

2 Fundamentals

This chapter explains the background in which the idea of social workflows evolved. In fact, the vision of social workflows was significantly influenced by the today's behaviour of private individuals in online communities and the possibilities offered by modern Process-Aware Information Systems (PAIS). The workflow technology of PAIS allows the automatic execution of social workflows. Since the execution of social workflows is one of the aims of the thesis and probably essential to create a benefit for private individuals, the fundamentals will start with the technological background of PAIS. The first section explains where workflow management comes from and how workflows are normally created and managed.

After that, online communities are regarded in detail. If people would not excessively use social networks and online communities then the idea to build a social workflow community – based on voluntary participation – would be pointless; but they do. Therefore, the second section elaborates the differences and commonalities between social networks and online communities. This is relevant for social workflows because a social workflow platform shall allow the easy integration of social network members to enable the delegation of personal tasks to friends. On the other side, the collection of existing social workflows shall serve an online community as knowledge repository. In this regard this section deals with the question why people tend to participate and share contents with a community.

For browsing a repository of social workflows in a social workflow platform, a technology is needed that allows to search for social workflows in a manner that a query can express the desired search result of a user. Such a search is mandatory to exploit the made experiences others encountered with social workflows and to foster the reuse of established workflows within the community. Because social workflows are not represented in a textual form but in a workflow structure, methods from Process-Oriented Case-Based Reasoning (PO-CBR) are presented for

the retrieval of workflows. These methods shall help new users to create their own social workflows by learning from existing social workflows provided by the the community.

The section on the fundamentals closes with a view on related research and existing workflow communities. Social workflows are a new research area but the idea to directly connect private individuals to workflow technology has been already picked up by research in different disciplines. Moreover, workflow communities in a non-business context already exist focussing on scientific workflows. The investigation of related research and workflow communities is the starting point for the positioning of social workflows in research as the first aim of this thesis.

2.1 Process-Aware Information Systems

In the 1990s, with the increasing complexity due to globalization and the linked competition, the business world shifted towards a process-oriented view how products and services are created. Whereas Adam Smith's first chapter "Of the Division of Labour" in his famous book "An Inquiry into the Nature and Causes of the Wealth of Nations" [Smi76] focussed on the distribution of tasks to the available set of workers, his theory was missing a holistic view of the process to create a product. In contrast to focus on the specialization of certain tasks in a company to increase the quality, Business Process Management (BPM) realized that only focussing on activities is not enough as each product is the outcome of a number of activities performed [Wes07]. The idea of process thinking originated at least as far back as Frederick Taylor's "scientific management" evolved at the end of the 19th century [JN06]. However, with the upcoming development of Information Technology, the idea was picked up in 1993 when Hammer and Champy [HC93] postulated their business reengineering approach. They argued that most enterprises require a reengineering of their business in accordance to their business processes to be competitive. Complex business processes involving dozens of people working across different departments cannot be made flexible enough to deal with special requests or to respond to enquiries without a process-oriented view which covers all activities of a business process. In a task-oriented view no one would be empowered to answer a question or to solve upcoming problems which go beyond a dedicated task or the

belonging department. The traces of the creation of a product would vanish and hence the comprehensive knowledge about its production process.

Rivard et al. [RAM09] illustrate the advantage of a process-oriented view with a relay race: A team does not necessarily win this type of race by lining up the four best sprinters in the world, but it is vital to pass the baton properly. A relay race is a process which links four athletes with a baton handover. In terms of a race in which team cohesion is more crucial than individual strength, a process-oriented view can be seen as a business attitude that aligns all individual business actings to existing business processes.

Knowledge of business processes requires a considerable amount of information and processes are modelled regarding different abstraction levels [RAM09]: Each activity must be documented in detail, the stakeholders, their organizations and the relationships that link them (by required or created information) must not be forgotten, just like the business rules that govern the processes and the infrastructure used to support them. Dealing with all these information requires a PAIS. It is defined as “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models” [DWA05, p. 7]¹. PAIS can be seen as special kinds of Information Systems (IS) that are process aware and link information technology to business processes [DAT05]. The relevance of process models for PAIS and the reasons why enterprises strive to reuse these process models is described in the following. It starts with a short description of workflow management, introduces some specific workflow terms, explains flexibility approaches for workflows, and ends with the invocation of web services in workflows.

2.1.1 Workflow Management

A workflow is usually regarded as “a computerised facilitation or 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” [Hol95, p. 6]. Oberweis

¹In [DAT05] the original definition is for process-aware information systems (PAIS). In some German publications it is often called process-oriented information systems (POIS) meaning the same.

[Obe05] adds that workflows consist of a set of activities that are executed to achieve a predefined goal. Further, Oberweis [Obe05] describes workflow management as the support for routing activities in an organization so that the work is efficiently done at the right time by the right person with the right software application. In principle, workflow management can be performed without the use of technology. In fact, this was the case before workflow management technology or even information technology was developed at all [Rei03]. In its simplest form workflow management is done on a piece of paper noting all necessary activities with the responsible persons. Sometimes, business processes within an organization are even hidden. They only manifest in the minds of the people and the employed application programmes, without being driven by an a priori conception of the way tasks should be conducted [DWA05]. In the following, these phases of workflow management are explained.

This thesis regards workflows and their management from the PAIS perspective (cmp. section 1.1) and Gadatsch [Gad08, p. 60] defines workflow management as “an operative concept for the fulfillment of strategic business process goals. For this purpose the workflow management provides methods and tools for the computer-supported *analysis*, *planning*, *simulation*, *controlling*, and *monitoring* of workflows”.

The analysis phase is not depending on a technology but rather encompasses the manual identification of all necessary resources in a company in order to enable the planning of a workflow. During the planning phase a workflow is designed and specified to a degree that enables its execution. This implementation² of a workflow results in a visual model of a workflow. It may cover the order of tasks in which they execute and the setting of all necessary resources to perform the tasks automatically. The simulated execution ensures that a workflow fulfils all identified requirements of the analysis phase and reduces the risk of costs by executing flawed workflows. The controlling phase accompanies the enactment of a workflow. Enacting creates a specific case from a previous designed process model and executes it [Zur04]. Monitoring the workflow enactment is a technique to determine the current state of workflow cases

²Zur Mühlen [Zur04] splits the planning phase into the steps planning and implementation. Some systems require a translation to an executable workflow representation because the representation used by the workflow modellers is not executable.

concerning the internal tasks, the used or available resources, etc. Monitoring checks the overall integrity of running workflows and initiates escalation procedures if deadlines of the aligned business process are exceeded [Zur04]. Its purpose is to provide information about the current state of workflows to interest parties (e.g. customers that enquire about the status of an order) [Zur04].

The analysis phase is upstream of the other phases. It is linked with the business strategy and necessary for the creation of workflows [Gad08]. Its aim is the identification of valuable business processes with all activities and stakeholders [Fin99]. For this purpose, interviews with different interest groups in an organization are a common method [Fin99]. Even though it is part of workflow management according to Gadatsch³ [Gad08], it is not part of a PAIS because it is performed empirical and not technical.

Simulation prevents a company from unforeseeable costs and monitoring becomes interesting when a lot of workflows are executed simultaneously. Therefore, these two phases are becoming more important when a PAIS is in operation. Then simulation and monitoring tools can be employed on the set of existing workflows and contribute to an enhanced execution of business processes. It does not matter whether some phases are ignored in a PAIS as Gadatsch [Gad08] adds that that scope and type of support workflow management offers is depending on the focussed workflows of the application domain. Since this thesis deals with foundations for a PAIS, the focus of the next sections is on the planning phase and execution control. The planning and implementation phase are also called build-time, while the execution control is considered as the run-time [Zur04]. Before these phases are explained in more detail, the terminology used in PAIS is described in the next section to give a common understanding of the terms.

Workflow Terminology

The following definitions are taken from the Workflow Management Coalition (WfMC) [Wor99]. Since 1993 the WfMC is a global organization of adopters, developers, practitioners, and research groups engaged in Workflow, Business Process Management (BPM). The WfMC develops

³Zur Muehlen does not consider the analysis phase as a part of workflow management (cmp. [Zur04, p. 101]).

process related standards, educates the market on related issues, and is the only standards organization that concentrates solely on business processes⁴. The WfMC defined these terms in 1999 and the definitions are widely acknowledged and still relevant today.

The workflow management phases are performed by a workflow management system (WfMS) that is defined as “a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications” [Wor99, p. 9]. Van der Aalst [Aal98, p. 23] states that “the main purpose of a workflow management system (WfMS) is to support the definition, execution, registration and control of business processes”.

According to the definition of Dumas et al. [DWA05] PAIS emphasize the reuse of process models and a WfMS provides the technology to model and execute process models. However, literature uses many terms to describe process models. Most commonly used terms are: process models, workflow definitions, workflow schemas, workflow templates or workflow prototypes. The WfMC coins the term workflow definition which “consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data” [Wor99, p. 11]. In all cases, a process model describes a modelled workflow which can be interpreted as a template for the execution of concrete workflow instances. Hence, process models serve as blueprints for a set of workflow instances. In the remainder of this thesis the term “workflow prototype” will be used to have a consistent terminology.

A workflow instance is “the representation of a single enactment of a process (specification)” [Wor99, p. 16]. Workflow instances are created and managed by a WfMS in accordance with the available workflow prototypes. Workflow instances are case data because “a workflow instance represents a concrete case in the operational business of a company, consisting of activity instances” [Wes07]. For this reason, workflow instances are sometimes called workflow cases [Zur04].

⁴These information are taken from <http://www.wfmc.org/about-us>.

A workflow normally consists of many activities which are logically related in terms of their contribution to the overall realisation of the goal of the workflow [Wor99]. An activity is “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 resource(s) to support process execution; where human resource is required an activity is allocated to a workflow participant” [Wor99, p. 13]. Today, automated activities often execute online services rather than local applications.

Build-Time and Run-Time

Workflow management is separated in the build-time of a workflow and its run-time [Rei03; Wes07]. The aim of the build-time is the design of a workflow prototype augmented with specifications which are necessary for an automated workflow execution [Gad08]. The modelling of a workflow prototype is finished when it satisfies the requirements imposed by the business process. The run-time enacts the workflow prototype by creating instances of it which can be executed. Workflow prototypes are typically instantiated several times (e.g. for every incoming order or an application for vacation) to ensure that each case is handled the pre-defined way [DWA05]. This process within the workflow management is illustrated in Figure 2.1 and described in the following showing the different roles of a WfMS in these phases.

