RESEARCH

Gilbert Müller

Workflow Modeling Assistance by Case-based Reasoning

Springer Vieweg

Workflow Modeling Assistance by Case-based Reasoning

Gilbert Müller

Workflow Modeling Assistance by Case-based Reasoning

Springer Vieweg

Gilbert Müller
Trier, Germany

Dissertation University of Trier, Department IV, 2018

ISBN 978-3-658-23558-1 ISBN 978-3-658-23559-8 (eBook)
https://doi.org/10.1007/978-3-658-23559-8

Library of Congress Control Number: 2018955144

Springer Vieweg

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH part of Springer Nature
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Acknowledgements

In recent years, many people greatly supported my work and the final outcome presented in this thesis. First and foremost, I would like to express my sincere gratitude to my supervisor *Prof. Dr. Ralph Bergmann* for his inspiring guidance and constant readiness to engage in dialogue. His great expertise strongly influenced my knowledge on Case-based Reasoning and Workflow Management. Not only for my research, but also for the completion of this thesis, his constructive advice was always highly valuable. There are countless things I learned from him during that time. Furthermore, I would like to thank *Prof. Dr. Mirjam Minor* for her immediate agreement to be my second supervisor. The discussions we shared since the very beginning of the EVER project and also her feedback regarding PO-CBR were very helpful for my work. Moreover, I am especially grateful for the cooperative and friendly atmosphere at the *Center for Informatics Research and Technology (CIRT)* at the *University of Trier*. In particular, I would like to thank my colleagues *Dr. Sebastian Görg*, *Christian Zeyen*, *Lisa Grumbach*, *Sarah Gessinger*, and *Maria Gindorf*, who supported me in various ways throughout the accomplishment of this thesis. For implementing some parts of the presented ideas in the CAKE framework, I appreciate the work of *Christian Zeyen*, *Maximilian Hoffman*, and *Jens Manderscheid*. Furthermore, I gratefully acknowledge the detailed proofreading of this thesis by *Lisa Grumbach* and *Mirna Stieler*. Their feedback was highly useful for the revision of my drafts and the final version. Also, I would like to thank the entire ICCBR community for the brilliant conferences and many great debates we had. Especially, I am very glad that *Prof. Dr. Miltos Petridis* and *Prof. Dr. Susan Craw* were my mentors at the ICCBR doctoral consortiums and took time to discuss my work with them. Certainly, all of this would not have been possible without the great support of my family and my friends, who always encouraged me on this exiting journey.

Gilbert Müller
Trier, January 2018

Summary

Nowadays, workflows are applied in an increasing number of application areas to represent and (semi-)automatically execute various kinds of processes. The modeling of workflows, however, is a demanding task as it is not only a time-consuming but also a complex and error-prone activity. Thus, there is a high demand for methods to support workflow modelers in this endeavor. As a consequence, several approaches were recently presented that enable the search for already modeled workflows. However, search is often not sufficient, because workflows more frequently need to be tailored to individual circumstances.

This work addresses that problem by presenting a novel workflow modeling assistance, which automatically constructs workflows based on a given query. The approach applies methods from artificial intelligence, in particular, from the field of Case-based Reasoning (CBR). Case-based Reasoning is a problem-solving methodology that reuses experience from previous problem-solving episodes (here, previously modeled workflows). Following the CBR principle, the workflow modeling assistance searches for the best-matching workflow from a repository of previously modeled workflows according to a specified query and then automatically adapts this workflow, if required. As a result, an individual workflow is automatically constructed. Overall, this work lays highly relevant and new foundations in the field of Process-Oriented Case-based Reasoning (PO-CBR), in which the automated workflow construction is hardly investigated so far. From a workflow management perspective, this work further presents an innovative contribution to support workflow modelers and may be a basis for further research in many workflow application areas.

In more detail, this work first summarizes the most important foundations of workflow management and Case-based Reasoning. Next, a novel query language for the retrieval of workflows is presented, which captures the restrictions and requirements on the desired workflow model. Based on this, three new workflow adaptation approaches are introduced, which are based on well-established methods in CBR. More precisely, the presented *compositional adaptation* decomposes workflows into meaningful components called workflow streams that can be replaced by other matching workflow streams. The *operator-based adaptation* uses operators that specify the deletion, insertion or replacement of workflow fragments. Finally, *generalization and specialization* enable workflow adaptation by means of generalized workflows. The methods are then integrated into a combined adaptation process. In general, all adaptation approaches require adaptation knowledge (e.g., rules or operators) that specifies appropriate modifications in the particular domain. In order to obviate the need for an extensive manual acquisition of adaptation knowledge, new methods are developed in order to learn the required adaptation knowledge automatically. This work also presents several new approaches to further improve the presented workflow modeling assistance, which comprises the automated completion of missing information in workflows, the adaptation-guided retrieval to identify better adaptable workflows, and the consideration of workflow complexity during the construction process. All methods were integrated into the workflow management system CAKE, which is developed at the University of Trier. Based on this, CookingCAKE was developed as a prototypical application in the field of cooking, in which workflows represent real cooking recipes. Furthermore, a comprehensive evaluation in the cooking domain demonstrates the feasibility of the workflow modeling assistance. For this purpose, automatically constructed workflows are compared with workflows resulting from search based on a set of user-generated queries. The evaluation shows that the automatically constructed workflows have a slightly lower quality than workflows manually modeled by humans. However, the queries can be better fulfilled by the automatically constructed workflows compared to the respective workflow resulting from search. Thus, the automatically constructed workflows were preferred and rated with a significantly higher utility by the participants of the study. Overall, this clearly demonstrates the benefit and potential of the developed approach. This work concludes with a discussion on the implications and limitations of the presented workflow modeling assistance and also highlights potential future research directions.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Aims of the Thesis 3
1.3 Approach 6
1.4 EVER Project 8
1.5 Outline 9

