process information and business processes. During the initial phase of the research, we conducted two exploratory case studies, an additional online survey, and a literature survey. Based on these activities as well as on practical insights we gathered in the automotive and clinical domains, we derived fundamental POIL requirements. Taking the latter into account, we then created the solution; i.e., the POIL framework. Finally, we validated POIL based on several prototypes, a validation case study in the automotive domain, an experiment, an online survey, and performance tests.

In detail, the contributions of this thesis are as follows:

- We identified requirements enabling POIL based on two exploratory case studies. Additionally, we presented results of an online survey with 219 participants supporting the case study findings.
- We introduced the POIL framework, a comprehensive approach enabling process-oriented and context-aware delivery of process information to process participants.
- We introduced the *semantic information network* (SIN), a directed, labeled and weighted graph that integrates process information, business processes, and their relationships. In particular, the SIN enables *information-* and *process-awareness* in the POIL framework.
- We introduced techniques and three fundamental algorithms for maintaining semantic networks (e.g., the SIN).
- We presented a *context model* (CM) for storing and handling context information in a meaningful machine- and user-interpretable form. The context framework enables *context-awareness* in the POIL framework. Based on the CM, it becomes possible to retrieve process information and business processes related to each other taking the process participants' work context into account.
- We introduced techniques and algorithms for identifying relevant process information. More specifically, we introduced two algorithms for determining the relevance of process information based on their link and rate popularity.
- We presented prototypes implementing the developed concepts and described results of ta validation case study in the automotive domain and an experiment in the agricultural domain. Additionally, we presented results of performance tests.

The POIL framework particularly focuses on knowledge-intensive business processes that involve large amounts of process information, user expertise, user interaction, creativity, and decision making [25, 96]. Basically, knowledge-intensive business processes are not or only partly automated (e.g., by process management technology). Examples of knowledge-intensive processes include the engineering of cars in the automotive domain [17] or the treatment of patients in integrated healthcare networks [20].

In practice, business processes and their tasks are often managed based on process management technology [36] and PAIS [21, 38]. Business processes may be characterized by hundreds or thousands of process tasks [39, 40], numerous process variants [90, 91], and large amounts of process information. Moreover, with the increasing adoption of PAIS, large process model repositories have emerged [301]. These and future developments may result in new requirements for POIL. Thus, we need to investigate how to effectively represent process variants in a SIN to avoid redundant representation of objects. For example, a particular process task that exists in two process variants should only be represented as a single SIN process object. Moreover, in recent years, collaborative business processes come into focus [302, 303]. Collaborative business processes constitute a special type of business processes that involve two or more enterprises. The complexity of collaborative business processes causes difficulties in process modeling and even more difficulties in representing them in a machine-interpretable form [304]. This may also lead to a refinement of existing POIL requirements.

In a follow-up research project, we will further implement components of the POIL framework to make POIL ready for an operative use at a large automotive manufacturer. Thereby, one key challenge will be to consider the evolution of a SIN over time. Besides, it will also be necessary to develop additional algorithms for determining the relevance of process information; i.e., self-learning algorithms taking SIN evolution into account. Furthermore, we will investigate the handling of process schemas and instances within SINs as well as of performance and scalability issues. Finally, it will be also subject of future research to extend the framework itself, for example, regarding its suitability to enable a more intuitive use of the SIN for process participants. For example, to provide further methods to query the SIN by the SIN Facade.

Note that the POIL framework has been developed in the automotive domain; i.e., in a domain which is characterized by large, distributed and heterogeneous data sources, and complex and distributed engineering processes. It will be an important task to investigate the suitability of the POIL framework for other domains (e.g., financial services, aerospace) as well as to prove the applicability for small- and medium-sized enterprises.

Bibliography

[1] B. Michelberger, B. Mutschler, and M. Reichert. *Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter.* In: Proc 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107–120, Hamburg, Germany, 2011.

[2] B. Michelberger, B. Mutschler, and M. Reichert. *On Handling Process Information: Results from Case Studies and a Survey.* In: Proc BPM'11 Workshops, 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333–344, Clermont-Ferrand, France, 2011.

[3] B. Michelberger, B. Mutschler, and M. Reichert. *Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes.* In: Proc 16th Int'l Enterprise Computing Conf (EDOC'12), IEEE Computer Society Press, pp. 21–30, Beijing, China, 2012.

[4] B. Michelberger, B. Mutschler, and M. Reichert. *A Context Framework for Process-oriented Information Logistics.* In: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 260–271, Vilnius, Lithuania, 2012.

[5] H. Girit, R. Eberhard, B. Michelberger, and B. Mutschler. *On the Precision of Search Engines: Results from a Controlled Experiment.* In: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 201–212, Vilnius, Lithuania, 2012.

[6] B. Michelberger, A. Reisch, B. Mutschler, J. Wurzer, M. Hipp, and M. Reichert. *iCare: Intelligent Medical Information Logistics.* In: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), ACM Press, pp. 396–399, Vienna, Austria, 2013.

[7] B. Michelberger, B. Mutschler, M. Hipp, and M. Reichert. *Determining the Link and Rate Popularity of Enterprise Process Information.* In: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS'13), LNCS 8185, pp. 112–129, Graz, Austria, 2013.

[8] B. Michelberger, R. Andris, H. Girit, and B. Mutschler. *A Literature Survey on Information Logistics.* In: Proc 16th Int'l Conf on Business Information Systems (BIS'13), LNBIP 157, pp. 138–150, Poznań, Poland, 2013.

[9] M. Hipp, B. Michelberger, B. Mutschler, and M. Reichert. *A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information.* In: Proc 28th Symposium on Applied Computing (SAC'13), ACM Press, pp. 1383–1390, Coimbra, Portugal, 2013.

[10] B. Michelberger, B. Mutschler, D. Binder, J. Meurer, and M. Hipp. *iGraph: Intelligent Enterprise Information Logistics.* In: Proc 10th Int'l Conf on Semantic Systems (SEMANTiCS'14), Posters & Demonstrations Track, CEUR Workshop Proc 1224, pp. 27–30, Leipzig, Germany, 2014.

[11] M. Hipp, A. Strauss, B. Michelberger, B. Mutschler, and M. Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* In: Proc BPM'14 Workshops, 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395–407, Haifa, Israel, 2014.

[12] M. Hipp, B. Mutschler, B. Michelberger, and M. Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* In: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE Computer Society Press, pp. 1–12, Marrakesh, Morocco, 2014.

[13] B. Michelberger, K. Ulmschneider, B. Glimm, B. Mutschler, and M. Reichert. *Maintaining Semantic Networks: Challenges and Algorithms.* In: Proc 16th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'14), ACM Press, pp. 365–374, Hanoi, Vietnam, 2014.

[14] K. Ulmschneider, B. Michelberger, B. Glimm, B. Mutschler, and M. Reichert. *On Maintaining Semantic Networks: Challenges, Algorithms, Use Cases.* In: Int'l J of Web Information Systems (IJWIS), 11(3), pp. 291–326, 2015.

[15] B. Michelberger, M. Hipp, and B. Mutschler. *Process-oriented Information Logistics: Requirements, Techniques, Application.* In: Advances in Intelligent Process-Aware Information Systems, 2015. Accepted for Publication.

[16] M.C. Burda and B. Dluhosch. *Cost Competition, Fragmentation, and Globalization.* In: Review of Int'l Economics, 10(3), pp. 424–441, 2002.

[17] D. Müller, J. Herbst, M. Hammori, and M. Reichert. *IT Support for Release Management Processes in the Automotive Industry.* In: Proc 4th Int'l Conf on Business Process Management (BPM'06), pp. 368–377, Vienna, Austria, 2006.

[18] G. Grambow, N. Mundbrod, V. Steller, M. Reichert, A. Schiffleitner, T. Bley, and C. Feick. *State-of-the-Art and Requirements for Collecting and Managing Sustainability Data Along Today's Supply Chains.* In: Proc 6th Int'l Conf on Life Cycle Management (LCM'13), pp. 548–552, Gothenburg, Sweden, 2013.

[19] G. Grambow, N. Mundbrod, V. Steller, and M. Reichert. *Challenges of Applying Adaptive Processes to Enable Variability in Sustainability Data Collection.* In: Proc 3rd Int'l Symposium on Data-Driven Process Discovery and Analysis (SIMPDA'13), pp. 74–88, Riva del Garda, Italy, 2013.

[20] R. Lenz and M. Reichert. *IT Support for Healthcare Processes – Premises, Challenges, Perspectives.* In: J of Data and Knowledge Engineering, 61(1), pp. 39–58, 2007.

[21] M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies.* Springer, 2012.

[22] M. Dumas, M. La Rosa, J. Mendling, and H.A. Reijers. *Fundamentals of Business Process Management.* Springer, 2013.

[23] L. Browning, A.S. Sætre, K.K. Stephens, and J. Sørnes. *Information and Communication Technologies in Action: Linking Theories and Narratives of Practice.* Routledge, 2008.

[24] B. Mutschler, M. Reichert, and J. Bumiller. *Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors and Implications.* In: J of IEEE Transactions on Systems, Man, and Cybernetics (SMC) – Part C: Applications and Reviews, 38(3), pp. 280–291, 2008.