The build-time starts⁵ with the modelling of a workflow prototype. Therefore, the WfMS provides a tool that enables workflow modelling. The tool is used by a workflow modeller who puts the activities of the workflow in a logical order and fully specifies them. Manual activities may receive a description, a deadline and associated documents and automated activities are properly parameterized to fulfil their function. When workflow modelling is finished and it satisfies all requirements of the involved stakeholders, a new model has been designed. This process model is stored by the WfMS in a workflow repository. The WfMS provides a persistent storage and manages the access to the workflow re-

⁵Weske [Wes07] also regards the analysis phase as part of the build-time while Zur Muehlen [Zur04] does not. During the analysis phase all relevant activities are identified, the persons in charge, and process relevant resources.

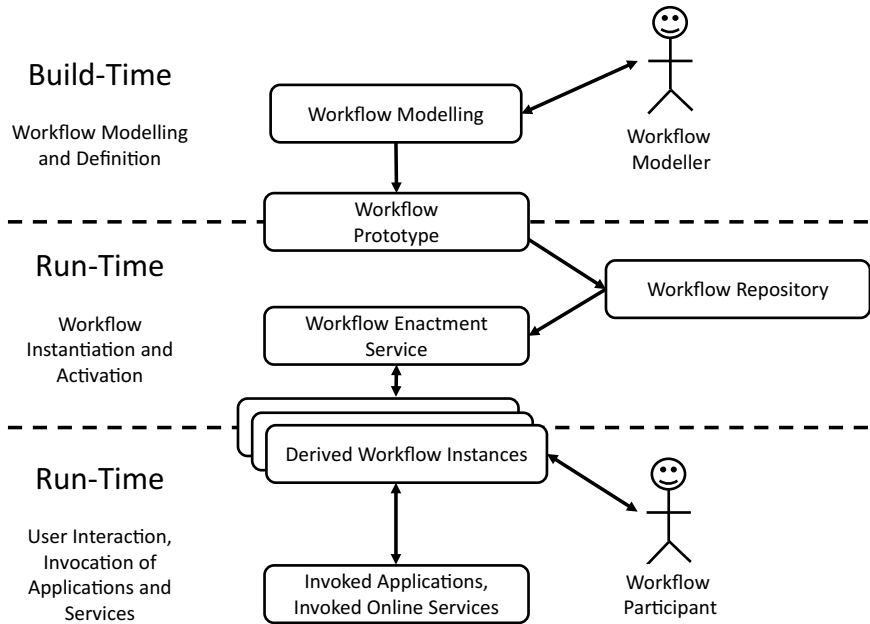


Figure 2.1: Illustration of the *build-time* and *run-time* in workflow management, adapted from [Wor99, p. 44] and [Wes07, p. 308].

pository. If a stored workflow prototype shall be executed for a business case, a workflow instance is created in the course of the workflow enactment. The instantiation has to take care that all modelled information of the workflow prototype are transferred into a machine executable form that can be processed by a WfMS and that all modelled information are correct. Basically, during the instantiation the WfMS creates a machine executable copy of a workflow prototype enriched with all information enabling its automated execution. Concerning the correctness of modelled information, the WfMS may check if all manual activities have an assigned person who is in charge and all automated activities have set their required parameters. Normally, several workflow instances are derived from the same workflow prototype and there is no link between a workflow instance and the prototype that was used to create it [Wes07]. This fact leads to the consequence that changes on a workflow definition do not affect a current execution of a corresponding workflow in-

stance. Whilst a workflow instance is running, the enactment service of the WfMS triggers the activation of the containing activities. If it is a manual activity, the described task is delegated to a person. These persons are called workflow participants and are in charge to perform a certain task. In case that an automated activity is triggered for activation by the enactment service of the WfMS, other external applications or online services might be invoked.

The phases in workflow management are similar to programming. In the application source code all circumstances are codified that can occur and the different operations are put in a strict order which also determines the order of the execution. When everything is “coded” the build-time has finished and the compiler starts the instantiation that translates the programming language in machine executable byte code. Then the run-time begins when the application executes.

After workflow management has been described from a functional point of view, the next section shows how workflow management is technically performed. It describes a reference architecture for WfMS proposed by the Workflow Management Coalition.

2.1.2 Workflow Management System

In 1995 the WfMC proposed a reference architecture for a Workflow Management System [Hol95]. Figure 2.2 shows the reference architecture with the major components and interfaces.

The heart of the architecture is the workflow enactment service already mentioned in Figure 2.1. It provides the run-time environment in which the workflow instantiation, activation, and execution occurs. Although the enactment service is drawn as one logical unit, physically it may be either centralised or functionally distributed⁶ [Hol95]. Thereby, the workflow enactment services utilizes one or more workflow engines. No matter how these workflow engines are implemented (central or decentral), they are responsible for the execution of workflow instances [Zur04]. A workflow engine coordinates the flow of activities by regulating the activation and execution of various activities depending on the state of

⁶The engine metaphor has been criticised by several authors because it suggests that all functionalities which a WfMS offers are build around one monolithic application [Zur04]. Instead, the described functionalities could be supplied by different online services that are locally distributed [Zur04].

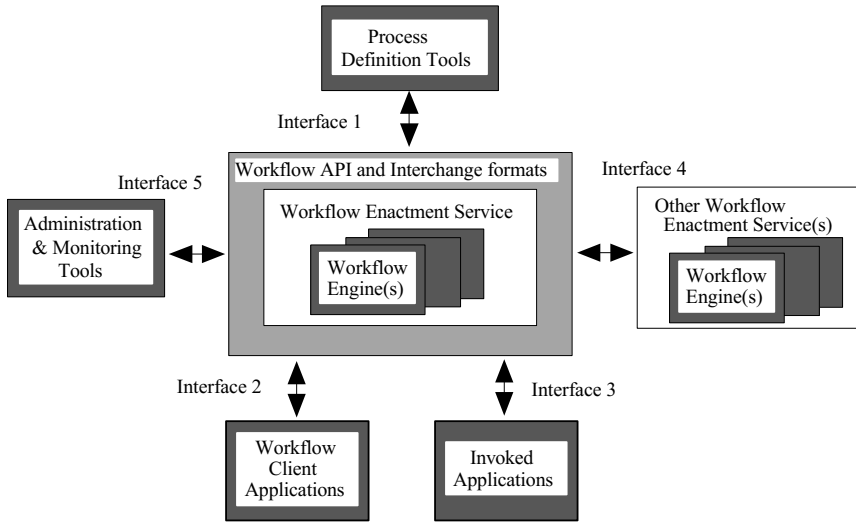


Figure 2.2: The WfMC reference architecture for workflow systems [Hol95, p. 20].

the overall process [Zur04]. The coordination of resource assignments for the activation of activities relates in workflow management to the delegation of tasks to workflow participants and the invocation of external applications and services. The functionality of the enactment service is encapsulated in a workflow API (WAPI) with data interchange formats which handle the interaction between workflow control software and other system components. Thereby, the reference model regards five system components to be part of a WfMS.

Interface 1 describes the modelling and definition tools for the specification of a workflow definition. The workflow modeller (see Figure 2.1) uses a modelling tool to design a workflow prototype. For this task she/he must have a holistic view on the corresponding business process and a deep understanding for each activity which shall become part of the workflow.

Interface 2 connects the workflow client applications to the enactment service. In a workflow instance, interaction occurs between the client application and the workflow engine through a well-defined interface embracing the concept of a worklist. The worklist organizes all work items

in form of a list which are assigned to a certain person. It may contain work items relating to several different active workflow instances of a single workflow prototype and/or from workflow instances from several different workflow prototypes. The worklist connects the workflow participants (see Figure 2.1) to manual activities of a workflow instance. For the workflow participant it is not necessary to regard the whole process. They are the involved resources to fulfil a certain task which has been delegated to them by the modelling of a workflow activity. The worklist provides all necessary information to the assigned person for the task and allows the person to give feedback on the status of a task. When the assigned person finishes a task she/he can notify the workflow engine about the completing of a work item on the worklist and the workflow engine proceeds with the execution of the workflow instance by triggering the activation of the next activity.

Interface 3 is for the invocation of applications. It is the equivalent of interface 2 for the execution of automatic activities. In the original description of this interface it is stated that “it may be assumed that any particular WFM implementation will not have sufficient logic to understand how to invoke all potential applications which might exist in an heterogeneous product environment. This would require the logic to cope with invocation across (potentially) all platform and network environments, together with a means of transferring application or workflow relevant data in a common format and encoding” [Hol95, p. 35]. However, in 1995 technologies that foster the data interchange between workflows were not advanced enough for a broad application and online services were quite unknown. Nowadays, in business workflows, web services are used to invoke services that can be linked and compute distributed (see section 2.1.5).

Interface 4 provides communication means to other workflow enactment services. In its drive for interoperability, the WfMC focussed on developing a variety of interoperability scenarios in which different workflow engines of different vendors can be combined. Interface 5 is for the administration and monitoring of workflows.

This section and section 2.1.1 have given a survey on workflow management from a functional and a technical perspective. Of course, there are also different opinions about the term “workflow management system” and how the architectures of a workflow management system should look

like (cmp. [Jab95; GG99]). The next section deals with the question how workflows are modelled in general. This refers to interface 1 of the WfMC reference architecture and the build-time in workflow management.

2.1.3 Workflow Modelling Perspectives

Rivard et al. put it simple when they say that “the best way of describing business processes is to model them” [RAM09, p. 134]. Until now, the reader was only given an understanding what workflow management is and how it is supported by a WfMS. This section is about the design of workflows and which perspectives have to be considered for their modelling. In traditional BPM, the design of a workflow often regards three perspectives [Sch00; Öst95]: an organisational, a functional, and a data perspective. The organisational perspective regards the hierarchies which exist in a company. Therefore, the structure and the relationships of business employees is captured resulting in a diagram to visualize the expertises and competences in the organisational hierarchy [Gad08]. The functional perspective describes the business activities and their relations in a structured manner to determine how business goals are supported by business processes [Gad08]. The data perspective regards structured information in a company with the aim to create a conceptual data model that gives a detailed description of all data objects occuring in the business processes [Gad08]. Data dependencies which may occur are captured and data is annotated with specific terms in order to create an organisational data model [Gad08]. Unified Modeling Language (UML) class diagrams or Entity-Relationship-Modell (ERM) diagrams can be employed to visualize their dependencies [Sch00].

Since the target group of this thesis are private groups and individuals who voluntary participate in an online community for social workflows, the traditional organisational view is of no account for this thesis. There is no predefined organization with departments and roles in an online community. The same applies to the traditional data view. Although, data (e.g. documents and media files) in online communities is passed between community members, it is unstructured and usually not related to each other. Hence, data in online communities is manageable but concerning the modelling of workflows nothing needs to be considered. For this reason, only the functional perspective is regarded in this section.

To describe activities of a workflow in a structured manner, they can be presented in various ways and notations. The notation is based on the workflow modelling language⁷ which is provided by a workflow modelling tool.