2 Foundations **11**
2.1 Process-Aware Information Systems 12
2.2 Business Process Modeling 15
2.2.1 Workflow Terminology 15
2.2.2 Build-time versus Run-time 16
2.2.3 Workflow Representations 18
2.2.4 Workflow Categories 22
2.2.5 Workflow Quality 24
2.2.6 Conclusions 28
2.3 Workflow Management 29
2.3.1 Workflow Management System Architecture 30
2.3.2 Flexible and Adaptive Workflow Management 33
2.3.3 Conclusions 37
2.4 Case-Based Reasoning 38
2.4.1 CBR Cycle 39
2.4.2 Knowledge Containers 41
2.4.3 Case Representation 42
2.4.4 Retrieval 43
2.4.5 Adaptation 45
2.4.6 Maintenance 53
2.4.7 Conclusions 56
2.5 Process-Oriented Case-Based Reasoning 57
2.5.1 Case Representation 58
2.5.2 Retrieval & Adaptation 58
2.5.3 Conclusions 60

2.6 Related Domains & Related Work 60
2.6.1 Application Visions 61
2.6.2 Taxonomy of Workflow Modeling Support Approaches 64
2.6.3 Related Fields in Case-Based Reasoning 69
2.6.4 Conclusions 72

3 Domain & Notation 75
3.1 Business Workflows and Cooking Workflows 75
3.2 Workflow Notation 78
3.2.1 Workflow Properties 80
3.2.2 Partial Workflows 83
3.3 Workflow Ontology 84
3.4 Semantic Workflows 86
3.5 Semantic Workflow Similarity 87
3.6 Conclusions 91

4 Query Language 93
4.1 Requirements 94
4.2 POQL Queries 96
4.3 Query Processing 98
4.4 POQL-Lite 101
4.5 Query Consistency 102
4.6 Potential Extensions 104
4.7 Resulting Modeling Support Scenarios 106
4.8 Conclusions and Related Work 107

5 Workflow Adaptation 109
5.1 Generalization and Specialization 111
5.1.1 Generalized Workflows 112
5.1.2 Specialization of Workflows 118
5.1.3 Summary 125
5.2 Compositional Adaptation by Workflow Streams 125
5.2.1 Automated Learning of Adaptation Knowledge 126
5.2.2 Application of Adaptation Knowledge 133
5.2.3 Additional Application Fields of Workflow Streams . . 138
5.3 Transformational Adaptation by Adaptation Operators . . . 144
5.3.1 Representation of Adaptation Operators 145
5.3.2 Semantics of Adaptation Operators 148
5.3.3 Details of the Operator Application 149
5.3.4 Automated Learning of Adaptation Knowledge 154

5.3.5 Workflow Adaptation using Adaptation Operators . . 156
5.4 Characteristics of Workflow Adaptation Methods 159
5.5 Integrated and Combined Workflow Adaptation 162
5.5.1 Learning of Adaptation Knowledge 163
5.5.2 Performing Adaptation 163
5.6 Conclusions 164

6 Improving Workflow Modeling Assistance 167
6.1 Data-Flow Completion of Workflows 169
6.1.1 Complete Workflows 171
6.1.2 Workflow Completion Operators 174
6.1.3 Learning Workflow Completion Operators 178
6.1.4 Workflow Completion 179
6.1.5 Evaluation 186
6.1.6 Conclusions and Related Work 189
6.2 Adaptation-Guided Workflow Retrieval 190
6.2.1 Adaptation-Guided Retrieval 192
6.2.2 Workflow Adaptability Assessment 194
6.2.3 Evaluation 198
6.2.4 Conclusions and Related Work 200
6.3 Complexity-Aware Workflow Construction 201
6.3.1 Process Complexity 202
6.3.2 Complexity Measure 203
6.3.3 Complexity-Aware Query Fulfillment 205
6.3.4 Experimental Evaluation 206
6.3.5 Conclusions 208
6.4 Summary 208

7 Workflow Modeling Assistance Architecture 209
7.1 Collaborative Agile Knowledge Engine 209
7.2 Workflow Modeling Assistance Architecture 212
7.3 CookingCAKE: Prototypical Application 217
7.4 Interactive Modeling Interface 220
7.5 Conclusions 222

8 Evaluation 223
8.1 Experimental Setup 223
8.1.1 Utility & Application Scenario 224
8.1.2 Evaluation Criteria 225
8.1.3 Hypotheses 227

8.1.4 Workflow Repository 228
8.1.5 Taxonomies 230
8.1.6 Real User Queries 231
8.1.7 Data Revision & Data Set Summary 233
8.1.8 Adaptation Parameters 233
8.1.9 Computing Evaluation Samples 235
8.2 Evaluation Results 236
8.2.1 Analytical Evaluation 236
8.2.2 Experimental Evaluation 240
8.3 Summary 246