[25] N. Mundbrod, J. Kolb, and M. Reichert. *Towards a System Support of Collaborative Knowledge Work.* In: Proc BPM'12 Workshops, 1st Int'l Workshop on Adaptive Case Management (ACM'12), pp. 31–42, Tallinn, Estonia, 2012.

[26] A. Edmunds and A. Morris. *The Problem of Information Overload in Business Organisations: A Review of the Literature.* In: Int'l J of Information Management, 20(1), pp. 17–28, 2000.

[27] A. Öhgren and K. Sandkuhl. *Information Overload in Industrial Enterprises – Results of an Empirical Investigation.* In: Proc 2nd European Conf on Information Management and Evaluation (ECIME'08), pp. 343–350, London, England, 2008.

[28] J. Gantz, A. Boyd, and S. Dowling. *Tackling Information Overload At the Source.* IDC White Papers, 2009.

[29] J. Short, R.E. Bohn, and C. Baru. *How Much Information? Report on Enterprise Server Information.* Report, University of California, United States, 2011.

[30] P. Bocij, D. Chaffey, A. Greasley, and S. Hickie. *Business Information Systems: Technology, Development and Management for the E-Business.* Prentice Hall, 2006.

[31] J. Rowley. *The Wisdom Hierarchy: Representations of the DIKW Hierarchy.* In: J of Information Science, 33(2), pp. 163–180, 2006.

[32] S. Sadiq. *Handbook of Data Quality: Research and Practice.* Springer, 2013.

[33] T.C. Redman. *Data Quality Management Past, Present, and Future: Towards a Management System for Data.* In: Handbook of Data Quality: Research and Practice, pp. 15–40, 2013.

[34] E. Simperl, I. Thurlow, P. Warren, F. Dengler, J. Davies, M. Grobelnik, D. Mladenić, J.M. Gomez-Perez, and C.R. Moreno. *Overcoming Information Overload in the Enterprise: The Active Approach.* In: IEEE Internet Computing, 14(6), pp. 39–46, 2010.

[35] F. Cabitza, M. Sarini, C. Simone, and M. Telaro. *When Once is Not Enough: The Role of Redundancy in a Hospital Ward Setting.* In: Proc 2005 Int'l ACM SIGGROUP Conf on Supporting Group Work (GROUP'05), pp. 158–167, Sanibel Island, United States, 2005.

[36] M. Weske. *Business Process Management: Concepts, Languages, Architectures.* Springer, 2007.

[37] M. Dumas, W.M.P. van der Aalst, and A.H. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software Through Process Technology.* John Wiley & Sons, 2005.

[38] B. Weber, M. Reichert, W. Wild, and S. Rinderle-Ma. *Providing Integrated Life Cycle Support in Process-Aware Information Systems.* In: Int'l J of Cooperative Information Systems (IJCIS), 18(1), pp. 115–165, 2009.

[39] M. Hipp, B. Mutschler, and M. Reichert. *Navigating in Process Model Collections: A new Approach Inspired by Google Earth.* In: Proc BPM'11 Workshops, 1st Int'l Workshop on Process Model Collections (PMC'11), pp. 87–98, Clermont-Ferrand, France, 2011.

[40] B. Weber, M. Reichert, J. Mendling, and H.A. Reijers. *Refactoring Large Process Model Repositories.* In: Computers in Industry, 62(5), pp. 467–486, 2011.

[41] J.J. van Wijk and W.A.A. Nuij. *Smooth and Efficient Zooming and Panning.* In: Proc 9th Annual IEEE Conf on Information Visualization (INFOVIS'03), pp. 15–22, Seattle, United States, 2003.

[42] S. Haseloff. *Context Awareness in Information Logistics.* PhD Thesis, Technical University of Berlin, Germany, 2005.

[43] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling.* John Wiley & Sons, 2013.

[44] J.M. Kolb. *Business Intelligence in Plain Language: A practical guide to Data Mining and Business Analytics.* Applied Data Labs, 2012.

[45] V.L. Sauter. *Decision Support Systems for Business Intelligence.* John Wiley & Sons, 2011.

[46] A. Rockley and C. Cooper. *Managing Enterprise Content: A Unified Content Strategy.* New Riders, 2012.

[47] J. Lechtenbörger. *Data Warehouse Schema Design.* PhD Thesis, University of Münster, Germany, 2001.

[48] T. Bucher and B. Dinter. *Process Orientation of Information Logistics – An Empirical Analysis to Assess Benefits, Design Factors, and Realization Approaches.* In: Proc 41st Hawaii Int'l Conf on System Sciences (HICSS'08), pp. 392–402, Hawaii, United States, 2008.

[49] H. Baars and H.G. Kemper. *Management Support with Structured and Unstructured Data – An Integrated Business Intelligence Framework.* In: J of Information Systems Management, 25(2), pp. 132–148, 2008.

[50] S. Rouhani, S. Asgari, and S.V. Mirhosseini. *Review Study: Business Intelligence Concepts and Approaches.* In: American J of Scientific Research, 50, pp. 62–75, 2012.

[51] V.S. Janakiraman and K. Sarukesi. *Decision Support Systems.* Prentice Hall, 2004.

[52] S.A. Cameron. *Enterprise Content Management: A Business and Technical Guide.* British Informatics Society, 2011.

[53] S. Rinderle, M. Reichert, and P. Dadam. *On Dealing with Structural Conflicts between Process Type and Instance Changes.* In: Proc 2nd Int'l Conf on Business Process Management (BPM'04), pp. 274–289, Potsdam, Germany, 2004.

[54] N. Gronau, C. Müller, and R. Korf. *KMDL – Capturing, Analysing and Improving Knowledge-Intensive Business Processes.* In: J of Universal Computer Science, 11(4), pp. 452–472, 2005.

[55] R.J. Wieringa, N.A.M. Maiden, N. Mead, and C. Rolland. *Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion.* In: J of Requirements Engineering, 11(1), pp. 102–107, 2006.

[56] R.J. Wieringa and J.M.G. Heerkens. *The Methodological Soundness of Requirements Engineering Papers: A Conceptual Framework and two Case Studies.* In: J of Requirements Engineering, 11(4), pp. 295–307, 2006.

[57] R.J. Wieringa. *Design Science As Nested Problem Solving.* In: Proc 4th Int'l Conf on Design Science Research in Information Systems and Technology (DESRIST'09), pp. 1–12, Philadelphia, United States, 2009.

[58] G. Stephanopoulos and C. Han. *Intelligent Systems in Process Engineering: A Review.* In: Computers & Chemical Engineering, 20(6-7), pp. 743–791, 1996.

[59] A.R. Hevner, S.T. March, J. Park, and S. Ram. *Design Science in Information Systems Research.* In: J of MIS Quarterly, 28(1), pp. 75–105, 2004.

[60] S.T. March and G.F. Smith. *Design and Natural Science Research on Information Technology.* In: J of Decision Support Systems, 15(4), pp. 251–266, 1995.

[61] A.R. Hevner and S. Chatterjee. *Design Research in Information Systems: Theory and Practice.* Springer, 2012.

[62] A.Y. Halevy, N. Ashish, D. Bitton, M. Carey, D. Draper, J. Pollock, A. Rosenthal, and V. Sikka. *Enterprise Information Integration: Successes, Challenges and Controversies.* In: Proc 2005 ACM SIGMOD Int'l Conf on Management of Data (SIGMOD'05), pp. 778–787, Baltimore, United States, 2005.

[63] M. Hipp, B. Mutschler, and M. Reichert. *On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements.* In: Proc 6th Int'l Conf on Availability, Reliability and Security (ARES'11), pp. 390–397, Vienna, Austria, 2011.

[64] Y. Park, S.C. Gates, W. Teiken, and P.C. Cheng. *An Experimental Study on the Measurement of Data Sensitivity.* In: Proc 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS'11), pp. 70–77, Salzburg, Austria, 2011.

[65] G.K. Tayi and D.P. Ballou. *Examining Data Quality.* In: Communications of the ACM, 41(2), pp. 54–57, 1998.

[66] T.C. Redman. *Data Quality: The Field Guide.* Digital Press, 2001.

[67] R.Y. Wang. *A Product Perspective on Total Data Quality Management.* In: J of Communications of the ACM, 41(2), pp. 58–65, 1998.

[68] R.L. Ackoff. *From Data to Wisdom.* In: J of Applied Systems Analysis, 16(1), pp. 3–9, 1989.

[69] M. Zeleny. *Management Support Systems: Towards Integrated Knowledge Management.* In: J of Human Systems Management, 7(1), pp. 59–70, 1987.

[70] P. Lake and R. Drake. *Information Systems Management in the Big Data Era.* Springer, 2014.

[71] K. North. *Wissensorientierte Unternehmensführung: Wertschöpfung durch Wissen.* Gabler, 2005.

[72] T.H. Davenport and L. Prusak. *Working Knowledge.* Harvard Business Review Press, 2000.

[73] D. Chaffey and S. Wood. *Business Information Management: Improving Performance Using Information Systems.* Prentice Hall, 2004.

[74] E.M. Awad and H.M. Ghaziri. *Knowledge Management.* Dorling Kindersley, 2004.