In the following, selected control-flow patterns, data patterns, and resource patterns are described [Aal+03; Rus+05a; Rus+05b]. These patterns are not notation-specific, but describe generic concepts for workflow modelling⁸. A process model must have at least information about activities to be executed, relationships between activities (control-flow and data patterns), and conditions that enable activities and resources for the task execution (resource patterns) [Obe05]. In the following the relevant patterns for this thesis are introduced.

Selected Control-Flow Patterns

The control-flow describes the temporal and logical progression of activities [Sch11, p. 55]. Therefore, the explicit representation of the control-flow can be visualized as a graph. Aalst et al. [Aal+03] identified 20 notation-independent control-flow patterns. Figure 2.3 shows some basic control-flow patterns.

The *sequence* (pattern 1) is a directed edge that defines the execution order of activities. An activity in a workflow is enabled after the completion of another activity in the same sequence.

The *parallel execution* of activities is expressed by the parallel split pattern (also “and-split”, pattern 2) and the synchronization pattern (also “and-join”, pattern 3). The parallel split is a point in the workflow where a single thread of control splits into multiple threads of control which execute in parallel, thus allowing activities to be executed in an arbitrary order but also in parallel. The synchronization pattern is a point in the workflow where multiple threads of control converge into a single thread of control.

⁷The notation of the workflow has not to be in a visual form (“diagram based”, cmp. [Gad08, p. 81]) but can also be noted in a declarative manner [WP06]. In this thesis only diagram-based descriptions of workflows are considered because declarative descriptions (e.g. in first order logic) are not intuitive for private individuals.

⁸They were developed by the *Workflow Patterns Initiative* (WPI) which started in 1999 with Wil van der Aalst and Arthur ter Hofstede when they researched on workflow patterns.

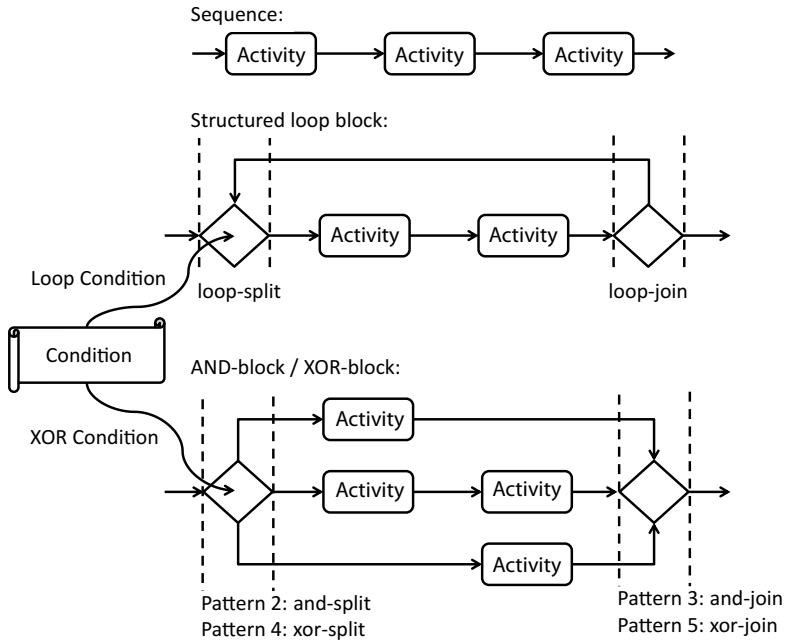


Figure 2.3: The control-flow patterns of sequence, loop, parallel, and exclusive execution.

The *exclusive execution* of activities is expressed by the exclusive choice pattern (also “xor-split”, pattern 4) and the simple merge pattern (also “xor-join”, pattern 5). The exclusive choice is a point in the workflow where a single thread of control forks into multiple threads of control which must not execute in parallel. The xor-split selects one following thread of control based on a decision or workflow control data. The xor-join pattern merges several threads of control with the assumption that only one has been executed. Hence, the xor-join is no synchronizing pattern.

For the *iteration of activities*, structured cycles can be expressed with the loop pattern (pattern 10). The structured loop pattern is a restricted version of pattern 10 which omits arbitrary cycles. Corresponding to the parallel and exclusive execution of activities, the iterative execution of a structured loop starts with a “loop-split” and completes with a “loop-

join”. It executes a set of activities in a specific order until a specific condition, based on a decision or workflow control data, terminates the iterative execution.

Figure 2.3 shows a block-oriented structure for control-flow patterns. The representation of branches as blocks leads to a symmetric structure of the control-flow because the order of split- and join-nodes is preserved [Rei00]. Blocks cannot be interleaved but they can be nested. The execution of sequences within exclusive or loop blocks requires evaluating a condition. Defined conditions are mandatory for a workflow engine to select the proper sequence after the execution of a xor-split and to compute the number of iterations in a loop block. Thereby, the conditions can be based on functions or events that concern workflow data. For the xor-split the modelled condition must ensure that at least one sequence evaluates to true [Wes07].

Selected Data Patterns

In graph-based workflow modelling languages, data dependencies are represented by data-flow between activities [Wes07]. Each activity is assigned a set of input and a set of output parameters. Upon its start, an activity reads its input parameters, and upon its completion it writes produced data to its output parameters. These input and output parameters are workflow data items and the transfer of data between activities is known as data-flow. For an automated activity, the input parameters configure the execution of the online service. For a manual activity, the input parameters provide all information a person receives for an assigned task. To enable a WfMS to manage data items and to interpret its content, the data items must be typed. For instance, the workflow engine cannot evaluate a condition that is based on a data item if it is not able to interpret the content of the data item in order to make a decision.

Russell et al. [Rus+05a] identify 40 data patterns and categorize them in four main categories: data visibility, data interaction, data transfer, and data-based routing patterns. The relevant patterns for social workflows among each category are:

- *Data visibility*: It is comparable to the scope of variables in programming languages because it characterizes the area in which a data item

is available for access [Wes07]. Typically, data items in a workflow are bound to a specific activity or a control-flow structure that requires a condition for executing (exclusive and loop blocks). Tasks data (pattern 1) are elements which are accessible only within the context of a single activity. Case data (pattern 5) are specific for a workflow instance or in other words for a case of a workflow prototype. They can be accessed by all components of the workflow during its execution. Produced data that is a result of the execution of a workflow instance is case-specific⁹. Folder data (pattern 4) allows to put data objects in folders and access these folders by multiple workflow instances. Environmental data (pattern 7) describes data elements which exist in the external operating environment of the WfMS. External data may be sourced from different locations [Rus+05a] including cloud services which mediate access to external data repositories.

- *Data interaction*: Data can be passed in several ways between the components of a workflow. The relevant data interaction pattern for social workflows is data interaction between activities. Task-to-task (pattern 8) interaction means the ability to communicate data items between one activity and another activity in the same workflow. Task-to-task interaction is normally visualized by the data-flow.
- *Data transfer*: The manner in which data interaction occurs. It can be seen as an extension to data interaction and aims to capture the various mechanisms by which data items can be passed in a WfMS. In general, the data transfer patterns distinguish between “call-by-value” and “call-by-reference” for data items.
- *Data-based routing*: Data-based routing itself (pattern 40) describes the ability of data to alter the control flow within a workflow instance as a consequence of the evaluation of conditions that contain data. As mentioned before, the exclusive execution of forked sequences and the loop are depending on this pattern. Additionally, there are data patterns which are originally intended to decide if single activities may be executed but can also be used for data-based routing in conditions for iterative and exclusive control-flow structures: The data existence

⁹In contrast to *workflow data* (pattern 6) case data is not visible between different instances of a workflow prototype.

(pattern 33) checks for the existence of a data item as precondition. The data value (pattern 34) checks data items according to its value¹⁰.

Resource Patterns

The resource perspective regards how resources can be used in PAIS to perform certain activities and their interaction with the PAIS [Rus+05b]. “A resource is an entity that is capable of doing work and can be classified as either human or non-human, i.e., a resource that does not correspond to an actual person – e.g. plant and equipment.” [Rus+05b, p. 219]. The resource patterns of Russell et al. [Rus+05b] do not focus on the allocation of non-human resources but on the assignment of humans to work items in the execution of workflows.

The simplest task assignment is the direct allocation (pattern 1). It enables a workflow modeller to set a concrete identity of a resource for the fulfilment of a task. This is particularly beneficial if tasks are critical as it prevents the problem of unexpected or wrong resource allocations during the run-time of the workflow.

The role-based allocation (pattern 2) takes organizational roles into account when resources are allocated to tasks. A resource can have one or more associated roles. Roles are groups of resources that comply with a similar job specification or responsibility levels (e.g. secretaries, managers, etc.). Roles can also be identified by common attributes of the containing resources. Thus, the attributes represent certain capabilities which are shared by the resources of the same role. The role-based allocation enables the workflow designer to specify at build-time that a task can be executed by resources which correspond to a given role. During the run-time the workflow engine will route the work items to a suitably qualified person. The advantage of the role-based allocation is that it segregates the assignment of a specific identity from the design of the task. The selection of a qualified resource is performed according to the profile of the selected role.

The delegation (pattern 27) provides a resource with a means to re-assign work items in case that it is unable to execute. A defined delegation is mandatory if a resource is unavailable (e.g. due to disease) or her/his workload is too high already.

¹⁰In such case, it requires data to be typed in order to allow an interpretation of the value.

The distribution by offer (pattern 13) is a means to advise a group of persons that a work item exists and it is committed to these (human) resources who undertake the activity. In contrast to the other resource allocation patterns, each person who is offered the task sees the work item on its worklist until one of them commits to starting it and thereby charging the workflow engine to exclusively assign the work item to this person. In the following of this thesis the behaviour of pattern 13 is denoted as the act of “bidding”.

2.1.4 Adaptive Workflow Management

Traditional PAIS are constantly reaching their limits to handle unforeseeable events and imponderabilia during the build-time of a workflow [Sch11]. This especially counts for the ability to react on unforeseen events when a workflow instance is already started and its activities are executing. The effect amplifies with the duration of a workflow and the number of workflow participants. With the increasing duration of a workflow it is hardly possible to plan all unexpected events during the build-time of a workflow [MTS08]. There are many application domains which require a flexible management of workflow instances [Sch11; RW12]. For such domains, adaptive workflow management [RD98; Sch11] is a means to interweave the build- and run-time of a workflow aiming to couple the two phases.

Schmalen [Sch11, p. 66] defines an adaptive WfMS (aWfMS) as a “process-oriented information system which enables the modelling of process templates for adaptive processes and the control and execution on the corresponding process instances. Such a system is characterized by its methods for managing unexpected exceptions that cause the adaptation of control-flow and data-flow of a process instance to new requirements”. In general, flexible methods of PAIS can be characterized by four major flexibility needs, namely support for *variability*, *looseness*, *evolution*, and *adaptation* [RW12]. These different methods for flexibility are described in the next paragraph. Then, it is explained how they can be of value for a social workflow platform.