9 Conclusion 247
9.1 Contributions 247
9.2 Future Research 251

List of Figures

1.1 Workflow modeling assistance by PO-CBR 4
1.2 Information systems research framework 7

2.1 PAIS lifecycle 14
2.2 Workflow build-time and workflow run-time 17
2.3 Sequence pattern 19
2.4 AND-split and AND-join nodes 19
2.5 XOR-split and XOR-join nodes 20
2.6 Cycle pattern 20
2.7 Example data-flow 21
2.8 Workflow categories 22
2.9 Semiotic quality model 24
2.10 Devil's quadrangle 27
2.11 Workflow reference model 30
2.12 Structure of a workflow management system 31
2.13 Status transitions of workflow instances 32
2.14 Status transitions of workflow activities 33
2.15 Types of workflow flexibility 34
2.16 CBR cycle 40
2.17 CBR knowledge containers 41
2.18 Case retrieval 44
2.19 Adaptation models 46
2.20 Rule-based adaptation 47
2.21 Operator-based adaptation 48
2.22 Compositional adaptation cycle 50
2.23 Learning adaptation knowledge from case base 53
2.24 Application and maintenance of CBR systems 54
2.25 PO-CBR setting for workflows 59
2.26 Workflow modeling support taxonomy 64

3.1 Example business workflow 76
3.2 Example cooking workflow 77
3.3 Workflow block elements 81

3.4 Example of a partial workflow 84
3.5 Example of a data taxonomy 85
3.6 Workflow fragment with semantic annotations 86

4.1 POQL query example 97
4.2 Query fulfillment computation 99
4.3 Inconsistent query examples 103
4.4 Example transitive control-flow and data-flow connectedness . 105

5.1 Example generalization/specialization relationships 113
5.2 Example generalization 115
5.3 Example specialization scenarios 121
5.4 Workflow and workflow streams 127
5.5 Workflow stream extraction for a creator task 129
5.6 Workflow stream extraction for remaining task 130
5.7 Example stream 131
5.8 Substitute stream 134
5.9 Example stream removal 135
5.10 Example stream insertion 136
5.11 Workflow stream abstraction examples 139
5.12 Example of component-based worfklow modeling 140
5.13 Late-binding by component-based workflow modeling 141
5.14 Workflow stream chains example 142
5.15 Example pasta workflow 145
5.16 Example streamlet 147
5.17 Example adaptation operator 148
5.18 Example of head data node removal 150
5.19 Example of plain streamlet insertion 152
5.20 Example cleanup after operator application 153
5.21 Mapping example and resulting adaptation operators 155
5.22 Operator-based adaptation 157
5.23 Generation of adaptation knowledge 163
5.24 Combined workflow adaptation process 164

6.1 Workflow modeling assistance (abstract illustration) 167
6.2 Framework of workflow completion 170
6.3 Examples for incomplete data-flow information 172
6.4 Example of strict consistency (identical branches) 173
6.5 Example of strict consistency (identical creator data node) . . 173
6.6 Example completion operator 174

6.7 Example generalized completion operator 175
6.8 Example mapping between operator and incomplete workflow 176
6.9 Example result for an operator application 177
6.10 Default completion rule bases 181
6.11 Example of default completion rule 1 182
6.12 Example of default completion rule 2 183
6.13 Example of default completion rule 3 183
6.14 Example of default completion rule 4 184
6.15 Example of default completion rule 5 184
6.16 Workflow retrieval and workflow adaptability 191
6.17 Adaptability assessment 194
6.18 Difference-based query construction 195
6.19 Evaluation result of adaptation-guided retrieval 200

7.1 CAKE architecture 210
7.2 CAKE configuration 212
7.3 CAKE data types 213
7.4 Workflow modeling assistance architecture 215
7.5 CookingCAKE: Query definition (simple mode) 217
7.6 CookingCAKE: Query definition (expert mode) 218
7.7 CookingCAKE: Result page 219
7.8 Interactive workflow modeling interface 220

8.1 Application environment and utility of a PO-CBR system . . 224
8.2 Workflow modeling example based on textual descriptions . . 229
8.3 Extract of task and data taxonomies 230
8.4 POQL-Lite query example 231
8.5 POQL-Full query example 232
8.6 Adaptations performed for evaluation 235
8.7 Success rate of adaptation 237
8.8 Example of textual workflow representation 240
8.9 Overall workflow quality ratings in comparison 243

List of Tables

5.1 Properties of workflow adaptation methods 160

6.1 Evaluation results of workflow completion 189
6.2 Adaptability table 197
6.3 Evaluation results of adaptation-guided retrieval 199
6.4 Complexity criteria 203
6.5 Evaluation results of complexity-aware workflow construction 207

8.1 Average workflow size 228
8.2 Average query size 232
8.3 Basic data set 233
8.4 Adaptation parameters 234
8.5 Average query fulfillment & computation time 236
8.6 Structural change of workflow 239
8.7 Quality criteria 241
8.8 Quality assessment 242
8.9 Average utility 244
8.10 Application preference 245

1 Introduction

1.1 Motivation

In recent years, workflows developed towards an important paradigm to represent and execute various kinds of processes. From a business perspective, workflows are "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [266][p. 8]. Broadly speaking, workflows serve as a model to represent sequences of activities as well as information that is shared between those activities in order to execute a certain business process.