[75] R.K. Rainer and E. Turban. *Introduction to Information Systems: Supporting and Transforming Business.* John Wiley & Sons, 2009.

[76] T. Aven. *A Conceptual Framework for Linking Risk and the Elements of the Data-Information-Knowledge-Wisdom (DIKW) Hierarchy.* In: Reliability Engineering & System Safety, 111, pp. 30–36, 2013.

[77] U. Meissen, S. Pfennigschmidt, A. Voisard, and T. Wahnfried. *Context- and Situation-Awareness in Information Logistics.* In: Proc 1st Int'l Workshop on Database Technologies for Handling XML Information on the Web (DataX'04), pp. 335–344, Heraklion, Greece, 2004.

[78] A. Schmidt, M. Beigl, and H.W. Gellersen. *There is more to Context than Location.* In: Computers & Graphics, 23(6), pp. 893–901, 1999.

[79] J.R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W.A.S. Buxton. *Evolution of a Reactive Environment.* In: Proc SIGCHI Conf on Human Factors in Computing Systems (CHI'95), pp. 170–177, Denver, United States, 1995.

[80] S. Elrod, G. Hall, R. Costanza, M. Dixon, and J. Des Rivières. *Responsive Office Environments.* In: Communications of the ACM, 36(7), pp. 84–85, 1993.

[81] R. Hull, P. Neaves, and J. Bedford-Roberts. *Towards Situated Computing.* In: Proc 1st IEEE Int'l Symposium on Wearable Computers (ISWC'97), pp. 146–153, Cambridge, United States, 1997.

[82] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. *Augment-able Reality: Situated Communication Through Physical and Digital Spaces.* In: Proc 2nd IEEE Int'l Symposium on Wearable Computers (ISWC'98), pp. 68–75, Pittsburgh, United States, 1998.

[83] S. Fickas, G. Kortuem, and Z. Segall. *Software Organization for Dynamic and Adaptable Wearable Systems.* In: Proc 1st IEEE Int'l Symposium on Wearable Computers (ISWC'97), pp. 56–63, Cambridge, United States, 1997.

[84] J. Pascoe, N.S. Ryan, and D.R. Morse. *Human-Computer-Giraffe Interaction: HCI in the Field.* In: Proc 1st Workshop on Human Computer Interaction with Mobile Devices, Glasgow, Scotland, 1998.

[85] J. Indulska and P. Sutton. *Location Management in Pervasive Systems.* In: Proc Australasian Information Security Workshop Conf on ACSW Frontiers'03, 21, pp. 143–151, Adelaide, Australia, 2003.

[86] B.N. Schilit, N. Adams, and R. Want. *Context-Aware Computing Applications.* In: Proc 1st Int'l Workshop on Mobile Computing Systems and Applications (WMC-SA'94), pp. 85–90, Santa Cruz, United States, 1994.

[87] A.K. Dey. *Providing Architectural Support for Building Context-Aware Applications.* PhD Thesis, Georgia Institute of Technology, United States, 2000.

[88] A.K. Dey and G.D. Abowd. *Towards a Better Understanding of Context and Context-Awareness.* In: Proc 1st Int'l Symposium on Handheld and Ubiquitous Computing (HUC'99), pp. 304–307, Karlsruhe, Germany, 1999.

[89] J. McCarthy and P.J. Hayes. *Some Philosophical Problems from the Standpoint of Artificial Intelligence.* In: Machine Intelligence, 4, pp. 463–502, 1969.

[90] A. Hallerbach. *Management von Prozessvarianten.* PhD Thesis, University of Ulm, Germany, 2010.

[91] C. Ayora, V. Torres, B. Weber, M. Reichert, and V. Pelechano. *VIVACE: A Framework for the Systematic Evaluation of Variability Support in Process-Aware Information Systems.* In: Information and Software Technology, 57, pp. 248–276, 2015.

[92] W.M.P. van der Aalst and M. Pesic. *DecSerFlow: Towards a Truly Declarative Service Flow Language.* In: Proc 3rd Int'l Workshop on Web Services and Formal Methods (WS-FM'06), pp. 1–23, Vienna, Austria, 2006.

[93] W.M.P. van der Aalst. *Process-Aware Information Systems: Lessons to Be Learned from Process Mining.* In: J of IEEE Transactions on Petri Nets and Other Models of Concurrency II, pp. 1–26, 2009.

[94] T.H. Davenport, S. Jarvenpaa, and T. Beers. *Improving Knowledge Work Processes.* In: Sloan Management Review, 37(4), pp. 53–65, 1996.

[95] M.J. Eppler, P.M. Seifried, and A. Röpnack. *Improving Knowledge Intensive Processes Through an Enterprise Knowledge Medium.* In: Proc 1999 ACM SIGCPR Conf on Computer Personnel Research (SIGCPR'99), pp. 222–230, New Orleans, United States, 1999.

[96] N. Mundbrod and M. Reichert. *Process-Aware Task Management Support for Knowledge-Intensive Business Processes: Findings, Challenges, Requirements.* In: Proc EDOC'14 Workshops, 3rd Int'l Workshop on Adaptive Case Management and other Non-workflow Approaches to BPM (AdaptiveCM'14), pp. 116–125, Ulm, Germany, 2014.

[97] V. Künzle. *Object-Aware Process Management.* PhD Thesis, University of Ulm, Germany, 2013.

[98] V. Künzle and M. Reichert. *Towards Object-aware Process Management Systems: Issues, Challenges, Benefits.* In: Proc 10th Int'l Workshop on Business Process Modeling, Development, and Support (BPMDS'09), pp. 197–210, Amsterdam, The Netherlands, 2009.

[99] K. Heuwinkel and W. Deiters. *Information logistics, E-Healthcare and Trust.* In: Proc Int'l Conf e-Society (IADIS'03), 2, pp. 791–794, Lisbon, Portugal, 2003.

[100] J.H. Dornberg and H.W. Keller. *Schlanke Informationslogistik: Wie die Prinzipien des Lean Managements helfen können, den Umgang mit Informationen und Wissen zu verbessern.* In: UdZ – Unternehmen der Zukunft, FIR-Zeitschrift für Betriebsorganisation und Unternehmensentwicklung, 3, pp. 57–60, 2010.

[101] W. Deiters, T. Löffeler, and S. Pfennigschmidt. *The Information Logistics Approach Toward User Demand-driven Information Supply.* In: Proc Conf on Cross-Media Service Delivery (CMSD'03), pp. 37–48, Santorini, Greece, 2003.

[102] J.P. Womack and D.T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* Free Press, 2003.

[103] B. Dinter and R. Winter. *Information Logistics Strategy – Analysis of Current Practices and Proposal of a Framework.* In: Proc 42nd Hawaii Int'l Conf on System Sciences (HICSS'09), pp. 1–10, Hawaii, United States, 2009.

[104] R. Winter. *Enterprise-wide Information Logistics: Conceptual Foundations, Technology Enablers, and Management Challenges.* In: Proc 30th Int'l Conf on Information Technology Interfaces (ITI'08), pp. 41–50, Dubrovnik, Croatia, 2008.

[105] C. Okoli and K. Schabram. *A Guide to Conducting a Systematic Literature Review of Information Systems Research.* In: Sprouts: Working Papers on Information Systems, 10(26), pp. 1–49, 2010.

[106] D.M. Haftor, M. Kajtazi, and A. Mirijamdotter. *A Review of Information Logistics Research Publications.* In: Proc 3rd Workshop on Information Logistics and Knowledge Supply (ILOG'10), pp. 244–255, Berlin, Germany, 2010.

[107] B. Dinter. *Success Factors for Information Logistics Strategy – An Empirical Investigation.* In: J of Decision Support Systems, 54(3), pp. 1207–1218, 2012.

[108] S. Klein. *Information Logistics.* In: J of the Electronic Markets, 3(3), pp. 11–12, 1993.

[109] G. Lahrmann. *Strategic Positioning of Information Logistics Service Providers: Guidelines for Selecting Appropriate Organizational Models.* In: Proc 43rd Hawaii Int'l Conf on System Sciences (HICSS'10), pp. 1–10, Hawaii, United States, 2010.

[110] G. Lahrmann and F. Stroh. *Towards a Classification of Information Logistics Scenarios – An Exploratory Analysis.* In: Proc 42nd Hawaii Int'l Conf on System Sciences (HICSS'09), pp. 1–10, Hawaii, United States, 2009.

[111] R. Winter, S. Bischoff, and F. Wortmann. *Revolution or Evolution? Reflections on in-memory Appliances from an Enterprise Information Logistics Perspective.* In: Proc Innovative Unternehmensanwendungen mit In-Memory Data Management (IMDM'11), pp. 23–34, Mainz, Germany, 2011.

[112] T. Bucher and J.M. Teich. *Design and Implementation of a Computerized Physician Order Entry System at Brigham and Women's Hospital – A Case Study on Process Orientation of Information Logistics*. Working Paper, University of St. Gallen, Switzerland, 2008.

[113] K. Sandkuhl. *Information Logistics in Networked Organizations: Selected Concepts and Applications*. In: Proc 11th Int'l Conf on Enterprise Information Systems (ICEIS'09), pp. 43–54, Milan, Italy, 2009.