Workflow variants typically share the same core and hence pursue the same goal whereas the concrete course of action fluctuates from variant to variant [RW12]. It is the simplest form of flexible methods because a set of predesigned workflow variants are provisioned. Thus, it is depend-

ing on an accurate and detailed planning of a workflow, anticipating all upcoming events. Looseness means that there are workflows for which only their goal is known a priori but the course of actions is highly unpredictable [RW12]. Hence, these workflows cannot be fully prespecified and each executed workflow instance may vary in its course. For this case, loosely specified workflows can unfold parts of the workflow and replace these parts with alternative activities or workflows. Evolution¹¹ represents the ability of a PAIS to change workflows when the corresponding business process evolves [RW12]. This means that changes on a workflow prototype lead to a migration of all derived workflow instances. Thereby, the WfMS must decide which changes may propagate to the instances and which may not [RD03]. Adaptation represents the ability of a PAIS to adapt the structure of a workflow to emerging events [RW12]. Adaptation is the method for special situations during workflow execution which have not been foreseen in the design of the workflow.

Methods for performing a structural adaptation on a running workflow instance provide the highest degree of flexibility because adaptation does not restrict changes to predesigned templates. Adaptive workflow technology is regarded to be crucial to enable a broader range of application areas for PAIS [RD03].

An insertion of not planned activities, deletion of not started activities, changing the execution order of activities by moving them, resetting and skipping of activities, are examples for workflow adaptation [Rei00]. For manual activities, adaptation also includes changing the assigned person, and in general to modify the properties of an activity that influence its execution. Such “ad hoc” changes on a running workflow instance can be realized with the breakpoint concept of Minor and Schmalen [Min+07; Sch11]. A breakpoint is a workflow element which can be put in the control-flow during the run-time of a workflow. As a consequence, all workflow elements (activities and control-flow structures) that are located behind the breakpoint in a sequence are suspended and thereby protected against the execution of the workflow engine. The partial suspension of a workflow then allows ad hoc changes in the protected area of the workflow. When all desired changes have been performed, the removal of a breakpoint triggers the workflow engine to continue the execution of the control-flow.

¹¹It is also called schema evolution.

It is important that every structural operation (insertion, removal, or moving of workflow elements) leads from a consistent state to a new consistent state. Dadam et al. [Dad+09] emphasize this importance of the syntactical consistency of a workflow during ad hoc changes. Otherwise, the workflow engine is not able to proceed with the workflow execution. A principle to ensure consistency is the *correctness-by-construction* method [RD98; Rei00]. The workflow modelling language for social workflows follows this principle due to its block-orientation (cmp. 2.1.3 Figure 2.3). Block-orientation enables an easier and safer manipulation of workflows [Dad+09]. It leads to another approach how workflow modellers start to design a workflow. Instead of connecting a set of nodes and edges in an arbitrary order, the modellers can only insert consistent blocks [Dad+09]. The execution of a workflow is depending on its structural consistency and a WfMS may omit consistency checks if it provides a block-oriented workflow modelling language.

Ad hoc changes on running workflow instances involve the workflow elements as well as the workflow data and data dependencies of workflow elements. Such dependencies exist for data-based routing and data interaction between activities. Schmalen [Sch11] describes comprehensive rules for altering workflow data during ad-hoc changes¹². His data consistency rules are for data with concrete values but also for references which might point to data values outside of the current workflow. Schmalen distinguishes between¹³:

- ad hoc changes on data that is in the scope of a workflow instance and has no dependencies to workflow elements and
- ad hoc changes on data with dependencies to workflow elements.

¹²Sadiq et al. [Sad+04] only regard correctness in the build-time of a workflow. The “Data-Flow Anti-Patterns” of Trčka et al. [TWS09] also mainly regard build-time correctness. Their “missing data” anti-pattern is applicable to the run-time of a workflow but it less specific than the rules by Schmalen [Sch11] omitting data-based routing. Rinderle-Ma [Rin09] suggest a trace-based correctness check (by capturing all read-write operations on a data object) and a theorem-based correctness. However, the proposed theorem does not regard data-based routing but only data interaction with activities. The trace-based correctness check takes too much computational effort [Rin09].

¹³Schmalen developed a technical reference model for flexible WfMS. Consistency rules that result from the technical aspects from the reference architecture are ignored.

If no dependencies to workflow elements exist then adding, deleting, and modifying of data elements is allowed, because the data elements do not interfere with the control-flow of the workflow. Otherwise changes on data elements are valid if changes on their associated activities are allowed.

Adaptive workflow technology helps to overcome the rigid restrictions of traditional workflow management by allowing a flexible execution of processes. As a consequence it is not required that all information are on hand when the workflow is designed at the build-time. Instead, a running workflow can be altered when a certain information becomes available at a later point. There are several forms how this flexibility can be achieved and in section 3.5 the reader is given an explanation which flexible methods shall be supported by a social workflow platform. Later on, section 4.3 presents in detail how social workflows can be modelled and altered during their execution.

2.1.5 Web Services

PAIS use WfMS to automatically execute workflows involving human executors as well as applications and external systems. This segmentation of a business process in activities enables a dynamic arrangement of their composition. The business logic is maintained in the modelled sequence of activities of a workflow. Thereby, enterprises get a modular view on their business processes and become more flexible in case that parts of the workflow must be changed. The idea of a modular construction of a business process is of particular importance for the execution of automated activities. Since automated activities do not need human interaction to be automatically executed, there are efforts to describe them according to their specific functionality in order to make them interchangeable in the execution of a workflow. With such a modularization of automatically executed activities within a workflow, a business process can be detached from an existing IT-infrastructure. From a business perspective this allows a company to outsource expensive IT-costs and likewise to become independent of proprietary IT-systems. To enable such an interchangeability of automated activities, it is required that the necessary systems are accessible via the Internet and an interface description for the communication. While a web-based access can

be assumed for a PAIS the interface description is in the focus. This section describes what kind of interface description is used nowadays in order to invoke services and which approaches exist to link services invoked this way.

For the description of service interfaces, the “Web Service Description Language” (WSDL) by the W3C¹⁴ is established. This interface description is a generic specification of the available operations of a service, the required input data, and the produced output data by the service. “Web services provide a standardized way to achieve interoperability between heterogeneous software systems independently from underlying implementation technologies and platforms” [Hat+12, p. 319]. The first WSDL standard in version 1.1 was initiated by IBM, Microsoft, and Ariba [Chr+01]. Today, the current version is 2.0 and a W3C recommendation [BL07] and hence a web standard. The structure of a WSDL interface description is described in XML syntax and basically consists of a concrete and an abstract, reusable part [RS04b].

In the abstract part, each service operation defined is a sequence of input and output messages [Ber+10]. Therefore, the abstract part defines the message structure which is used for the communication, data types of the messages, and the sequence of the messages [HKI06]. So-called message exchange patterns can be associated to one or more messages. The exchange patterns clarify the type of communication between two parties. Basically, the message exchange patterns differentiate the direction of the communication (unilateral and bilateral) and the origin that initiated the communication. This allows complex communication processes as well as simple event-based notifications between two parties. The concrete part of a WSDL file defines a binding of the abstract operations to a concrete network endpoint [HKI06]. This includes the network address of the endpoint and the selection of the communication protocols (e.g. SMTP or HTTP) supported by the services.

In the past, web services were manually invoked in proprietary source code and their processing results were programmatically linked to downstream invoked web services [RS04b]. Such a static linkage omits the advantages of web services because all link changes must be performed

¹⁴W3C stands for “World Wide Web Consortium”. The W3C was invented by Tim Berners-Lee and works on the development of web standards. For more information refer to <http://www.w3.org>.

by a software engineer. Moreover, a programmatical inclusion of a web service inhibits the dynamic invocation which web services offer. Using WSDL as interface description language supersedes the implementation of an interface for each web service. According to the interface description a generic implementation can invoke and communicate with a web service. To overcome such a static linkage of web services, WfMS are obviously beneficial because the sequence of the web services and their linkage can be re-designed in a workflow. The linkage of web services is called the *service composition* [RS04b]. In this area two terms have emerged which are closely related but describe well distinguished concepts [Pel03]: orchestration and choreography. The difference between them is the point of view on business processes.

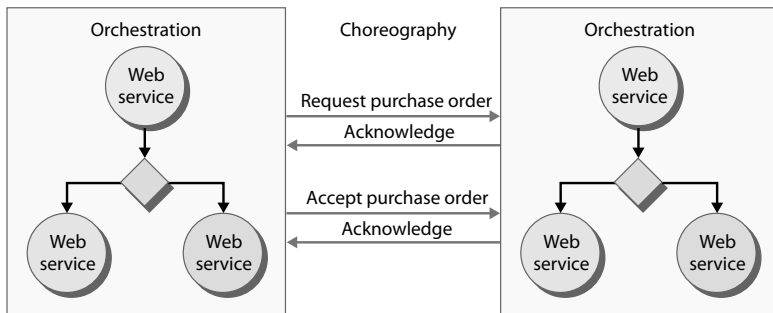


Figure 2.4: The differences between orchestration and choreography. *Orchestration* refers to a workflow. *Choreography* tracks the message interaction between parties. Taken from [Pel03, p. 47].

“*Orchestration* refers to an executable business process that can interact with both internal and external web services. The interactions occur at the message level. They include business logic and task execution order, and they can span applications and organizations to define [...] process models” [Pel03, p. 47]. In this sense, the orchestration of web services can be considered as a workflow with its business logic represented in a control-flow. However, this view on control is always from one party’s perspective [Pel03]. This differs from *choreography* which

tracks the message sequences among multiple parties [Pel03]. The choreography is more collaborative and allows each involved party to describe its part of interaction [Pel03]. The differences between both are illustrated in Figure 2.4.

It is evident that the choreography describes business processes from a top view in a peer-to-peer collaboration [HH10]. The orchestration of web services corresponds to the internal view of a company on its business processes while the choreography describes the public message exchange between companies/parties [RS04b]. Thus, the choreography can set the valid sequence of messages between the invoked web services but the internal workflow execution remains hidden. Sometimes the orchestration of web services is understood as the enactment of web services [RS04b].

For PAIS the orchestration is more important because the enactment of workflow instances is part of common workflow management. To invoke and orchestrate web services in a workflow, a PAIS requires a workflow execution language that supports these mechanisms, namely to understand WSDL files, to put them in a defined order, and to link their input and outputs. For this purpose the OASIS¹⁵ (Organization for the Advancement of Structured Information Standards) approved the WS-BPEL 2.0 specification¹⁶ (Web Service-Business Process Execution Language)¹⁷ in 2007 [OAS07]. OASIS is a not-for-profit, international consortium that drives the development, convergence and adoption of open standards. WS-BPEL 2.0 – often shortened to BPEL – was driven by Microsoft and IBM to merge their proprietary web service orchestration languages. In contrast to the human-centric view of the WfMC on workflows, WS-BPEL has a process-centric view on workflows and does not regard the inclusion of workflow participants¹⁸. Today, WS-BPEL

¹⁵For more information please refer to <http://www.oasis-open.org>.