Workflows representing business processes are applied in e-commerce [123], financial services [214, 196], the service industry [197], and many other application areas. Beyond this business perspective, workflows recently gained significant importance to represent and execute many kinds of processes in various domains. A well-established application area, for example, is the analysis of large data sets employing so-called scientific workflows [242, 76, 36] including domains such as biology, ecology, physics and geology [34]. Besides, workflows are also used to support medical treatment processes [176, 52] and provide a basis for a novel programming paradigm. Flow-based programming [165] links several small sub components with each other in order to construct an entire program code. Moreover, workflows are considered to support processes in everyday life [80] and can also be applied to represent information gathering processes [69, p. 90-93] or cooking instructions [222]. In summary, it can be stated that workflows are applied in numerous domains and application areas.

It is commonly agreed that the creation of workflows, also referred to as workflow modeling, is a time-consuming task with a high degree of complexity [253]. Thus, in all the previously mentioned domains, an important key to the successful application of the workflow paradigm is the *reuse* of those workflows. This implies that once created workflows are used several times again in same or similar scenarios. Hence, workflows do not have to be modeled from scratch, but can be based on already modeled workflows,

G. Müller, *Workflow Modeling Assistance by Case-based Reasoning*,
https://doi.org/10.1007/978-3-658-23559-8_1

which significantly reduces workflow modeling time [79]. Furthermore, the quality of modeled workflows is increased by utilizing experience gained by other workflow modelers in previous workflow modeling scenarios [79]. Moreover, based on established and validated workflows, reuse additionally leads to less error-prone workflow models [79]. Workflow reuse is thus of vital importance, as it substantially facilitates workflow modeling and improves the resulting workflow in many respects.

Reuse of workflows is currently supported by various search methods (e.g., [20, 59, 6, 13, 14, 6]). They enable to identify the most suitable workflow stored in a workflow repository with regard to given restrictions and requirements. More precisely, a search for the workflow best matching the current situation is executed. This workflow can either be reused directly or requires a manual modification of the workflow to suit the particular situation. Consequently, the workflows do not have to be modeled from scratch, which significantly supports workflow modelers.

However, current trends in workflows and its application areas indicate an increasing demand for individuality and flexibility, which is partially caused by a turbulent market and technological innovation force [3]. Flexibility, addresses the need to modify or change the workflow on demand during execution and has been addressed by adaptive and flexible workflow management [262, 249, 194, 219]. Individuality, in contrast, implies the requirement to model workflows tailored to the particular needs in the current scenario or situation. The increasing demand for individuality causes that the reapplication or exclusive search for workflows is not sufficient anymore, since available workflows often do not match the current situation. As a result, workflow modelers are frequently required to adapt the provided workflow manually. However, the manual modification of the workflow can be a time-consuming and complex task. If the found workflow, consequently, differs clearly from the desired workflow, the requirement of manual modification can be a threat to the successful reuse. At worst, workflow reuse is circumvented, if the creation of a new workflow from scratch is preferred. This then results in a loss of all the previously described benefits of workflow reuse.

Consequently, new adaptive methods are required to support workflow modelers during the construction of individual workflows tailored to their particular needs. This thesis presents an approach to workflow modeling assistance by automated workflow construction. More precisely, in order to support the user, the requirements and restrictions of the particular scenario are captured in a query and a most suitable workflow from a workflow repository is identified. Subsequently, this workflow is adapted automatically by removing, adding, or replacing workflow elements such that the workflow better matches the specified query. As a result, a workflow is automatically constructed according to the requirements and restrictions in the particular scenario. This substantially facilitates the elaborate task of workflow modeling and ensures that the workflow modeler can still benefit from workflow reuse, since the automatically created workflow is based on previously modeled workflows. Thus, the described automated workflow construction can be a means to a novel workflow modeling assistance that can cope with the increasing demand for individuality.

1.2 Aims of the Thesis

This thesis aims at developing methods to provide a workflow modeling assistance that facilitates the complex, error-phone, and time-consuming task of workflow modeling [253, 79]. The developed workflow modeling assistance is based on methods that have already been applied successfully in Case-Based Reasoning (CBR). Case-Based Reasoning is a problem-solving methodology that reuses previously gathered experiences [200, 1, 16]. Problems are solved by searching for the most similar situation experienced in the past. Then, the corresponding solution can be reused directly or is adapted automatically to suit the current situation. Process-Oriented Case-Based Reasoning (PO-CBR) [157] applies this Case-Based Reasoning methodology on processes, for example, represented as workflows. This implies that workflows can be suggested based on experience gained in previous workflow modeling scenarios. For this purpose, a search is performed to identify a workflow from a repository of previously modeled workflows that best matches the current scenario. Thus, PO-CBR supports workflow modeling, since workflows do not have to be modeled from scratch, but can be created based on an existing workflow that already (partially) matches the desired solution. By means of a subsequent adaptation process, PO-CBR would further enable the automated construction of workflows tailored to particular needs. However, besides some initial investigations (e.g., [152, 155]), research in PO-CBR not yet addressed workflow adaptation. This gap in research on workflow adaptation certainly limits the scope and application of current PO-CBR systems[1] with regard to workflow modeling assistance, as they can only provide a search for previously modeled workflows. Therefore, this thesis will particularly investigate the important topic of workflow adaptation in PO-CBR in order to establish an enhanced workflow modeling assistance

[1] Any CBR system without adaptation capabilities is limited with regard to scope and application [257]

that may cope with the increasing demand for individual workflow reuse, i.e., that workflows more frequently need to be tailored to the particular needs. Hence, this work addresses a highly relevant problem in numerous domains.