[114] K. Heuwinkel, W. Deiters, T. Königmann, and T. Löffeler. *Information Logistics and Wearable Computing*. In: Proc 23rd Int'l Conf on Distributed Computing Systems Workshops (ICDCS'03), pp. 283–289, Los Alamitos, United States, 2003.

[115] S. Jaksch, S. Pfennigschmidt, K. Sandkuhl, and C. Thiel. *Information Logistic Applications for Information-on-Demand Scenarios: Concepts and Experiences from WIND Project*. In: Proc 29th EUROMICRO Conf (EUROMICRO'03), pp. 141–147, Belek, Turkey, 2003.

[116] M. Böhringer and M. Gaedke. *Ubiquitous Microblogging: A Flow-Based Front-End for Information Logistics*. In: Proc 3rd Workshop on Information Logistics and Knowledge Supply (ILOG'10), pp. 158–167, Berlin, Germany, 2010.

[117] K. Sandkuhl, A. Smirnov, and N. Shilov. *Information Logistics in Engineering Change Management: Integrating Demand Patterns and Recommendation Systems*. In: Proc 4th Workshop on Information Logistics and Knowledge Supply for Viable Enterprises (ILOG'11), pp. 14–25, Riga, Latvia, 2012.

[118] K. Sandkuhl, U. Borchardt, B. Lantow, D. Stamer, and M. Wißotzki. *Towards Adaptive Business Models for Intelligent Information Logistics in Transportation*. In: Proc 11th Int'l Conf on Perspectives in Business Informatics Research (BIR'12), pp. 18–30, Nizhny Novgorod, Russia, 2012.

[119] B.H.P.J. Vermeer. *Information Logistics: A Data Integration Method for Solving Data Quality Problems with Article Information in Large Interorganizational Networks*. In: Proc 4th Int'l Conf on Information Quality (ICIQ'99), pp. 185–208, Cambridge, United States, 1999.

[120] T. Levashova, M. Lundqvist, K. Sandkuhl, and A. Smirnov. *Context-based Modelling of Information Demand: Approaches from Information Logistics and Decision Support*. In: Proc 14th European Conf on Information Systems (ECIS'06), pp. 1511–1522, Göteborg, Sweden, 2006.

[121] M. Lundqvist, K. Sandkuhl, T. Levashova, and A. Smirnov. *Context-Driven Information Demand Analysis in Information Logistics and Decision Support Practices*. In: Proc 1st Int'l Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O'05), pp. 124–127, Pittsburgh, United States, 2005.

[122] U. Meissen, S. Pfennigschmidt, T. Wahnfried, and K. Sandkuhl. *Situation-based Message Rating in Information Logistics and its Applicability in Collaboration Scenarios.* In: Proc 30th EUROMICRO Conf (EUROMICRO'04), pp. 484–491, Rennes, France, 2004.

[123] M. Apelkrans and A. Håkansson. *Enterprise Systems Configuration as an Information Logistics Process – A Study.* In: Proc 9th Int'l Conf on Enterprise Information Systems (ICEIS'07), pp. 212–220, Madeira, Portugal, 2007.

[124] M. Apelkrans and A. Håkansson. *Information Coordination Using Meta-agents in Information Logistics Processes.* In: Proc 12th Int'l Conf on Knowledge-Based Intelligent Information and Engineering Systems (KES'08), pp. 788–798, Zagreb, Croatia, 2008.

[125] M. Apelkrans. *Multi Project Control As An Information Logistics Process.* In: Proc Australian Institute of Project Management Conf (AIPM'03), Alice Springs, Australia, 2003.

[126] M. Apelkrans and A. Håkansson. *Visual Knowledge Modeling of an Information Logistics Process – A Case Study.* In: Proc 2nd Int'l Conf on Intellectual Capital, Knowledge Management and Organisational Learning (ICICKM'05), pp. 61–70, Dubai, United Arab Emirates, 2005.

[127] H. Knublauch and T. Rose. *Application Scenarios of Agent-Based Information Logistics in Clinical and Engineering Domains.* In: Proc 14th European Conf on Artificial Intelligence (ECAI'00), pp. 85–88, Berlin, Germany, 2000.

[128] I.J. Timm, P.O. Woelk, P. Knirsch, H.K. Tönshoff, and O. Herzog. *Flexible Mass Customisation: Managing its Information Logistics Using Adaptive Co-operative Multiagent Systems.* In: Proc 6th Int'l Symposium on Logistics (ISL'01), pp. 227–232, Salzburg, Austria, 2001.

[129] S. Winkler, R. Zimmermann, and F. Bodendorf. *An Agent-Based Information Logistics Architecture for Process Management.* In: Proc Int'l Conf on Computational Intelligence for Modelling, Control and Automation (CIMCA'05) and Int'l Conf on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'05), pp. 745–751, Vienna, Austria, 2005.

[130] F. Bodendorf, S. Winkler, R. Zimmermann, and B. Vögele. *Agent-Based Information Logistics in Planning Processes.* In: Proc 2nd Int'l Conf on Systems (ICONS'07), pp. 54–59, Sainte-Luce, Martinique, 2007.

[131] O. Candell, R. Karim, and P. Söderholm. *eMaintenance-Information Logistics for Maintenance Support.* In: J of Robotics and Computer-Integrated Manufacturing, 25(6), pp. 937–944, 2009.

[132] O. Candell, R. Karim, P. Söderholm, and U. Kumar. *Service-oriented Information Logistics as Support to Intelligent Transport Services.* In: Proc 16th World Congress and Exhibition on Intelligent Transport Systems (ITS'09), pp. 21–25, Stockholm, Sweden, 2010.

[133] D.M. Haftor, M. Kajtazi, and A. Mirijamdotter. *Research and Practice Agenda of Industrial e-Maintenance: Information Logistics as a Driver for Development.* In: Proc 1st Int'l Workshop and Congress on eMaintenance, pp. 56–61, Luleå, Sweden, 2010.

[134] R. Karim, O. Candell, and P. Sölderholm. *E-Maintenance and Information Logistics: Aspects of Content Format.* In: J of Quality in Maintenance Engineering, 15(3), pp. 308–324, 2009.

[135] A.C.V. Vieira and A.J.M. Cardoso. *The Role of Information Logistics and Data Warehousing in Educational Facilities Asset Management.* In: Int'l J of Systems Assurance Engineering and Management, 1(3), pp. 229–238, 2010.

[136] B.D. Czejdo and M. Baszun. *Information Logistics for Incomplete Knowledge Processing.* In: Proc Organizational, Business, and Technological Aspects of the Knowledge Society Communications in Computer and Information Science (WSKS'10), pp. 295–302, Corfu, Greece, 2010.

[137] B.D. Czejdo, M. Baszun, and T. Cummings. *Knowledge Workers' Advisor Based On Information Logistics Models.* In: Proc 4th Int'l Conf on Knowledge Generation, Communication and Management (KGCM'10), Orlando, United States, 2010.

[138] P. Rudzajs and M. Kirikova. *Multimode Information Logistics for Conceptual Correspondence Monitoring.* In: Proc 11th Int'l Conf on Perspectives in Business Informatics Research (BIR'12), pp. 31–42, Nizhny Novgorod, Russia, 2012.

[139] J. Willems. *From Having to Using: Information Logistics Experience Centre is Born.* Working Paper, Nyenrode Business University, The Netherlands, 2008.

[140] A. Willems, J. Willems, and A. Hajdasinski. *Information Logistics Research Report – Frameworks in the Healthcare Industry.* Working Paper, Nyenrode Business University, The Netherlands, 2009.

[141] M. Lendholt and M. Hammitzsch. *Generic Information Logistics for Early Warning Systems.* In: Proc 8th Int'l Conf on Information Systems for Crisis Response and Management (ISCRAM'11), pp. 1–10, Lisbon, Portugal, 2011.

[142] M. Lendholt and M. Hammitzsch. *Towards an Integrated Information Logistics for Multi Hazard Early Warning Systems.* In: The Open Environmental Engineering J, 5, pp. 27–42, 2012.

[143] L. Soibelman and C. Caldas. *Information Logistics Approach for Construction Inter-Organizational Information Systems.* In: Proc Int'l Conf on Construction Information Technology (CIT'00), Reykjavik, Iceland, 2000.

[144] L. Soibelman and C. Caldas. *Information Logistics for Construction Design Team Collaboration.* In: Proc 8th Int'l Conf on Computing in Civil and Building Engineering (ICCCBE'00), pp. 341–348, Stanford, United States, 2000.

[145] R.J. Scherer. *Information Logistics for Supporting the Collaborative Design Process.* In: Collaborative Design and Learning: Competence Building for Innovation, pp. 198–222, 2004.

[146] S. Nuntasunti. *The Effects of Visual-based Information Logistics in Construction.* PhD Thesis, North Carolina State University, United States, 2003.

[147] M. Vegetti, S. Gonnet, G. Henning, and H. Leone. *Information Logistics for Supply Chain Management within Process Industry Environments.* In: J of the Computer Aided Chemical Engineering, 20, pp. 1231–1236, 2005.