¹⁶It should be noted that WS-BPEL 2.0 is incompatible with WSDL 2.0. It can only be used with the WSDL 1.1 specification. Therefore WS-BPEL 2.0 cannot make use of the technical features of WSDL 2.0 which include full support of the HTTP protocol [OAS; Ley03].

¹⁷In former versions it was known as BPEL and BPEL4WS.

¹⁸The OASIS started a technical committee (BPEL4People) for the inclusion of humans in the execution of WS-BPEL but until today this technical committee presented no results (<https://www.oasis-open.org/committees/bpel4people/>). Russell and van der Aalst [RW08] note that BPEL4People is missing a user interface description that corresponds to the task delegation on a worklist.

is seen as the industry standard when it comes to web service orchestration [RR06]. It is used to orchestrate the functionalities exposed through web services into new composite applications [JW14].

However, the invocation of internal and external web services in the execution of a workflow instance using WS-BPEL is a challenging task considering occurring service exceptions and rapidly-changing workflows [WN10; Kar+05; CM04; RR06; RW12]. To solve this problem Wolff and Singh [WN10] propose to enrich WS-BPEL with semantic technologies and business rules. Therefore, they employ OWL-S¹⁹ as ontology language for web services. Karastoyanova et al. [Kar+05] address the problem of dynamic binding of web services to workflow instances. Basically, they perform a late-binding of web services according to a controlled policy-based selection. Charfi and Mezini [CM04] propose to combine the execution of WS-BPEL with the Aspect-Oriented-Programming (AOP) paradigm. Thereby, an “aspect manager” should be linked to a WS-BPEL enactment service and enrich the WS-BPEL code with “before” and “after” conditions for activities that must comply to business rules²⁰. As a conclusion, these approaches try to enhance WS-BPEL with semantic constraints or rules to increase the flexibility of WS-BPEL. A more general investigation on the possible flexibility of WS-BPEL is done by Reichert, Rinderle, and Weber [RR06; RW12] according to flexible methods for PAIS.

In 2006 Reichert and Rinderle [RR06] investigated how far schema evolution and ad hoc changes can be applied on WS-BPEL instances. They focus on the question how such changes can be realized in a correct and consistent manner on a subset of WS-BPEL elements. In 2012 Reichert and Weber [RW12] state that WS-BPEL provides exception handling patterns for anticipated exceptions but not for unanticipated exceptions of single workflow instances. This lack of flexibility of WS-BPEL leads to other approaches for web service composition that do not rely on workflow technology. To enable a dynamic service composition methods from Artificial Intelligence, especially from planning, are often employed [RS04a].

¹⁹OWL-S has been submitted to the W3C in 2004 (see also <http://www.w3.org/Submission/OWL-S>). Its development was driven by the DARPA and is formerly known as DAML-S.

²⁰AOP is further described in section 5.2.

Another approach using workflow technology is described by Schmalen [Sch11]. He describes a technical reference architecture for the execution of agile workflows. Similar to the proposed reference architecture of the WfMC (see section 2.1.2), the technical reference architecture of Schmalen includes humans as well as external applications like services in the execution of a workflow. Thus, his developed workflow execution language SAW²¹ is able to cope with agile methods for adaptation and to wrap service interface descriptions. However, SAW is not intended for orchestrating web services. It is able to read WSDL descriptions and it describes the invocation of web services. Thereby, SAW restricts the invocation of web services on the message exchange patterns initiated by the automated activity. Hence, it is not able to react on external notifications like WS-BPEL.

As a concluding remark the author of this thesis agrees to Bertoni et al. [Ber+10] as they state the concept of web services is not a new architecture or technology. Web services "can be considered, to a large extent, the point of convergence of the evolution of several ICT technologies [...], intertwined with an increasing adoption of standards" [Ber+10, p. 9]. For a PAIS this means that the selection of the workflow execution language is depending on the application domain and the current available web standards. WS-BPEL has no official support for human activities and has no constructs for unexpected ad hoc changes. However, it is the industry standard for fully automated workflows in which only systems are invoked and orchestrated. Contrary, workflow execution languages like SAW may focus on an adaptive workflow execution with a web service invocation.

2.1.6 Conclusion

In this section PAIS were introduced which come from the business world. This is of immediate relevance for the course of the thesis because social workflows avail themselves of existing structures and methods from workflow management. Later on, in section 4.3 a modelling language for social workflows is presented. It uses a control-flow to determine the sequence of activities in a social workflow. The separation of build- and run-time in traditional WfMS exists to guarantee the repeatability of

²¹SAW is an abbreviation for the German words "Sprache für adaptive Workflows".

business processes. The workflow prototypes created during the build-time are templates for business cases in order to avoid failures in the daily business. The application of social workflows is probably not subject to such static restrictions. Online services which may be included in a social workflow are not assumed to work as expected all the time and individuals are no process modelling experts expected to plan all details of a social workflow before its execution. For these reasons, the social workflow platform includes methods of adaptive workflow management and for the invocation of services.

2.2 Online Communities

Social Workflows are for private individuals and the technical concepts for a social workflow platform shall make it possible that a community for social workflows evolves. Primarily, this thesis is about these technical concepts that allow to build the infrastructure for social workflows: to design, share, and automatically execute social workflows. However, in contrast to business processes with their underlying business workflows, a social workflow community is based on voluntary participation of its members. In an enterprise's organisation the employees are included in the business workflows in order to perform their work in the best experienced manner. In a social workflow community people are driven by other motivational factors than monetary rewards or career advancements. For this reason, it is likely that technology is not the only enabler of a social workflow community. The incitements of users in online communities are complex and research on motivational factors for participation resides in sociological and psychosocial studies²².

Nevertheless, there is no doubt that people use online communities to discover new knowledge and to share their knowledge with the community. Wikipedia is a well known example for such user behaviour. Social network sites – as a new form of online communities – offer additional media sharing features and thereby provide a virtual space for friends and acquainted people for exchanging news and contents. The

²²Literature on established online communities and social networks and thus on platforms with functionality enabling the growth of a body of experiential knowledge emphasize sociological or psychological aspects and normally leave out technical aspects that are in the focus of this thesis [Sta02].

integration of social network members in the execution of a social workflow shall help to easily delegate tasks to friends. While the execution will benefit from the existing interpersonal relationships, the technical concepts for the platform allow a social workflow community to grow and to create a new form of knowledge management for private individuals.

To deal with this matter, online communities and social networks are regarded closer in this section and delimited from each other. There are structural characteristics which are specific either for online communities or for social networks and a social workflow platform can exploit both characteristics. A social workflow platform shall provide benefit to its users and therefore it is necessary to understand why people tend to work collaboratively and take efforts for an online community. For this reason, in the end of this section, generally accepted sociological theories are described which deliver motivational factors for participation in communities.

2.2.1 Aspects of Online Communities

Since the beginning of the Internet, private individuals use it to get information and to contribute their specific knowledge to communities. There have been many software tools to create online communities and to foster collaboration: bulletin boards, mailing lists, forums, the Internet Relay Chat (IRC) or Instant Messaging (IM). All these tools are means to collaboratively solve problems or to disseminate new knowledge. If someone asks a question about a specific topic in a discussion thread, other people provide solution steps to solve the problem. In some online communities, the members are eager to provide tutorials to a community. Examples are cooking communities (providing different categories of recipes), automotive forums (providing repair solutions or tuning tips), computer forums (providing tweaks for operation systems), or beauty forums (providing makeup tutorials). All these online communities have in common that they provide processes to solve specific problems in a textual manner.

For many people the emergence of online communities goes along with the Web 2.0. This term was coined by Tim O'Reilly in 2005 [ORe05] and is often used to characterize sites that consist of communities [IMP09]. Among these, social networks (e.g. Facebook or Google+), video streaming platforms (e.g. Youtube or Dailymotion), Wikis (e.g. Wikipedia),

and other sites foster user collaboration and/or information exchange [IMP09]. Literature on the Web 2.0 reinforces the idea that it did not emerge as a consequence of new available technology. Instead, it initiated a new way of Internet usage for the people [IMP09] by shifting the Internet to a service platform for its users [ORe05].

Long before the term Web 2.0 occurred first, one of the first definitions [Pre01] for online communities is by Howard Rheingold [Rhe93] when he wrote²³ that “online communities are cultural aggregations that emerge when enough people bump into each other often enough in cyberspace. An online community is a group of people who may or may not meet one another face-to-face, and who exchange words and ideas through the mediation of computer bulletin boards and networks” [Rhe93, p. 57f]. It is obvious that this definition remains quite vague and “antique” compared to the today’s common understanding of an online community.

Ten years later in 2004, Plant [Pla04, p. 54] defined online communities as “a collective group of entities, individuals or organizations that come together either temporarily or permanently through an electronic medium to interact in a common problem or interest space”. This definition is more concrete than Rheingold’s one. Most important is that the scope of the definition is hereby extended from the tools (e.g. computer bulletin boards) towards a common problem and interest space.

In 2009 Buss and Strauss [BS09] point to the same direction with additional focus on the relationships between the users of an online community. They state that online communities are “a group of people who regularly interact with each other on a website. As in other kinds of communities, online community members are united by a common activity or interest; they can get to know who the other members are, relationships form, and a sense of connectedness develops” [BS09, p. 4]. Following this definition, online communities would require web technology and would be part of the Web 2.0 development, but this is an ambiguous description. In fact, online communities are older than the Web 2.0. First online communities evolved without “enabling technologies” of the Web 2.0: Newsgroups, the Internet Relay Chat (IRC), and also online games are examples for virtual spaces in which people form online communities without web technology. Otherwise, you could assert that meanwhile every web page is a latent community [Shi08].

²³Rheingold talked of “virtual communities” instead of “online communities”. In this thesis these two terms are not distinguished.

Online community members made a commitment to achieve shared goals and towards achieving the goals they users agree to cooperate through a shared set of means [How10]. Additionally, online communities tend to be much more structured, have more rules, and rituals than social networks. This stronger structuring of online communities is regarded in a frequently used definition of online communities by Jenny Preece [Pre00]. Preece thinks about online communities from an operational perspective and defines them as “a group of *people* with a common *purpose* whose interaction is mediated and supported by *computer systems*, and governed by formal and informal *policies*” [Pre00, p. 16]. Of course, scratching on the surface of the four highlighted aspects – people, purpose, computer systems and policies – reveals a much more complex set of attributes [Pre00]. For instance, people in online communities have identities that are determined by how and what they reveal about themselves, which may be influenced by the design of the software, their expertise in using it, or by their motivation. However, depending on the individual perspective and academic discipline (technical, sociological, or psychological), the different characteristics of online communities take on different levels of importance in the definitions [PM03]. For this reason, there are several views about what an online community is and there is no single accepted definition of “online community”. The term means different things to different people [Pre00].