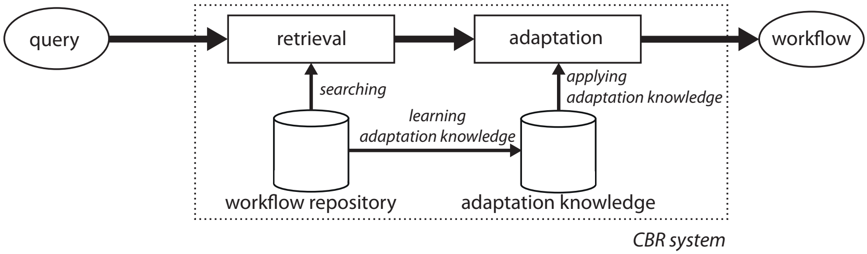

Figure 1.1: Workflow modeling assistance by PO-CBR

The basic idea of the intended workflow modeling assistance by means of PO-CBR is illustrated in Figure 1.1. More precisely, the user first specifies the restrictions and requirements on the desired workflow in a query. Next, for the specified query, the most suitable workflow from a workflow repository is retrieved. In a subsequent adaptation process, workflow elements are modified such that deficiencies of the workflow with regard to the particular query are compensated. For this purpose, adaptation knowledge describing valid transformations of a workflow is employed. As a result, a workflow according to the specified query is automatically constructed, which can significantly support users in the elaborate task of workflow modeling. The main research objective of this thesis is to investigate this sketched idea of workflow modeling assistance by means of PO-CBR by addressing the following partial objectives.

Novel Query Language. As this thesis aims at assisting the user in the elaborate task of workflow modeling, an easy to understand query language is required. Additionally, the query language needs to capture the restrictions and requirements on the desired workflow. This query language further must be able to assess the query fulfillment of workflow solutions and possible adaptations in order to guide the retrieval and adaptation process.

Novel Adaptation Methods. A workflow W can be transformed into an adapted workflow W_n by chaining various adaptation steps $W \overset{s_1}{\to} W_1 \overset{s_2}{\to} \ldots \overset{s_n}{\to} W_n$. This process can be considered as a search process towards an optimal solution with regard to the query. For enabling workflow adaptation in PO-CBR, novel adaptation methods have to be developed. The approaches should be able to consider the previously defined query language during the transformation of the workflow in order to find the best possible solution. For this purpose, various workflow adaptation methods for PO-CBR will be developed that draw on adaptation principles successfully applied in CBR.

Integrated Adaptation. Adaptation methods are usually associated with their respective advantages and disadvantages or restrictions and requirements. Consequently, the competence of the system is affected depending on the particular scenario or domain. Thus, this thesis further aims at integrating and combining all developed adaptation approaches into a single adaptation process in order to compensate the disadvantages of the particular workflow adaptation methods. Thereby, a capable tool for the adaptation of workflows should be constructed.

Automated Learning of Adaptation Knowledge. A major threat to the adaptation in CBR, so far not mentioned, is that adaptation knowledge usually has to be defined manually. Such a manual modeling process is expensive with regard to time and complexity. This concerns in particular PO-CBR due to the inherent intricacy of workflows. During the definition of adaptation knowledge, many restrictions and scenarios in the particular domain have to be considered. Moreover, the adaptation knowledge depends on the adaptation algorithm. Hence, an expert in workflow modeling, who is further able to understand the consequences of the adaptation knowledge with regard to the adaptation algorithms in the particular domain, would be required in order to acquire useful adaptation knowledge. This usually leads to an acquisition bottleneck of adaptation knowledge [93], as the manual acquisition of adaptation knowledge is mostly not feasible. As a result, the competence of the CBR system is reduced. The developed adaptation methods should therefore include methods to learn adaptation knowledge automatically. In this thesis, the required adaptation knowledge will be obtained from the workflows stored in the repository (see Figure 1.1). Thus, the applicability of the presented adaptation methods is increased by reusing domain specific-knowledge automatically. This additionally reduces the setup time of the PO-CBR system significantly, as no expert is required to define adaptation knowledge manually.

Workflow Completion. The intended workflow modeling assistance by means of PO-CBR needs to be based on mostly complete workflows. However, existing workflow repositories frequently contain incomplete workflows with insufficiently specified information. This may not only lead to inappropriate and incomplete workflows selected during retrieval, but also results in adaptation knowledge that is incomplete in itself. Employing such incomplete workflows may thus significantly affect the entire PO-CBR system. Hence, methods are required to complete missing information in the stored workflows automatically. By this means, the PO-CBR system can be based on complete workflows, which improves the quality of the workflows constructed by the workflow modeling assistance.

Integrating Retrieval and Adaptation. The previous aims address the retrieval and adaptation of workflows by considering adaptation as a post-processing step after workflow retrieval. However, Smyth and Keane [233] already stated that it is important to consider the adaptability during the retrieval stage. Otherwise, retrieval may provide a workflow that cannot be at best adapted according to the query, resulting in a non-optimal workflow solution. Hence, methods are required to assess the adaptability of workflows, for instance, by performing several example adaptations. Considering this estimated adaptability value during workflow retrieval could ensure that the selection of the workflow is not a limiting factor for the subsequent adaptation.