[148] M. Vegetti, S. Gonnet, G. Henning, and H. Leone. *Towards a Supply Chain Ontology of Information Logistics within Process Industry Environments.* In: Proc 2nd Mercosur Congress on Chemical Engineering and 4th Mercosur Congress on Process Systems Engineering, pp. 15–25, Rio de Janeiro, Brasil, 2005.

[149] J. Timlon and S. Harryson. *Realizing a New Supply Chain Strategy – Re-Conceptualizing Actors: Meaning Structures of Information Logistics Activities.* In: Proc 22nd Annual IMP Conf, Milan, Italy, 2006.

[150] P. Söderholm and P. Norrbin. *Information Logistics for Continuous Dependability Improvement.* In: J of Quality in Maintenance Engineering, 20(3), pp. 249–261, 2014.

[151] K. Sandkuhl. *Pattern Use in Knowledge Architectures: An Example from Information Logistics.* In: Proc 12th Int'l Conf on Perspectives in Business Informatics Research (BIR'13), pp. 91–103, Warsaw, Poland, 2013.

[152] H. Koç, B. Lantow, and K. Sandkuhl. *Ontology Development for Intelligent Information Logistics in Transportation.* In: Proc 2nd Int'l Workshop on Ontologies and Information Systems (WOIS'14), pp. 2–17, Lund, Sweden, 2014.

[153] P. Rudzajs and M. Kirikova. *Conceptual Correspondence Monitoring: Multimode Information Logistics Approach.* In: J on Complex Systems Informatics and Modeling Quarterly (CSIMQ), 1, pp. 57–73, 2014.

[154] S. Meister, W. Deiters, S. Schafer, and V. Stahlmann. *Pattern Use in Knowledge Architectures: An Example from Information Logistics.* In: J on Complex Systems Informatics and Modeling Quarterly (CSIMQ), 1, pp. 42–56, 2014.

[155] K. Ignáczová. *Opportunities to increase of Establishment's Competitiveness applying Resources of Information Logistics in the concrete Company's Environment*. In: Int'l Scientific J about Logistics – Acta Logistica, 1(2), pp. 1–5, 2014.

[156] M. Straka, P. Bindzár, and M. Balog. *Information Logistics as means of Information Services for Freight Transport Needs*. In: Int'l Scientific J about Logistics – Acta Logistica, 1(2), pp. 19–26, 2014.

[157] A. Gaidukovs and M. Kirikova. *The Time Dimension in Information Logistics*. In: Proc 6th Int'l Workshop on Information Logistics, Knowledge Supply and Ontologies in Information Systems (ILOG'13), pp. 35–43, Warsaw, Poland, 2013.

[158] D. Stamer, A. Ponomarev, K. Sandkuhl, N. Shilov, and A. Smirnov. *Collaborative Recommendation System for Improved Information Logistics: Adaption of Information Demand Pattern in E-Mail Communication*. In: Proc 7th Int'l Workshop on Information Logistics and Knowledge Supply (ILOG'14), pp. 35–48, Lund, Sweden, 2014.

[159] T. Evgeniou. *Information Integration and Information Strategies for Adaptive Enterprises*. In: European Management J, 20(5), pp. 486–494, 2002.

[160] R.K. Yin. *Case Study Research: Design and Methods*. Sage Publications, 2009.

[161] B. Kitchenham, L. Pickard, and S.L. Pfleeger. *Case Studies for Method and Tool Evaluation*. In: J of IEEE Software, 12(4), pp. 52–62, 1995.

[162] LinkedIn. http://www.linkedin.com, December 2014.

[163] Xing. http://www.xing.com, December 2014.

[164] T. Allweyer. *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. BoD – Book on Demand, 2010.

[165] A.W. Scheer. *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. Springer, 2002.

[166] R.Y. Wang and D.M. Strong. *Beyond Accuracy: What Data Quality Means to Data Consumers*. In: J of Management Information Systems, 12(4), pp. 5–34, 1996.

[167] M. Hipp, B. Mutschler, and M. Reichert. *Navigating in Complex Business Processes*. In: Proc 23rd Int'l Conf on Database and Expert Systems Applications (DEXA'12), pp. 466–480, Vienna, Austria, 2012.

[168] R. Pryss, N. Mundbrod, D. Langer, and M. Reichert. *Supporting Medical Ward Rounds through Mobile Task and Process Management*. In: J of Information Systems and e-Business Management, 2014.

[169] R. Elmasri and S. Navathe. *Fundamentals of Database Systems.* Addison-Wesley, 2010.

[170] J.E. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann Publishers, 1991.

[171] J. Wurzer and B. Mutschler. *Bringing Innovative Semantic Technology to Practice: The iQser Approach and its Use Cases.* In: Proc 4th Int'l Workshop on Applications of Semantic Technologies (AST'09), pp. 3026–3040, Lübeck, Germany, 2009.

[172] B. Weber, S. Rinderle, W. Wild, and M. Reichert. *CCBR-Driven Business Process Evolution.* In: Proc 6th Int'l Conf on Case-Based Reasoning (ICCBR'05), pp. 610–624, Chicago, United States, 2005.

[173] L. Thom, M. Reichert, and C. Iochpe. *Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence.* In: Int'l J of Business Process Integration and Management (IJBPIM), 4(2), pp. 93–110, 2009.

[174] N.V. Findler. *Associative Networks: The Representation and Use of Knowledge of Computers.* Academic Press, 1979.

[175] J. Park and S. Hunting. *XML Topic Maps: Creating and Using Topic Maps for the Web.* Addison-Wesley, 2002.

[176] K. Reichenberger. *Kompendium semantische Netze: Konzepte, Technologie, Modellierung.* Springer, 2010.

[177] L.W. Lacy. *OWL: Representing Information Using the Web Ontology Language.* Trafford Publishing, 2005.

[178] H. Stuckenschmidt. *Ontologien: Konzepte, Technologien und Anwendungen.* Springer, 2011.

[179] K.D. Schewe and B. Thalheim. *About Semantics.* In: Proc 4th Int'l Conf on Semantics in Data and Knowledge Bases (SDKB'10), pp. 1–22, Bordeaux, France, 2010.

[180] R. Diestel. *Graph Theory.* Springer, 2010.

[181] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. *Combating Web Spam with TrustRank.* In: Proc 13th Int'l Conf on Very Large Data Bases (VLDB'04), pp. 576–587, Toronto, Canada, 2004.

[182] J. Wurzer. *New Approach for Semantic Web by Automatic Semantics.* In: Proc 2nd European Semantic Technology Conf (ESCT'08), Vienna, Austria, 2008.

[183] A. Hotho, A. Nürnberger, and G. Paaß. *A Brief Survey of Text Mining.* In: J for Computational Linguistics and Language Technology, 20(1), pp. 19–62, 2005.

[184] R. Kimball and J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* John Wiley & Sons, 2004.

[185] S. Oertelt and K. Ulmschneider. *Prozessintegrierter Einsatz virtueller Methoden im strategischen Technologie- und Innovationsmanagement.* In: Proc 15th KnowTech – Wissensmanagement und Social Media, pp. 485–498, Hanau, Germany, 2013.

[186] A. Borodin, G.O. Roberts, J.S. Rosenthal, and P. Tsaparas. *Link Analysis Ranking: Algorithms, Theory, and Experiments.* In: J of ACM Transactions on Internet Technology (TOIT), 5(1), pp. 231–297, 2005.

[187] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report, Stanford University, United States, 1999.

[188] J.M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A.S. Tomkins. *The Web as a Graph: Measurements, Models, and Methods.* In: Proc 5th Annual Int'l Conf on Computing and Combinatorics (COCOON'99), pp. 1–17, Tokyo, Japan, 1999.

[189] W. Xing and A. Ghorbani. *Weighted PageRank Algorithm.* In: Proc 2nd Annual Conf on Communication Networks and Services Research (CNSR'04), pp. 305–314, Fredericton, Canada, 2004.

[190] R. Lempel and S. Moran. *The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect.* In: Proc 9th Int'l World Wide Web Conf (WWW'00), pp. 387–401, Amsterdam, The Netherlands, 2000.

[191] T.H. Haveliwala. *Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search.* In: J of IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 784–796, 2003.

[192] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[193] S. Vassilvitskii and E. Brill. *Using Web-Graph Distance for Relevance Feedback in Web Search.* In: Proc 29th Int'l ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'06), pp. 147–153, Seattle, United States, 2006.

[194] D. Lowd, O. Godde, M. McLaughlin, S. Nong, Y. Wang, and J.L. Herlocker. *Challenges and Solutions for Synthesis of Knowledge Regarding Collaborative Filtering Algorithms.* Technical Report, Oregon State University, United States, 2004.

[195] J. Bian, Y. Liu, E. Agichtein, and H. Zha. *A few Bad Votes too Many?: Towards Robust Ranking in Social Media.* In: Proc 4th Int'l Workshop on Adversarial Information Retrieval on the Web (AIRWeb'08), pp. 53–60, Bejing, China, 2008.

[196] J. Wurzer. *iQser GIN Plattform.* In: Künstliche Intelligenz (KI), 24(1), pp. 83–86, 2010.

[197] T. Čapek. *Semantic Network Integrity Maintenance via Heuristic Semi-Automatic Tests.* In: Proc 3rd Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN'09), pp. 63–67, Karlova Studánka, Czech Republic, 2009.