Online communities can achieve the kind of organizational structures needed for both cooperation and collaboration (collective action); social networks can’t [How10]. In contrast to an online community, the organizational structure in a social network focuses around an individual user’s one-to-one relationships [How10]. These interpersonal ties in (online) communities have been subject of research since the 1960s. In 2005 the sociologist Mark Granovetter [Gra05] summarized his decades lasting work on interpersonal ties. He regards interpersonal ties as information-carrying connections between people and he categorizes ties in two variants: strong and weak ties. Tie strength characterizes the closeness and interaction frequency of a relationship between two parties [LC04]. An example for a strong tie would be a parent-child relationship whereas a weak ties describe connections between acquaintances. Interpersonal ties are the connections between people in an online community. These relationships in social networks are fundamentally different than in on-

line communities [How10]. In social networks such as Facebook, the individual user is at the center of the network. In online communities, the relationship between the users are much stronger, even though they have never met [How10; Shi08].

Communities are not only a collection of individuals; Communities are often described by their focus of interest expressed by the shared goals [Pre04]. This results in stronger secondary interpersonal ties and allows to engage in more complex tasks in online communities than in social networks. Shirky [Shi08] puts different types of tasks that occur in an online community in a typology of activities:

- Sharing
- Cooperation
- Collective action / Collaboration

Sharing requires the least amount of organizational complexity. Knowledge sharing with others is the simplest way to take advantage of online communities [Shi08]. *Cooperation* is the next level and it is harder than simply sharing, because the activities must be synchronized with other people. Unlike a sharing community, where the group is mainly an aggregate of users, cooperative activities create a group identity [Shi08]. One simple form of cooperation is conversation; In an online community, the conversation is often virtually and several tools support this: Instant Messaging, e-Mails, or by other media. Conversation creates more of a sense of community than sharing does [Shi08]. *Collaboration* is a more involved form of cooperation, as it includes aspects that are not part of normal cooperation. With cooperation, small teams work on their own goals and not on (global) community goals. For collaboration some collective decisions have to be made and a person does often not get credit for what she/he has been created [Shi08].

The next section takes a closer look at social networks to illustrate the contrast to online communities. Social networks are a subclass of online communities with a special interpersonal tie structure focussing more on the individual than on group interests.

2.2.2 Social Network Sites

Research on social networks is a branch of sociology that examines the patterns and characteristics of social connections and their relationship to individual's lives and societal organization [PM03]. Social network theory investigates how people relate to each other through computer-mediated networks. Wellman [Wel99; Boa+06] has shown how the shift from solidary communities to social networks began before the Internet and the Web 2.0. Event though, the Internet has made it easy for people to connect without living nearby and without knowing each other well. Wellman and Frank [WF01] believe that a multi-level approach is required to understand the interactive effects in social networks and their characteristics. These characteristics include i.a. the composition of networks (its structure), the interpersonal ties (frequency of contacts between people and the density of their connection), and the resources available in the network [WF01]. However, these characteristics are rather attributes with a certain value and thereby identify a social network. For a definition of social networks the functional aspects are more interesting and a common definition for social networks is by boyd and Ellison²⁴[BE08]. They define [BE08, p. 211] social network sites as web-based services that allow individuals to:

1. construct a public or semi-public profile within a bounded system,
2. articulate a list of other users with whom they share a connection, and
3. view and traverse their list of connections and those made by others within the system.

The nature and nomenclature of these connections may vary from site to site. Whereas boyd and Ellison use the term “social network site” in literature the term “social networking site” can be found as well (cmp. [OZ09]). Although these two terms are often used interchangeably, the scope is different [BE08]. “Networking” emphasizes the ambition to find new people and to get in contact with them. Such networks are often between strangers. For boyd and Ellison [BE08] social network sites also allow individuals to meet strangers, but they enable the users

²⁴danah m. boyd does not capitalize her name in her publications.

to articulate and represent their real “off-line” social networks. Hence, communication in social network sites is primarily between people who already know each other or are somehow part of their extended real social network. In this thesis, the terms “social network” and “social network site” are equivalent.

Each user of a social network site has a user profile. Therefore, a set of questions are given to the user which shall reflect her/his character and provide information in advance. Typically, these questions concern personal attributes like age, gender, location, and interests [BE08]. A personal picture is part of most user profiles. The visibility of a user profile is depending on the policies of the social network site and the possible user settings. Some social network sites allow search engines to crawl the profile information other restrict the visibility to network members or to directly connected profiles only. The public display of the friends list of a user is crucial component of social network sites [BE08]. The list enables users to traverse the social network along their connection graphs, as far as the connections are visible. This fact can raise serious problems. Depending on the available user information – which can be very intimate and extensive – risks range from identity theft to stalking of a user; from embarrassment to discrimination and blackmailing [RA05].

Social network sites vary greatly in their features [BE08]. Some have media-sharing (photo, video, music) capabilities; others have instant messaging, video calls and built-in blogging services. However, the interpersonal (social) tie structure in a social network site is unique. In a social network secondary connections of an individual are weak [How10]. Even though a person is directly connected to 1000 people in a social network, it is not likely that this person has anything in common with a person who is connected to one of her/his 1000 connected people. The interests and values of a person cannot be ascertained or predicted by the connection to this person. In an online community secondary connections are much stronger, because the members share common goals and interests.

Because in a social network the friends on our personal friend list tend to move in the same circles as we do, the information they receive overlaps with what we already know [Gra05]. People that are linked with us via one or two persons – acquaintances in some cases – receive more

novel information. This is part of the “The Strength of Weak Ties” effect identified by Granovetter in 1973 [Gra73]: In a social network more novel information flows to individuals through weak than through strong ties. A likely reason for this statement is that acquaintances are typically less similar to us than close friends, because they simply spend less time with us [Gra05]. Moving ourselves in new circles, connects us to a wider world and therefore acquaintances are better knowledge sources when we need to go beyond what our own group already knows [Gra05]. In a social network site, there are many more weak ties than strong ones, and most such ties may carry information of little significance for ourselves. Granovetter [Gra05] points out that such ties are much more likely than strong ones to play the key role of transmitting unique and non-redundant information across a social network. For this reason, weak ties – and hence to possibility to connect to new people – can be seen as vital for the existence and growth of a social network site.

2.2.3 Motivational Factors for Participation

Social behaviour is complex, whether it occurs on- or offline. Theories from sociology and psychology help to explain social behaviour online, but there is discourse about the impact of identified factors (cmp. [CME06; WF00; WF05]). Research on motivational factors that foster participation in online communities is hard to generalize and unify²⁵. The employed theories to explain the behaviour in online communities focus on different human aspects and the findings of developed prediction models for participation are sometimes contradictory. Therefore,

²⁵Most research on motivational factors in existing online communities can be categorized in two types: Research on communities of practice and empirical studies in public communities. “A community of practice consists of a tightly knit group of members engaged in a shared practice who know each other and work together, typically meet face-to-face, and continually negotiate, communicate, and coordinate with each other directly” [WF05, p. 37]. Hence, some motivational factors which are identified in communities of practice are in the context of a business organization and therefore are of no account for social workflows. An example for such a specific organizational motivational factor is career advancement. Additionally, research on motivational factors for participation in public online communities or social network sites often tends to use surveys which directly ask the community members for their reasons to participate. The given answers are community specific and their terminology may vary.

results from a study cannot easily be transferred to another online community. A cause for that lies in the case study based evaluations of the postulated models. Usually, the prediction models are evaluated with an online community and it is in the nature of communities that the individuals within a community can strongly differ in their attitude, culture, and language [Pre00]. Another cause is the timeline between different research results. For instance, the members of the “WELL”²⁶ system (it started in 1985 and was described by Rheingold in 1993 [Rhe93]) cannot have the same expectations and motivations like current research on Facebook members.

On a social workflow platform, participation is open and voluntary, and the users typically don’t know each other. Knowledge contributors – users who share their social workflows – have no assurances that those who reuse their workflows will ever return the favor and some users will only observe the provided knowledge without actively contributing it. Therefore the leading question of this section is: Which factors influence users to participate in an online community and why would they help inexperienced users by sharing their knowledge?

In the following, this section describes parts of generally accepted sociological theories that are used to describe the social behaviour in online communities. Afterwards, identified factors which influence the user participation in online communities are presented.

Sociological Theories

The Social Cognitive Theory Bandura’s social cognitive theory defines human behaviour as a triadic, dynamic, and reciprocal interaction of personal factors, and the according social network [Ban97; CME06]. The social cognitive theory explains how people acquire and maintain behavioral patterns [Ban97]. Figure 2.5 shows the triadic reciprocal determinism model [WB89; Ban78] of the social cognitive theory. According to Wood and Bandura [WB89] a person’s behaviour is partially shaped and controlled by the influences of the environment (e.g. her/his social network) and the individual person (e.g. her/his expectations, beliefs). Bandura advances two types of expectation beliefs as the major personal forces guiding behaviour [CME06]: *outcome expectations* and *self-efficacy*. An outcome expectation is a person’s estimate that a behaviour

²⁶For more information, please refer to <http://www.well.com/aboutwell.html>.

will lead to a certain outcome. Self-efficacy is the individual conviction that one can successfully execute a behaviour/task to get the expected outcome [Ban77].

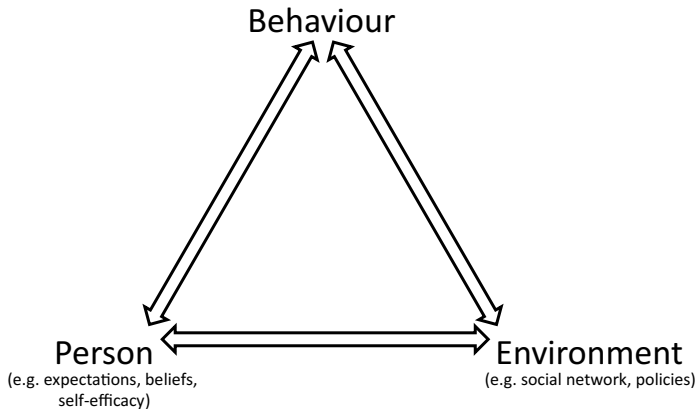


Figure 2.5: Illustration of the triadic reciprocal model, adapted from Wood and Bandura 1989 [WB89]

Bandura’s concept of self-efficacy is defined as “people’s beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives. Self-efficacy beliefs determine how people feel, think, motivate themselves and behave” [Ban94, p. 71]. For example, people with a strong sense of self-efficacy, form a stronger sense of commitment to their interests and recover quickly from set-backs. People with a weak sense of self-efficacy avoid challenging tasks and believe that many tasks are beyond their personal capabilities. People’s belief about their efficacy is dynamic and can thereby influenced and strengthened in four principal ways [WB89]: by *mastery experiences*, *modelling*, *social persuasion*, and their *physiological states*. They are described in the following.