Complexity-Aware Workflow Construction. In certain situations, further requirements on the desired workflow must be considered in addition to the query. In particular, the reduction of complexity is an important criterion, since it increases the understandability of the workflow model (see, e.g., [198, 148]). Thus, less complex workflow models can be highly beneficial, especially for novice workflow modelers. Moreover, a lower complexity facilitates the maintenance and contributes to a reduced error-proneness (see ,e.g, [43, 145]) of the workflow model. Consequently, an approach is required to integrate this additional criterion into the workflow modeling assistance such that workflows with a low complexity can be generated, if required.

1.3 Approach

The research approach of this work follows the design science theory in information systems research [99, 185], which "[...] addresses important unsolved problems in unique or innovative ways or solves problems in more effective or efficient ways" [99][p. 81]. According to Hevner et al. design science "[...] creates and evaluates IT artifacts intended to solve identified organizational problems. Such artifacts are represented in a structured form that may vary from software, formal logic, and rigorous mathematics to informal natural language descriptions." [99][p. 77].

Furthermore, the authors present an information systems research framework (see Fig. 1.2), which describes that the environment of people, organizations and technology defines business needs on information systems research. Pursuing information systems research with the goal to address those needs with the developed methods in the given environment ensures the relevance of the research. Thus, artifacts have to be constructed that aim at fulfilling or satisfying business needs. On the other hand, research rigor must be ensured, which can be "derived from the effective use of the knowledge base - theoretical foundations and research methodologies" [99]. Hevner et al. [99] state that information systems research has to build artifacts that are both relevant and rigorous.

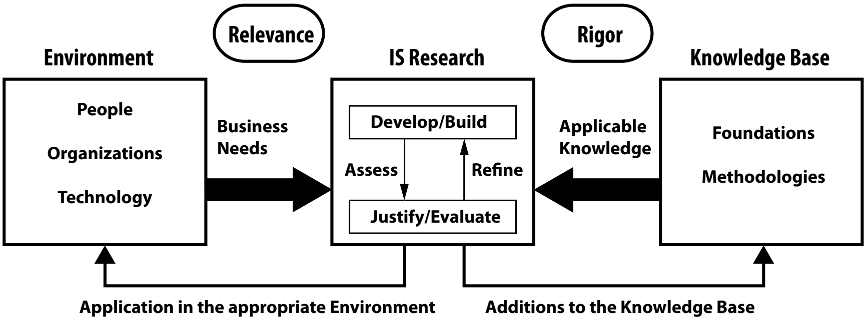

Figure 1.2: Information systems research framework by Hevner et al. [99] (simplified)

The business needs can usually be identified by a field or case study or result directly from gaps in the environment. This thesis addresses a highly relevant problem in numerous workflow domains, which directly results from the increasing demand to construct individual workflows tailored to particular needs. The developed artifacts are based on theories and methodologies from Case-Based Reasoning, methods in Artificial Intelligence, and on related work already addressing workflow modeling support. Hence, research rigor and relevance of the presented research is ensured. The developed methods will be integrated into the CAKE framework [19]. CAKE "[...] is a prototypical generic software system for integrated process and knowledge management. CAKE integrates recent research results on agile workflows, process-oriented Case-Based Reasoning, and web technologies into a common platform that can be configured to different application domains and needs." [19][p. 1].

To access the utility of the developed artifacts, a comprehensive evaluation in the cooking domain is conducted. In the cooking domain, workflows represent cooking recipes that consist of cooking instructions describing preparation steps to be executed and ingredients to be used in order to prepare a particular dish. Cooking workflows will not only be used for the evaluation, but also as a running example for demonstrating the presented approaches throughout this thesis. The cooking domain has been chosen due to multiple reasons. A major argument is that repositories of workflows can be acquired easily, as the information is readily accessible through cooking recipes. Moreover, the required background knowledge is easy to obtain (if not already available). This can be very difficult in other domains leading to an increased effort in developing methods and finding experts evaluating the presented approach. Furthermore, in CBR research the cooking domain has been established as a frequently used example domain, which enables to compare the presented approaches. A natural reason is also that the presented methods have been developed within the EVER project, in which the cooking domain has been chosen for similar reasons.

1.4 EVER Project

The author was a research associate within the EVER project [22][2] "Extraction and Processing of Procedural Experience Knowledge in Workflows"[3] funded by the German Research Foundation DFG[4], which initiated the contributions presented in this thesis. The project was a joint work by the University of Trier, Goethe University Frankfurt, and the University of Marburg. The objective of this research project was to gather procedural experience that is implicitly available on websites, blogs or discussion boards, and to make this knowledge available for others in a structured manner, i.e., workflows. Methods were consequently developed to gather the procedural knowledge automatically from websites in the form of workflows. Based on this, the search for workflows was investigated, which enables the reuse of the gathered procedural knowledge. Also, the reasoning on workflows was addressed such that procedural knowledge for new situations can be provided based on the gathered experience. In this sense, retrieval and adaptation of workflows also played a key role in the EVER project.

[2] http://ever.wi2.uni-trier.de
[3] Ger. "Extraktion und Verarbeitung von prozeduralem Erfahrungswissen in Workflows"
[4] Ger. "Deutsche Forschungsgemeinschaft"

1.5 Outline

The remainder of this thesis is organized as follows. In the next chapter, the fundamentals for the intended workflow modeling assistance will be explained. This comprises workflow management, workflow modeling, as well as Case-Based Reasoning (CBR). Moreover, PO-CBR is described in more detail. This chapter further sketches some visionary applications and describes various related approaches to workflow modeling assistance in general.