[198] Y. Gargouri, B. Lefebvre, and J.G. Meunier. *Ontology Maintenance using Textual Analysis.* In: Proc 7th World Multi-Conf on Systemics, Cybernetics and Informatics (SCI'03), pp. 248–253, Orlando, United States, 2003.

[199] D. Dinh, J.C. Dos Reis, C. Pruskia, M. Da Silveira, and C. Reynaud-Delaître. *Identifying Relevant Concept Attributes to Support Mapping Maintenance under Ontology Evolution.* In: J of Web Semantics: Science, Services and Agents on the World Wide Web, 2014.

[200] G.H. Forman and J. Zahorjan. *The Challenges of Mobile Computing.* In: IEEE Computer, 27(4), pp. 38–47, 1994.

[201] J. Krumm. *Ubiquitous Computing Fundamentals.* Chapman and Hall/CRC, 2009.

[202] W.J. Kaltz, J. Ziegler, and S. Lohmannn. *Context-aware Web Engineering: Modeling and Applications.* In: Revue d'Intelligence Artificielle (RIA), Special Issue on Applying Context-Management, 19(3), pp. 439–458, 2005.

[203] G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, and P. Steggles. *Towards a Better Understanding of Context and Context-Awareness.* In: Proc 1st Int'l Symposium on Handheld and Ubiquitous Computing (HUC'99), pp. 304–307, 1999.

[204] P. Moore and B. Hu. *A Context Framework for Entity Identification for the Personalisation of Learning in Pedagogic Systems.* In: Proc 10th Int'l Conf on Computer Supported Cooperative Work in Design (CSCWD'06), pp. 1–6, Nanjing, China, 2006.

[205] R. Pryss, S. Musiol, and M. Reichert. *Collaboration Support Through Mobile Processes and Entailment Constraints.* In: Proc 9th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'13), pp. 178–187, Austin, United States, 2013.

[206] R. Pryss, S. Musiol, and M. Reichert. *Integrating Mobile Tasks with Business Processes: A Self-Healing Approach.* In: Handbook of Research on Architectural Trends in Service-Driven Computing, pp. 103–135, 2014.

[207] G. Grambow, R. Oberhauser, and M. Reichert. *Contextual Generation of Declarative Workflows and their Application to Software Engineering Processes.* In: Int'l J on Advances in Intelligent Systems, 4(3&4), pp. 158–179, 2011.

[208] G. Grambow, R. Oberhauser, and M. Reichert. *Contextual Injection of Quality Measures into Software Engineering Processes.* In: Int'l J on Advances in Software, 4(1&2), pp. 76–99, 2011.

[209] G. Grambow, R. Oberhauser, and M. Reichert. *Knowledge Provisioning: A Context-sensitive Process-oriented Approach Applied to Software Engineering Environments.* In: Proc 7th Int'l Conf on Software Paradigm Trends (ICSOFT'12), pp. 506–515, Rome, Italy, 2012.

[210] H.L. Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems.* PhD Thesis, University of Maryland, United States, 2004.

[211] M. Fowler. *Patterns of Enterprise Application Architecture.* Addison-Wesley, 2002.

[212] M. Baldauf, S. Dustdar, and F. Rosenberg. *A Survey on Context-aware systems.* In: Int'l J of Ad Hoc and Ubiquitous Computing, 2(4), pp. 263–277, 2007.

[213] C. Traynor and M.G. Williams. *Why are Geographic Information Systems hard to Use?* In: Proc SIGCHI Conf on Human Factors in Computing Systems (CHI'95), pp. 288–289, Denver, United States, 1995.

[214] L. Sun, P. Dai, L. Tao, and G. Xu. *A Generic Context Information System for Intelligent Vision Applications.* In: Proc 8th Int'l Conf on Computational Science (ICCS'08), pp. 759–769, Kraków, Poland, 2008.

[215] R. Klemke. *Modelling Context in Information Brokering Processes.* PhD Thesis, RWTH Aachen University, Germany, 2002.

[216] M. Ge and M. Helfert. *Cost and Value Management for Data Quality.* In: Handbook of Data Quality: Research and Practice, pp. 75–92, 2013.

[217] R. Jung. *Architekturen zur Datenintegration.* Deutscher Universitäts-Verlag/GWV Fachverlage, 2006.

[218] C. Fürber and M. Hepp. *Using Semantic Web Technologies for Data Quality Management.* In: Handbook of Data Quality: Research and Practice, pp. 141–161, 2013.

[219] X. Wu, L. Zhang, and Y. Yu. *Exploring Social Annotations for the Semantic Web.* In: Proc 15th Int'l World Wide Web Conf (WWW'06), pp. 417–426, Edinburgh, Scotland, 2006.

[220] D. Salber, A.K. Dey, and G.D. Abowd. *The Context Toolkit: Aiding the Development of Context-Enabled Applications.* In: Proc SIGCHI Conf on Human Factors in Computing Systems (CHI'99), pp. 434–441, Pittsburgh, United States, 1999.

[221] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger. *Context-Awareness on Mobile Devices – the Hydrogen Approach.* In: Proc 36th Annual Hawaii Int'l Conf on System Sciences (HICSS'03), pp. 292–301, Hawaii, United States, 2003.

[222] T. Gross and R. Klemke. *Context Modelling for Information Retrieval – Requirements and Approaches.* In: IADIS Int'l J on WWW/Internet, 1(1), pp. 29–42, 2003.

[223] B. Thönssen and D. Wolff. *A broader view on Context Models to support Business Process Agility.* In: Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications, pp. 337–358, IGI Global, 2010.

[224] J. Ziegler, S. Lohmannn, and W.J. Kaltz. *Kontextmodellierung für adaptive webbasierte Systeme.* In: Mensch und Computer 2005: Kunst und Wissenschaft – Grenzüberschreitung der interaktiven Art, pp. 181–189, 2005.

[225] T. Strang and C. Linnhoff-Popien. *A Context Modeling Survey.* In: Proc 1st Int'l Workshop on Advanced Context Modelling, Reasoning and Management at UbiComp'04, pp. 33–40, Nottingham, England, 2004.

[226] Y. Wand and R.Y. Wang. *Anchoring Data Quality Dimensions in Ontological Foundations.* In: Communications of the ACM, 39(11), pp. 86–95, 1996.

[227] L.L. Pipino, Y.W. Lee, and R.Y. Wang. *Data Quality Assessment.* In: Communications of the ACM, 45(4), pp. 211–218, 2002.

[228] F. Naumann, U. Leser, and J.C. Freytag. *Quality-driven Integration of Heterogeneous Information Systems.* In: Proc 25th Int'l Conf on Very Large Data Bases (VLDB'99), pp. 447–458, Edinburgh, Scotland, 1999.

[229] J. Bailey, F. Bry, T. Furche, and S. Schaffert. *Web and Semantic Web Query Languages: A Survey.* In: Proc 1st Int'l Conf on Reasoning Web (RW'05), pp. 35–133, Msida, Malta, 2005.

[230] W3C XPath. http://www.w3schools.com/xpath, December 2014.

[231] W3C XSLT. http://www.w3schools.com/xsl, December 2014.

[232] Ontopia Tolog. http://www.ontopia.net/topicmaps/materials/tolog.html, December 2014.

[233] W3C SPARQL. http://www.w3.org/TR/rdf-sparql-query, December 2014.

[234] K. Tzompanaki and M. Doerr. *A New Framework for Querying Semantic Networks.* Technical Report, Foundation for Research and Technology – Hellas (FORTH), Greece, 2012.

[235] R. Lenz, M. Peleg, and M. Reichert. *Healthcare Process Support: Achievements, Challenges, Current Research.* In: Int'l J of Knowledge-Based Organizations (IJKBO), 2(4), pp. 1–12, 2012.

[236] H. Weber, M. Stöckli, M. Nübling, and W.A. Langewitz. *Communication during Ward Rounds in Internal Medicine: An Analysis of Patient-Nurse-Physician Interactions using RIAS.* In: Proc European Association for Communication in Healthcare (EACH'06) Conf, 67(3), pp. 343–348, Basel, Switzerland, 2007.

[237] Onmeda. http://www.onmeda.de, November 2014.

[238] R. Lenz and M. Reichert. *IT Support for Healthcare Processes.* In: Proc 3rd Int'l Conf on Business Process Management (BPM'05), pp. 354–363, Nancy, France, 2005.

[239] J.S. Burgers, R. Grol, N.S. Klazinga, M. Mäkelä, and J. Zaat. *Towards Evidence-based Clinical Practice: An International Survey of 18 Clinical Guideline Programs.* In: J for Quality in Health Care 2003, 15(1), pp. 31–45, 2003.

[240] B. Fervers, J.S. Burgers, M.C. Haugh, J. Latreille, N. Mlika-Cabanne, L. Paquet, M. Coulombe, M. Poirier, and B. Burnand. *Adaptation of Clinical Guidelines: Literature Review and Proposition for a Framework and Procedure.* In: J for Quality in Health Care 2003, 18(3), pp. 167–176, 2006.