Mastery experiences suggests that performance successes strengthen self-beliefs of capability. To gain a resilient sense of self-efficacy, people

need experience in overcoming drawbacks through perseverant effort. The gained experiences through repeated successes will enable people to manage set-backs and failures with less affection [WB89].

The second way to strengthen self-efficacy is through modelling. Modelling in the sense of Bandura [Ban74; Ban77] means observational learning. If a person observes a certain behaviour of a proficient model (person) then the observing person may internalise and adapt the observed behaviour patterns if it raises positive outcome expectations²⁷. As examples Bandura calls driving automobiles or skiing [Ban74]. By making such an observation an individual takes a decision if it is worthwhile to take the effort to pursue the activity. In these cases one's mobility through driving and having fun while skiing. "Seeing similar others succeed by sustained effort raises observers' beliefs about their own capabilities, whereas observing similar others fail despite high effort lowers observers' judgement of their own capabilities and undermines their efforts" [WB89, p. 365]. Observational learning affects self-efficacy beliefs through a comparison with others judging their personal capabilities. It is one of the central keys that influence behaviour [Ban77]. Bandura states that "fortunately, most human behaviour is learned observationally through modeling: from observing others one forms an idea of how new behaviors are performed, and on later occasions this coded information serves as a guide for action" [Ban77, p. 22].

Social persuasion is a third way of increasing people's beliefs in their capabilities. It evolves when third persons encourage a person to perform and convey positive appraisals [WB89]. In addition to raising people's self-efficacy, they assign tasks to them in ways that bring success and avoid placing them in situations in which they are unable to cope [WB89]. The physiological state is especially important for activities which rely on activities involving strength and stamina.

Since people have a conceptualization about behaviour patterns, they tend to anticipate and appraise certain situations. Therefore, conditions of reinforcement also have a strong motivational effect [Ban77]. As a result of prior experiences, people gain outcome expectations about certain actions. The decision to participate on an activity is therefore

²⁷To interested reader might refer to the "Bobo doll experiment" by Bandura et al. [BRR61; BRR63] where they showed that the behaviour of children can be influenced by showing them a certain interaction behaviour with the "Bobo doll".

to a large extent depending on the anticipated consequences. According to Bandura [Ban77], observing the reinforcement of others is not only informative for an individual but can also have incentive motivational effects. Seeing others reinforced can function as motivator, because outcome expectation arise with the observers' assumption that they will be similarly rewarded for analogous performance. The motivational effects can be classified according to Ryan and Deci [RD00] between intrinsic motivation, which refers to doing something because it is inherently interesting or enjoyable, and extrinsic motivation, which refers to doing something because it leads to a separable outcome. In online communities an important extrinsic motivation is reputation [WF05]. In addition to enhancing their reputation in a social network, individuals may receive intrinsic benefits from contributing knowledge [WF05]. Therefore, individuals must think that their contribution to others will be worth the effort and that some new value will be created [NG98]. Nevertheless, an original intrinsic motivation can result in an extrinsic reward. An example for such a case would be the writing of a recipe for a cooking community. Although, it was written with the intention in mind that people will enjoy it, others might provide additional tips to refine the recipe in a reciprocal manner. This is also supported by Kollock [Kol99] as he states that a person is motivated to contribute knowledge to a group in the expectation that one will receive useful help and information in return; that is an anticipated reciprocal outcome.

Besides the reciprocal expectations there is also altruism that engages individual motivation to share knowledge. However, reciprocity is fundamentally different from altruism [FG00]: In case of reciprocity, the actor is responding to friendly or hostile actions even if no material gains (extrinsic reinforcements) can be expected. Altruism is a form of unconditional kindness, i.e. altruism given does not emerge cause of received altruism. Many of the things that people like to do, is for their own sake [Ban77]. Altruistic people do things just because they like to do it.

A social workflow platform serves an online community in which private individuals with common interests and goals interact to share information and knowledge. Thereby, the people are engaged in social interactions. Bandura's social cognitive theory deals with the human behaviour in general and admits the influencing factor of the environment. How-

ever, the social interactions within the environmental social network are not regarded in detail. At this point Bandura's theory is limited in addressing social interactions within a social network and how they influence a person's behavior [CME06]. Consequently, the social capital theory is now described which centres on the relationships in a social network.

The Social Capital Theory In 1998 Nahapiet and Ghoshal [NG98] summarized research on the social capital theory since the 1960s concluding with a prediction model for the creation of new intellectual capital. Thereby, they point out, that the "central proposition of social capital theory is that networks of relationships constitute a valuable resource for the conduct of social affairs, providing their members with the collectively-owned capital, a 'credential' which entitles them to credit, in the various senses of the word" [NG98, p. 243]. Much of this social capital is embedded within social networks. Network members can gain privileged access to information and to opportunities by explicitly exploring the network of "weak ties" [NG98; Gra05]. They add that significant social capital in the form of social status or reputation can be derived from membership in specific networks. The main difference between social capital and other forms of capital is that social capital is embedded in the social realm and resides in the fabric of relationships between individuals and groups of individuals [Put95; WF05]. It is the fundamental proposition of social capital theory that these relationships provide access to resources [NG98]. This way, the interpersonal ties in a social network directly affect the personal knowledge (i.e. "who you know" affects "what you know") [NG98]. Nahapiet and Ghoshal [NG98] define social capital as "the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships possessed by an individual or social unit" [NG98, p. 243]. As a set of resources rooted in relationships, social capital has many different attributes and therefore Nahapiet and Ghoshal divide social capital in three forms: the *structural*, the *relational*, and the *cognitive* dimension of social capital. Figure 2.6 illustrates the structural and the

relational dimension²⁸ of the social capital theory with their impact on the creation of new knowledge²⁹.

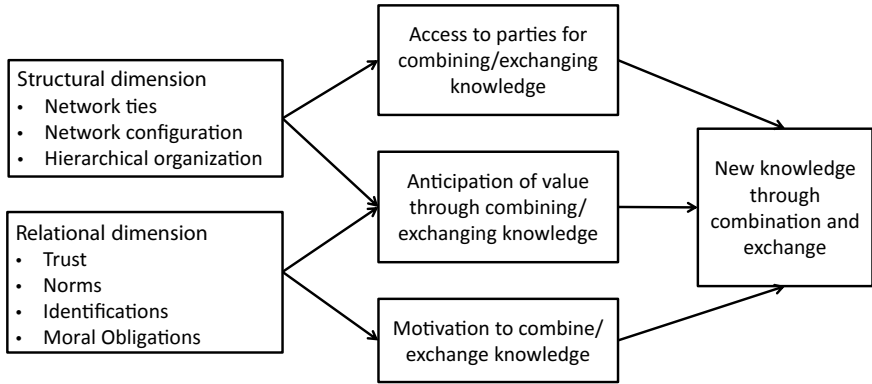


Figure 2.6: Impact of the structural and relational dimension of social capital on creating new knowledge, adapted from Nahapiet and Ghoshal 1998 [NG98].

The structural dimension concerns the presence or absence of network ties between individuals, the network configuration, and the network hierarchy [NG98]. Network ties may suggest that there are three information benefits [NG98]: access, timing, and referrals. Access refers to the individual's possibility to use networks to gain a valuable piece of information. Timing means that the personal social ties may be beneficial in time critical situations. A person who receives an information sooner

²⁸The cognitive dimension has been left out of the figure because it focuses on a common cultural background. This dimension is within the boundaries of an organization, suggesting that organizations whose members share a common language, narratives and understanding are better suited for the creation of new knowledge/intellectual capital [WF05]. The distinction and the empirically refinement of the dimensions is also a current topic in research. In fact, often only the structural and relational dimension are regarded as influencing factor for social capital and there is much discourse on the terminology [RNZ10].

²⁹The work of Nahapiet and Ghoshal [NG98] draws on the work of Granovetter's discussion [Gra92] of structural and relational embeddedness.

than another person may have an advantage to react faster. Referrals are processes that provide information about opportunities for individuals and frequently include reputational endorsements. The network configuration describes the structure of the network according to measures like density, centrality, and also hierarchy. These features are associated with the ease of information exchange and flexibility regarding how a piece of information can be accessed. The aspects of the structural dimension cannot directly be seen as motivational factors for contributing knowledge in a network, but they describe mainly how individuals and groups can access and exchange knowledge [NG98]. In this dimension the network ties are the only aspect which also influences an anticipation of valuable information. The access to a valuable piece of information includes knowing who can use it. Therefore, network ties provide an efficient information-screening and distribution process [NG98].

The relational dimension describes the motivational factors based on relationships in a network [NG98]. It focuses on the personal relations people have according to their interaction history, such as respect and friendship, that influence people's behaviour over time. For example, two people may have similar positions in a network configuration, but if their attitude and personal attachments to other members differs, their actions are likely to differ too [NG98]. The key motivational factors of the relational dimension are trust and trustworthiness, norms of cooperation, obligations and expectations, and identification. If there is a higher level of trust, people are more willing to take risks for others [NG98]. This does not necessarily mean that people are also more willing to provide information to a network but people seem to have a greater openness to the creation of new knowledge [NG98]. Moreover, trust has a direct interaction with cooperation between individuals. A cooperation between individuals leads to trust and trust engages in new cooperations [NG98]. Norms of cooperation can evolve to "expectations that bind" [NG98]. Teamwork is an example for such a norm of cooperation. It can establish a strong foundation for the creation of new knowledge [NG98]. Moral obligations³⁰ are a feeling of a person if this person has availed herself/himself of the actions of another person [Col88]. Reciprocally,

³⁰Nahapiet and Ghoshal do not directly talk of moral obligations but on "obligations and expectations". Thereby, they focus rather on a company context. However, obligations in the social capital theory also include moral obligations which are here explained.

the person who performed the actions gets outcome expectations for the future. Felt obligations and the expectations are always based on trust between two parties [Col88]. Identification is the process whereby an individual recognizes herself/himself as part of a group of people [NG98]. The degree of identification with a group or collective enhances concerns for collective actions [NG98]. For this reason, identification influences the anticipation of value to be achieved through exchanging knowledge and the motivation to perform this way [NG98].

Identified Motivational Factors

Bandura's explanation for human behaviour [Ban77; Ban78; Ban94; Ban97; WB89] and the importance of the social capital [Col88; Col90; NG98; Gra92; Gra05] give some insights. Insights about individual's cognitive processes and the importance of interaction ties between individuals which influence the degree of participation in a community. However, these theoretical prediction frameworks cannot be directly applied to online communities. Rather, they are the fundamentals for empirical research and new derived prediction models which focus on online communities (cmp. [CME06; WF05; SU03; End+07]).