Next, Chapter 3 introduces the example domain and formal notations used in the following chapters. More precisely, cooking workflows that represent cooking recipes will be described, which serve as a running example for illustrating the approaches throughout this thesis. Based on this, the notations on workflows are introduced, which are use to formalize and explain the developed approaches. Furthermore, workflow ontologies, semantic workflows, and a semantic workflow similarity measure are described that also represent the foundations for the remaining chapters.

In Chapter 4, a novel query language POQL for the retrieval and adaptation of workflows is introduced. It captures the restrictions and requirements of the user and enables to assess how well these are fulfilled. Thus, POQL can guide the entire workflow modeling assistance process.

In the next chapter (see Chap. 5), three different adaptation methods for workflows are presented that are able to adapt a workflow with regard to a given POQL query. For all introduced adaptation approaches, methods are presented to obtain the required adaptation knowledge automatically. Further, the characteristics of the different approaches are discussed. These adaptation methods are then integrated and combined into a single adaptation process.

Chapter 6 illustrates various approaches to improve the retrieval and adaptation process of the workflow modeling assistance. This comprises the completion of missing information within workflows, an approach to complexity-aware workflow construction, and an integrated method of retrieval and adaptation aiming at regarding the adaptability already during retrieval.

In Chapter 7, the *CAKE* framework developed at the University of Trier is described and technical insights about the implementation of the novel approaches within this framework are provided. Furthermore, a prototypical application called *CookingCAKE* for demonstrating the workflow modeling assistance is presented, which enables the automated construction of cooking recipes.

Next, Chapter 8 presents a comprehensive evaluation in the cooking domain. The evaluation is based on a comparison of workflows constructed by the presented workflow modeling assistance with workflows resulting from search for a set of user-generated queries. Automatically computed evaluation criteria as well as evaluation criteria assessed by experts are investigated in this evaluation in order to demonstrate the feasibility and usability of the novel workflow modeling assistance.

Finally, Chapter 9 summarizes the achievements and discusses potential future research directions.

2 Foundations

Triggered by the emergence of Business Process Reengineering in the '90s, the perspective of software systems shifted from a pure data perspective towards process orientation. This resulted in the development of Process-Aware Information Systems (PAIS), which are systems that deal with the management and execution of processes. One instance of PAIS are workflow management systems (WfMS), which manage and execute processes represented as workflows. While PAIS were traditionally developed for business processes, the application field of workflows nowadays goes beyond this traditional perspective. One of the most essential characteristics in any WfMS is that workflows have to be modeled, prior to execution. This thesis addresses that particular stage. Thus, this chapter explains the foundations for structuring and modeling workflows. It will further be illustrated how the modeled workflows are executed within a workflow management system, since this has to be considered also during the modeling stage. Moreover, workflow quality will be discussed in this chapter. The quality of the modeled workflow is highly important, since it greatly determines the successful execution of the modeled process. Quality dimensions will be described and comprise, for example, the performance of the process with regard to quality of the final product or the understandability of the modeled process.

Furthermore, the foundations will also describe the rudiments from a computational perspective. Case-Based Reasoning (CBR) has been successfully employed as a problem-solving paradigm in many application areas and domains. During workflow modeling, the workflow designer has to identify and create a suitable workflow process model for a particular scenario, which is basically such a kind of a problem-solving activity. Consequently, Process-Oriented CBR (PO-CBR) can be a means to support the modeling of workflows. However, only little research exists in the field of PO-CBR so far. Thus, this thesis will illustrate how to transfer successfully applied methods from CBR to PO-CBR in order to support workflow modeling. This section will therefore explain the fundamentals of CBR, in particular focusing on retrieval as well as adaptation methods. For modeling support, the retrieval (search) for already known workflow models is crucial. This helps the user by avoiding the need to model the workflow from scratch. Instead, the user can

G. Müller, *Workflow Modeling Assistance by Case-based Reasoning*,
https://doi.org/10.1007/978-3-658-23559-8_2

reuse the identified model either directly or performs few modifications to adapt the model to match the particular demands. However, as this thesis addresses the creation of individual workflows, only considering search is not sufficient. More useful workflows can be created only if they are adapted automatically towards the desired restrictions and needs of the user. Thus, this chapter also discusses various adaptation methods successfully applied in CBR. These approaches usually base on adaptation knowledge. In order to prevent the extensive manual creation of such adaptation knowledge, this chapter further illustrates how it is possible to learn the required adaptation knowledge automatically.

After introducing the foundations of Business Process Management and Case-Based Reasoning, this chapter will describe the basic idea of Process-Oriented Case-Based Reasoning. Based on this, several application visions will be sketched. Next, related approaches to workflow modeling assistance will be presented.

2.1 Process-Aware Information Systems

In *The Wealth of Nations* published in 1776, Adam Smith described the fragmentation of work into specialized tasks as the so-called *division of labor*, which significantly increased enterprises productivity [230]. Hammer and Champy [87][p.7-30] argued in the early '90s that task-oriented jobs according to the division of labor became inappropriate to handle new developments on the market. That is because the organization around activities neglect the perspective on the outcoming result. Since then, customer demands could no longer be fulfilled with the mass market as customers call for individual products for their particular needs. Furthermore, globalization has led to an increased competition and the way of doing business has changed through technology. Another reason is that changes in the business environment became ubiquitous and continuous. Specialized work in fragmented processes is not very responsive and flexible to changes on the market. Finally, task-orientation also hampers innovation and creativity in an organization, results in a high overhead, and leads to an absence of customer orientation. In the course of Business Process Reengineering in the early '90s the process perspective has consequently gained significant importance. According to Hammer and Champy "[...] reengineering is the fundamental rethinking and radical design of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed" [87][p. 32]. Thus, business process redesign focuses on the improvement of processes in order to increase, for example, customer satisfaction, improve the efficiency and quality of business processes, reduce the cost of processes, and cope with new business challenges and opportunities [75].