[241] L. Ohno-Machado, J.H. Gennari, S.N. Murphy, N.L. Jain, S.W. Tu, D.E. Oliver, E. Pattison-Gordon, R.A. Greenes, E.H. Shortliffe, and G.O. Barnett. *The Guideline Interchange Format: A Model for Representing Guidelines.* In: J of the American Medical Informatics Association, 5(4), pp. 357–372, 1998.

[242] J. Fox, N. Johns, C. Lyons, A. Rahmanzadeh, R. Thomson, and P. Wilson. *PROforma: A General Technology for Clinical Decision Support Systems.* In: Computer Methods and Programs in Biomedicine, 54(1-2), pp. 59–67, 1997.

[243] E.H. Shortliffe and J.J. Cimino. *Biomedical Informatics: Computer Applications in Health Care and Biomedicin.* Springer, 2006.

[244] GNUmed. http://www.gnumed.org, June 2013.

[245] GNUHealth. http://health.gnu.org, June 2013.

[246] OpenEMR. http://www.open-emr.org, January 2015.

[247] FreeMED. http://www.freemedsoftware.org, June 2013.

[248] J. Tiedeken, M. Reichert, and J. Herbst. *On the Integration of Electrical/Electronic Product Data in the Automotive Domain.* In: Datenbank Spektrum, 13(3), pp. 189–199, 2013.

[249] Signavio Process Editor. http://www.signavio.com, November 2014.

[250] Activiti BPM Platform. http://www.activiti.org, November 2014.

[251] Apache Solr. http://lucene.apache.org/solr, November 2014.

[252] Elasticsearch. http://www.elasticsearch.com, November 2014.

[253] OpenSearchServer. http://www.opensearchserver.com, November 2014.

[254] Datameer. http://www.datameer.com, November 2014.

[255] Lumify. http://lumify.io, November 2014.

[256] I. Herman, G. Melancon, and M.S. Marshall. *Graph Visualization and Navigation in Information Visualization: A Survey.* In: J of IEEE Transactions on Visualization and Computer Graphics, 6(1), pp. 24–43, 2000.

[257] D. Binder. *Konzeption und Implementierung einer Anwendung zur semantischen Analyse wissenschaftlicher Publikationen.* Bachelor Thesis, University of Applied Sciences Ravensburg-Weingarten, Germany, 2013.

[258] SpringerLink. http://link.springer.com, November 2014.

[259] DBLP Computer Science Bibliography. http://www.dblp.org, November 2014.

[260] Google Scholar. http://scholar.google.de, November 2014.

[261] Microsoft Academic Search. http://academic.research.microsoft.com, November 2014.

[262] CiteSeerX. http://citeseerx.ist.psu.edu/index, November 2014.

[263] ScienceDirect. http://www.sciencedirect.com, November 2014.

[264] ArnetMiner. http://www.arnetminer.org, November 2014.

[265] R. Bobrik, M. Reichert, and T. Bauer. *View-Based Process Visualization.* In: Proc 5th Int'l Conf on Business Process Management (BPM'07), pp. 88–95, Brisbane, Australia, 2007.

[266] M. Reichert, J. Kolb, R. Bobrik, and T. Bauer. *Enabling Personalized Visualization of Large Business Processes through Parameterizable Views.* In: Proc 27th Symposium on Applied Computing (SAC'12), pp. 1653–1660, Trento, Italy, 2012.

[267] J. Kolb and M. Reichert. *Data Flow Abstractions and Adaptations through Updatable Process Views.* In: Proc 28th Symposium on Applied Computing (SAC'13), pp. 1447–1453, Coimbra, Portugal, 2013.

[268] M. Hipp. *Navigating in Complex Process Model Collections.* PhD Thesis, University of Ulm, Germany, 2015.

[269] S. Smirnov. *Business Process Model Abstraction.* PhD Thesis, University of Potsdam, Germany, 2011.

[270] J. Kolb, M. Reichert, and B. Weber. *Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes.* In: Proc 4th Int'l Conf on Subject-Oriented Business Process Management (S-BPM ONE'12), pp. 237–251, Vienna, Austria, 2012.

[271] J. Kolb, K. Kammerer, and M. Reichert. *Updatable Process Views for User-centered Adaption of Large Process Models.* In: Proc 10th Int'l Conf on Service Oriented Computing (ICSOC'12), pp. 484–498, Shanghai, China, 2012.

[272] S. Kabicher-Fuchs, S. Kriglstein, and K. Figl. *Timeline Visualization for Documenting Process Model Change.* In: Proc 5th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'12), pp. 95–108, Vienna, Austria, 2012.

[273] H. Reiterer and T. Büring. *Zooming Techniques.* In: Encyclopedia of Database Systems, pp. 3684–3689, 2009.

[274] B.B. Bederson, J. Meyer, and L. Good. *Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java.* In: Proc 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00), pp. 171–180, San Diego, United States, 2000.

[275] W.A. König, R. Rädle, and H. Reiterer. *Squidy: A Zoomable Design Environment for Natural User Interfaces.* In: Proc 27th Int'l Conf on Human Factors in Computing Systems (CHI'09), Extended Abstracts on CHI, Work-In-Progress Session, pp. 4561-4566, Boston, United States, 2009.

[276] M. Zöllner, H.C. Jetter, and H. Reiterer. *ZOIL: A Design Paradigm and Software Framework for Post-WIMP Distributed User Interfaces.* In: Proc 1st Workshop on Distributed User Interfaces (DUI'11), pp. 37–40, Vancouver, Canada, 2011.

[277] F. Gundelsweiler, T. Memmel, and H. Reiterer. *ZEUS – Zoomable Explorative User Interface for Searching and Object Presentation.* In: Proc Symposium on Human Interface (HCI'07), pp. 288–297, Beijing, China, 2007.

[278] P. Effinger. *A 3D-Navigator for Business Process Models.* In: Proc BPM'12 Workshops, 1st Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'12), pp. 737–743, Tallinn, Estonia, 2012.

[279] R.A. Brown and R.K. Rasmussen. *Virtual Environment Visualisation of Executable Business Process Models.* In: Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches, pp. 68–87, 2010.

[280] S. West, R.A. Brown, and J.C. Recker. *Collaborative Business Process Modeling using 3D Virtual Environments.* In: Proc 16th Americas Conf on Information Systems: Sustainable IT Collaboration around the Globe, Association for Information Systems (AIS), pp. 1–11, Lima, Peru, 2013.

[281] A. Seyfang, K. Kaiser, T. Gschwandtner, and S. Miksch. *Visualizing Complex Process Hierarchies during the Modeling Process.* In: Proc BPM'12 Workshops, 1st Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'12), pp. 768–779, Tallinn, Estonia, 2012.

[282] B. Shneiderman. *Tree Visualization with Tree-maps: 2-d Space-filling Approach.* In: J of ACM Transactions on Graphics (TOG), 11(1), pp. 92–99, 1992.

[283] K. Misue and S. Yazaki. *Panoramic View for Visual Analysis of Large-Scale Activity Data.* In: Proc BPM'12 Workshops, 1st Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'12), pp. 756–767, Tallinn, Estonia, 2012.

[284] R. Likert. *A Technique for the Measurement of Attitudes.* In: Archives of Psychology, 140, pp. 1–55, 1932.

[285] S. Oertelt. *Innovationscontrolling: Ganzheitliches Verfahren zur Priorisierung und Steuerung von Vorentwicklungsprojekten.* Shaker, 2009.

[286] V.R. Basili, R.W. Selby, and D.H. Hutchens. *Experimentation in Software Engineering.* In: J IEEE Transactions on Software Engineering, pp. 733–743, 1986.

[287] N. Juristo and A.M. Moreno. *Basics of Software Engineering Experimentation.* Springer, 2010.

[288] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction.* Springer, 2012.

[289] G.W. Snecdecor and W.G. Cochran. *Statistical Methods.* Blackwell Pub Professional, 1989.

[290] C.H. Lewis. *Using the Thinking Aloud Method In Cognitive Interface Design.* Technical Report, IBM Research Report, RC-9265, 1982.

[291] H.W. Lilliefors. *On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown.* In: J of the American Statistical Association, 62(318), pp. 399–402, 1967.

[292] H.W. Lilliefors. *On the Kolmogorov-Smirnov Test for the Exponential Distribution with Mean Unknown.* In: J of the American Statistical Association, 64(325), pp. 387–389, 1969.

[293] H.B. Mann and D.R. Whitney. *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other.* In: The Annals of Mathematical Statistics, 18(1), pp. 50–60, 1947.

[294] VisualVM. http://visualvm.java.net, February 2015.

[295] L.G. Fine. *The SWOT Analysis: Using your Strength to overcome Weaknesses, Using Opportunities to overcome Threats.* CreateSpace Independent Publishing Platform, 2009.

[296] B. Michelberger. *Suchmaschinen: Einsatz moderner und innovativer Suchtechnologien in Unternehmen.* Master Thesis, University of Applied Sciences Ravensburg-Weingarten, Germany, 2010.

[297] M. White. *Enterprise Search.* O'Reilly & Associates, 2012.

[298] D. Wood, M. Zaidman, and L. Ruth. *Linked Data.* Manning Publications, 2014.

[299] B. Marr. *Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance.* John Wiley & Sons, 2015.

[300] T.H. Davenport. *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities.* Harvard Business School Press, 2014.