A shift towards process-orientation means that the work is organized around the business process. Davenport has defined a business process as “simply a structured, measured set of activities designed to produce a specified output for a particular customer or market. [...] A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action.” [55, p.5]. This definition captures the customer-orientation as well as the process-orientation focusing on the particular output. Hammer and Champy [87][p. 47] highlight that information technology enables a radical redesign of business processes whereby the process-orientation facilitates to overcome organizational boundaries improving the performance of business processes. Consequently, business process reengineering has significantly changed the perspective of information systems.

From a technological perspective, the early business information systems mainly focused on storing, searching, and displaying of information [64][p. 4-5]. These systems were thus solely driven by data for the purpose of supporting employees in specific tasks following the division of labor. For enterprises, this means that the actual business processes have been neglected in those IT systems. In the worst case, business processes were structured such that they suit the present IT systems. The entire business processes consequently involved multiple IT systems and manual procedures which hampered the efficient execution of business processes. Factors for inefficiencies included manual resource allocation and work routing, no clear separation of responsibilities, possible work overflows, and redundant manual data input [64][p. 4]. Furthermore, due to multiple IT systems, a need for a global view on the operation of information systems emerged. Thus, there was a high demand for so called Process-Aware Information Systems (PAIS) systems following Business Process Reengineering developments towards a process-orientation. A holistic view on process-orientation with regard to the modeling of information systems to support business processes has also been described by the ARIS architecture [217] [219][p. 51].

A PAIS is “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models” [64][p. 7]. Thus, Process-Aware Information Systems support entire business processes and have been enablers for a radical redesign of business processes. PAIS usually separate process logic from the applications and facilitate a modeling of processes without a recoding of the system [64][p. 4-5]. This is also reflected by the four phases of the so-called PAIS lifecycle (see Fig. 2.1) [64][p. 11-12]. In the design phase process models are constructed, i.e., processes are modeled. Next, software systems, for example, a workflow management system, have to be implemented and developed which support the execution of these process models (system configuration phase). In the next phase, these process models can be executed, which is also referred to as process enactment. In the final phase, the executed processes are analyzed to identify problems and optimization options in order to revise the design of the processes (diagnosis phase). Thus, the modeling of processes is a main aspect in any PAIS as without process models no system can be designed and no processes can be executed. Prominent examples of PAIS systems are, for example, Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) systems, and Workflow Management systems (WfMS) [209].

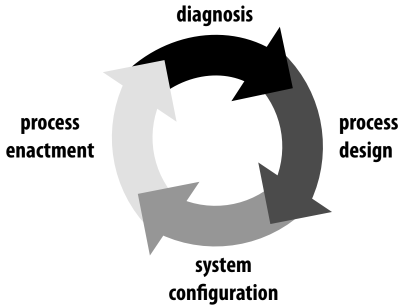

Figure 2.1: PAIS lifeycle by Dumas et al. [64][p. 11-12]]

The shift towards a process-orientation by means of PAIS enabled an improved communication between the involved stakeholders, for example, managers and IT professionals through the use of explicit process models. These explicit process models further support the management during controlling, evaluating and improving of processes. Moreover, they enabled a more global view and an automatic enactment of the business processes. The separation of process models from the PAIS also allows the modification of business processes, which is demanded by continuously changing business environments.

2.2 Business Process Modeling

Business Modeling [129][p. 31] is one of the first phases of business process management, which commonly involves different stakeholders [252] [263][p. 11-16]. During this process, surveys, discussions, and process improvement activities lead to an informal business process description in which process goals, critical success factors, organization structures, and business objects have to be identified primarily [129][p. 31-32]. Business Process Modeling is the phase in which this informal description is formalized by constructing a business process model using a particular process model notation. A process model is also often used during the entire design process, since it is continuously validated and improved until it represents the desired business process. Thus, a comprehensible representation of a process model is obviously required.

Various types of PAIS have been created to support different process representations such as event-driven process chains (EPCS) [218], petri nets [57] or UML [67]. This thesis focuses on processes represented as workflows, which will be introduced in the following.

2.2.1 Workflow Terminology

The workflow terminology is based on the specifications of the Workflow Management Coalition (WfMC)[1]. The WfMC is a non-profit organization founded in 1993 aiming at the development of common terminologies and standards to enhance the opportunities of the workflow technology [266][p. 5]. The WfMC defines workflows as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [266][p. 8]. Broadly speaking, a workflow consists of an ordered set of tasks. The WfMC defined these tasks or activities as "a description of a piece of work that forms one logical step within a process. An activity may be a manual activity, which does not support computer automation, or a workflow (automated) activity. A workflow activity requires human and/or machine resources(s) to support process execution; where human resource is required an activity is allocated to a workflow participant." [266][p. 13] Consequently, these tasks may be either manual tasks such that they have to be performed by a particular participant or automated tasks which are autonomously executed by a certain application. The execution order of these tasks, also referred to as the *control-flow*, is defined by constructors

[1] http://www.wfmc.org