[301] B. Weber and M. Reichert. *Refactoring Process Models in Large Process Repositories.* In: Proc 20th Int'l Conf on Advanced Information Systems Engineering (CAiSE'08), pp. 124–139, Montpellier, France, 2008.

[302] D. Knuplesch, M. Reichert, R. Pryss, W. Fdhila, and S. Rinderle-Ma. *Ensuring Compliance of Distributed and Collaborative Workflows.* In: Proc 9th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'13), pp. 133–142, Austin, United States, 2013.

[303] W. Fdhila, C. Indiono, S. Rinderle-Ma, and M. Reichert. *Dealing with change in process choreographies: Design and implementation of propagation algorithms.* In: J of Information Systems, 49, pp. 1–24, Cancun, Mexico, 2015.

[304] W. Abramowicz, K. Haniewicz, M. Kaczmarek, and D. Zyskowski. *Semantic Modelling of Collaborative Business Processes.* In: Proc 1st Int'l Conf on Information, Process, and Knowledge Management (eKNOW'09), pp. 116–122, Cancun, Mexico, 2009.

Part V

Appendices

A Case Study: Raw Results

This chapter shows the raw results of the validation case study from Section 9.3. Section A.1 provides background information for understanding this chapter. Section A.2 then presents the raw results of the performance tests conducted in the case study.

A.1 Background Information

As mentioned in Section 9.3.1, we created six SINs containing 5, 50 and 500 objects, either with smaller files (1 KB) or larger files (100 KB) (cf. Figure A.1). We investigated the performance of add, update and delete operations for both the *pull* and *push algorithm* (cf. Section 5.4.3) in the study. To ensure comparability, all objects within a SIN were equal to each other; i.e., all objects had same properties "author" and "content".

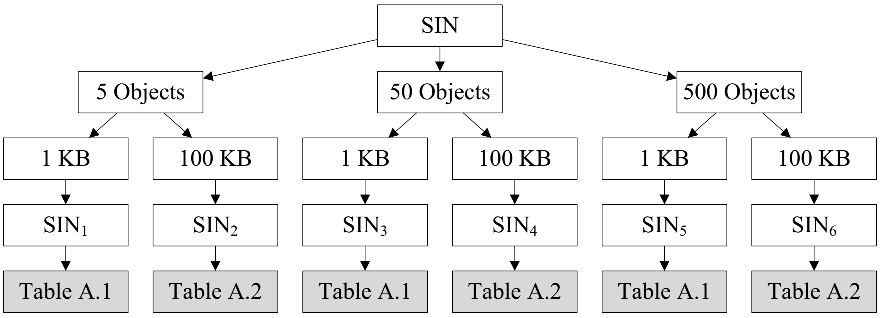

Figure A.1: Correlation between the SINs and the tables.

Note that we simulated worst-case scenarios for the performance tests. This means that each SIN object is related with every other object in the SIN. This results in 40 (considering 5 objects), 4,900 (considering 50 objects), and 499,000 relationships (considering 500 objects). Note that only "is similar to" and "is author of" relationships were discovered since the text files we used do not have the property "title" and therefore no "has same title as" relationships were discovered during the technical analysis.

A.2 Raw Results

Tables A.1 shows the performance of the push and pull algorithm for a SIN comprising smaller files (i.e., 1 KB files). Tables A.2, in turn, shows the performance of the push and pull algorithm for a SIN comprising larger files (i.e., 100 KB files)

			Push Algorithm			Pull Algorithm		
Operation	**Run**	**Objects**	**Object**	**Relation**	**Both**	**Object**	**Relation**	**Both**
Add	1	5	93 ms	440 ms	533 ms	100 ms	436 ms	536 ms
Update	1	5	109 ms	358 ms	467 ms	114 ms	365 ms	479 ms
Delete	1	5	94 ms	31 ms	125 ms	93 ms	31 ms	124 ms
Add	2	5	78 ms	328 ms	406 ms	93 ms	420 ms	513 ms
Update	2	5	93 ms	343 ms	436 ms	106 ms	352 ms	458 ms
Delete	2	5	62 ms	16 ms	78 ms	109 ms	31 ms	140 ms
Add	3	5	78 ms	358 ms	436 ms	98 ms	355 ms	453 ms
Update	3	5	94 ms	327 ms	421 ms	115 ms	332 ms	447 ms
Delete	3	5	62 ms	16 ms	78 ms	109 ms	31 ms	140 ms
Add	1	50	94 ms	1076 ms	1170 ms	110 ms	1155 ms	1265 ms
Update	1	50	93 ms	967 ms	1060 ms	125 ms	1030 ms	1155 ms
Delete	1	50	78 ms	16 ms	94 ms	109 ms	16 ms	125 ms
Add	2	50	109 ms	1014 ms	1123 ms	125 ms	1108 ms	1233 ms
Update	2	50	125 ms	967 ms	1092 ms	110 ms	1014 ms	1124 ms
Delete	2	50	94 ms	15 ms	109 ms	110 ms	16 ms	126 ms
Add	3	50	94 ms	1045 ms	1139 ms	109 ms	1170 ms	1279 ms
Update	3	50	125 ms	967 ms	1092 ms	94 ms	1061 ms	1155 ms
Delete	3	50	94 ms	15 ms	109 ms	110 ms	16 ms	126 ms
Add	1	500	246 ms	8966 ms	9212 ms	312 ms	8751 ms	9063 ms
Update	1	500	293 ms	7998 ms	8291 ms	296 ms	8252 ms	8548 ms
Delete	1	500	243 ms	19 ms	262 ms	250 ms	31 ms	281 ms
Add	2	500	259 ms	8715 ms	8974 ms	265 ms	9111 ms	9376 ms
Update	2	500	272 ms	7847 ms	8119 ms	281 ms	7722 ms	8003 ms
Delete	2	500	244 ms	21 ms	265 ms	187 ms	16 ms	203 ms
Add	3	500	257 ms	8835 ms	9092 ms	328 ms	9064 ms	9392 ms
Update	3	500	284 ms	7869 ms	8153 ms	390 ms	8127 ms	8517 ms
Delete	3	500	244 ms	21 ms	265 ms	187 ms	16 ms	203 ms

Table A.1: Performance of add, update and delete operations using 1 KB files.

			Push Algorithm			Pull Algorithm		
Operation	**Run**	**Objects**	**Object**	**Relation**	**Both**	**Object**	**Relation**	**Both**
Add	1	5	109 ms	3338 ms	3447 ms	125 ms	3572 ms	3697 ms
Update	1	5	109 ms	4898 ms	5007 ms	124 ms	4290 ms	4414 ms
Delete	1	5	78 ms	967 ms	1045 ms	109 ms	1045 ms	1154 ms
Add	2	5	78 ms	4898 ms	4976 ms	140 ms	3620 ms	3760 ms
Update	2	5	93 ms	4820 ms	4913 ms	125 ms	4446 ms	4571 ms
Delete	2	5	47 ms	1498 ms	1545 ms	109 ms	952 ms	1061 ms
Add	3	5	78 ms	3713 ms	3791 ms	109 ms	3448 ms	3557 ms
Update	3	5	110 ms	5554 ms	5664 ms	140 ms	4244 ms	4384 ms
Delete	3	5	47 ms	1498 ms	1545 ms	109 ms	952 ms	1061 ms
Add	1	50	124 ms	5570 ms	5694 ms	140 ms	4649 ms	4789 ms
Update	1	50	125 ms	5819 ms	5944 ms	109 ms	6194 ms	6303 ms
Delete	1	50	124 ms	1124 ms	1248 ms	109 ms	1186 ms	1295 ms
Add	2	50	140 ms	6536 ms	6676 ms	172 ms	5039 ms	5211 ms
Update	2	50	109 ms	9797 ms	9906 ms	171 ms	5881 ms	6052 ms
Delete	2	50	109 ms	1123 ms	1232 ms	110 ms	1092 ms	1202 ms
Add	3	50	93 ms	5507 ms	5600 ms	156 ms	5007 ms	5163 ms
Update	3	50	62 ms	8049 ms	8111 ms	94 ms	6302 ms	6396 ms
Delete	3	50	109 ms	1123 ms	1232 ms	110 ms	1092 ms	1202 ms
Add	1	500	395 ms	31778 ms	32173 ms	265 ms	22667 ms	22932 ms
Update	1	500	296 ms	31577 ms	31873 ms	328 ms	23446 ms	23774 ms
Delete	1	500	265 ms	7903 ms	8168 ms	250 ms	4649 ms	4899 ms
Add	2	500	265 ms	25079 ms	25344 ms	312 ms	21746 ms	22058 ms
Update	2	500	281 ms	29546 ms	29827 ms	312 ms	23697 ms	24009 ms
Delete	2	500	250 ms	8174 ms	8424 ms	234 ms	4648 ms	4882 ms
Add	3	500	281 ms	27347 ms	27628 ms	265 ms	22199 ms	22464 ms
Update	3	500	265 ms	28480 ms	28745 ms	390 ms	24401 ms	24791 ms
Delete	3	500	250 ms	8174 ms	8424 ms	234 ms	4648 ms	4882 ms

Table A.2: Performance of add, update and delete operations using 100 KB files.