In the following, motivational factors are presented and their impact on online communities is discussed. Trust has been selected because all motivational factors of the relational dimension of social capital theory are based on trust between individuals. Reciprocity has been selected because in Bandura's social cognitive theory the social network and a person's attitude have a reciprocal relation (the outcome expectations) that determines an individual's behaviour and commitment. A sense of self-efficacy has been selected because Bandura has shown that it is mandatory for a person to engage in actions in a social network. However, these motivational factors must not be regarded in isolation. They are correlated and they influence each other.

Trust In his book "Trust" Fukuyama [Fuk96, p. 26] explains trust as "the expectation that arises within a community of regular, honest, and cooperative behavior, based on commonly shared norms, on the part of other members of that community. [...] Social capital (theory) is a capability that arises from the prevalence of trust in a society or in certain parts of it. It can be embodied in the smallest and most basic

social group, the family, as well as the largest of all groups, the nation, and in all the other groups in between”. An individual is not likely to be motivated to share or contribute knowledge with another individual or a group if she/he perceives them to be dishonest or unreliable [SU03]. In return, when one views a community as trustworthy, there is likely to be a greater motivation to participate and to share knowledge [SU03] in the expectation of mutual reciprocity. In the context of resource exchange, the experiences on a relationship between individuals – the social ties – have a strong effect on trust [CME06]. However, it seems that trust is especially important in risky knowledge sharing communities and not in volunteer online communities [CME06; Col90]. A risky knowledge sharing community would exist in a competitive organizational context of a company.

Commitment and Reciprocity Commitment or attachment is another motivational factor³¹. There are studies that examine the influence of altruism for knowledge sharing [BH09; WF05] but it is likely that the impacts of identified altruistic factors are interrelated with non-altruistic factors originating in the individuals’ environment. According to Kollock [Kol99], complete devotion to a group – and thereby pure altruism – is rare. Attachment to a community rather results in the expectation of certain reciprocal outcomes. The environment has an impact on the behaviour and hence on the commitment of an individual in a community. Outcome expectations³² like reputation or even prestige play a key role for the participation in online communities [CME06; BH09; Kol99; WF05]. Reputation is an identity-based encouraging norm of reciprocity and the contributions to a community will be likely increased to the extent that a person feels a commitment to the community [Kol99].

³¹ Although psychology distinguishes between three types of commitment, literature on motivational factors for knowledge contribution does often not differentiate. Theories of group psychology distinguishes between (1) affective commitment, based on feelings of closeness and attachment to a group or members of the group (2) normative commitment, based on feelings of rightness or felt obligation to the group, and (3) need-based commitment, based on an incentive structure in the group [KRK11].

³² Concerning a social workflow community, which is based on the willingness of its members, monetary outcome expectations are not taken into account in this thesis.

Sense of Self-Efficacy Another explanation that one contributes valuable information is that the act is in a sense of self-efficacy [Kol99]. Bandura's research on the importance of self-efficacy is well-developed and making contributions to a community can help a person believe she/he has an impact on the group [Kol99]. Thereby, a person would gain a stronger sense of self-efficacy and then contributions are likely to be increased to the extent that people can observe how their attributable actions influence the community³³ [Kol99; SU03]. Feedback³⁴ is a form of social persuasion that could strengthen self-efficacy because the personal actions become attributable (cmp. [BS93; Gam+11]). Although it is apparent that a greater self-efficacy can lead to a higher motivation to contribute knowledge, some empirical research find it difficult to verify its measurable impact as the construct of self-efficacy also influences other factors mentioned before³⁵ [WF05].

A last aspect which has an effect on knowledge contribution considers the fact that the distribution costs in an online community for a piece of information can be near zero [Kol99]. In contrast to the aforementioned motivational factors this aspect is independent of theoretical models and frameworks trying to predict the behaviour of humans in communities. Kollock [Kol99, p. 229] states that "it may be the case that many people spend time and effort producing goods they intend to contribute to the group, another path to the production of public goods³⁶ is as a simple side-effect of private behaviour". Kollock compares it with one's need to write a particular computer program for her/his own use

³³Sharat [SU03] refers to the broader term of a "sense of community". A sense of community is "a feeling that members have of belonging, a feeling that members matter to one another and to the group, and a shared faith that members' needs will be met through their commitment to be together" [MC86, p. 9] and it influences knowledge sharing in a community.

³⁴Bellotti and Sellen [BS93] propose a technological feedback mechanism in a ubiquitous computing environment for the act of conveying information to others to lower disembodiment and dissociation from the environment.

³⁵Wasko and Faraj [WF05] split Bandura's self-efficacy into "self-rated expertise" and "tenure in the field". However, none of these attributes could significantly improve knowledge contribution nor the quality of knowledge contribution. Bandura himself states that self-efficacy, at least to some extent, touches everything people do [Ban84].

³⁶"Public goods are goods that anyone might benefit from, regardless of whether they have helped contribute to their production." [Kol99, p. 223]

with the ambition to solve her/his particular problem at hand. Having written the program, the distribution costs of sharing it with others or a community may be near zero. This is what he means by “side-effect” as it was not the original motivation to create a valuable good/piece of information.

2.2.4 Conclusion

In this section online communities were described and reasons were given why people tend to participate voluntary in online communities. Thereby, the differences between online communities and social networks were highlighted. Although both terms are often used interchangeable, they have different characteristics which are important for a social workflow platform. Similar to scientific workflow communities, an online community for social workflows shall emerge enabling its members to collaboratively work on complex problems and to share their experiences among the community. Social network sites could play an important role in social workflows for the delegation of tasks to personal friends and the “The Strength of Weak Ties”-effect could lead to a better dissemination of social workflows. Particularly with regard to the fact that this thesis deals with foundations for a social workflow platform, it should be discussed how technology could support new users. The last part of this section was about general motivational factors that cause people to participate voluntary – in contrast to people in an organization of a company – in an online community. Since many research results aim at specific online communities or business communities, generally accepted sociological theories were summarized to identify the factors trust, self-efficacy, and reciprocity. Section 3.5.3 discusses how a social workflow platform could influence these factors positively.

2.3 Process-Oriented Case-Based Reasoning

The body of experiential knowledge that evolves in online communities can be searched by traditional Information Retrieval methods and collaborative tagging. “Collaborative tagging describes the process by which many users add metadata in the form of keywords to shared content” [GH06, p. 198]. Independent of the media type of the shared content,

(collaborative) social tagging can be used to create an index structure known as “Folksonomy”. People who consume the shared information are usually responsible for the act of tagging [Van07]. For this reason, the categorisation performed by individuals is of good quality because they are familiar with the content. The act of tagging itself is simply performed by allowing people to freely annotate shared objects.

To benefit from the experiences other people already made, private individuals shall be supported in collecting their social workflows in a repository. Allowing such social tagging could produce a folksonomy for a workflow community. Then, a text-based retrieval like in the afore-said online communities can be used for the recorded information in a workflow repository. Such information comprehend the annotated tags of a workflow as well as the workflow’s title and description. Recently, tag-based similarity searches were employed for the retrieval of scientific workflows which are backed by an online community³⁷ [STD10; Cos+12]. In general, tag-based searches have been proven to be a good means to search for content in an endless number of online communities that deal with media objects (e.g. pictures or videos). However, a tag-based retrieval is backed by human-provided textual knowledge and does not take advantage of the structure of a workflow. The structure itself gives information about the order of the contained activities and reflects specific experiences of other people expressed by modelled data dependencies.

For designing a new workflow, inexperienced users may start describing their problem by creating a brief sketch of the aspired workflow (e.g. a sequence of a few tasks) they are interested in, thereby specifying already some structure of the intended solution. In application domains in which people have a lot of in-depth knowledge but are not trained workflow modellers, they shall be assisted in doing so and relieved from the burden to formulate her/his intended workflow with an additional textual description. Methods for a workflow retrieval are required that deal with underspecified workflows allowing a search that anticipates the individual utility of existing workflows in a workflow repository. Similar to a text-based retrieval, Process-Oriented Case-Based Reasoning (PO-CBR) offers methods to search for workflows including structure-based similarity computation as well as textual/tag-based similarity measures. In the following the workflow retrieval with PO-CBR

³⁷ Among workflow communities, myExperiment can be considered as one of the biggest. It provides a tag-based search for scientific workflows.

is described. Starting with an introduction of Case-Based Reasoning (CBR) as part of experience management up to a generic method for the retrieval of workflows.

2.3.1 Case-Based Reasoning

The ability to solve problems is one of the most important aspects of human thinking. Especially experienced persons, experts, are able to associate past events with present events [Elg09]. An expert excels by her/his comprehensive knowledge, effective methods for recognising and solving problems, correctness, and a flexibility regarding new problems [Hum02]. Hence, it is expertise of an organisation which is one of the most valuable resources of a company to succeed in the market. For this reason, companies tend to employ ICT-systems to find the achieved knowledge, to conceive it, and to retain it. This broad field dealing with these issues is called knowledge management [Ber02]. “Experience management is a special kind³⁸ of knowledge management that is restricted to the management of experience knowledge, i.e., specific knowledge situated in a particular problem solving context” [Ber02, p. 1]. For the performance of an intelligent organisation which demands problem-specific knowledge, it is of particular interest if available structured organisational knowledge increases the ability to solve specific problems. Case-Based Reasoning (CBR) is a methodology for structuring, retrieving, and storing problem-specific experiential knowledge.

CBR provides appropriate means for representing experience [Ber02]. Past experience items are referred to as cases. According to Kolodner [Kol93, p. 145] “a case is a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner”. In simple terms, a case is a problem solution pair consisting of a description of a past specific problem and the concrete solution to solve this problem. The collection of all cases (experience items) is stored in the so called case base. Following the traditional

³⁸The distinction between general knowledge and experiential knowledge cannot always be clearly defined. It is depending on the question how much the context of a situation/experience can be regarded as general knowledge. A different view is taken by Althoff et al. [Alt+01]. For them the main difference between general knowledge and experiential knowledge is that experiential knowledge is acquired through learning from a continuous stream of experience.

CBR problem solving cycle by Aamodt and Plaza [AP94], CBR provides methods for *retrieving* cases similar to a current problem and to *reuse* the solution of the best fitting case. If the found solution requires an adaptation to be applicable on a new problem, a new solution has been obtained as well. This new case (problem-solution pair) might be *revised* by a person before it is *retained* in the case base. The interaction between a CBR system and a user following these phases of the “4-REs”-cycle (REtrieve, REuse, REvise, and RETain) leads to an increasing case base and thereby to an increased problem-solving capability of a CBR system.

An aim of this theses is to enable the reuse of social workflows (cmp. section 1.2). For this purpose, only the retrieval and reuse phase in CBR need to be regarded ³⁹. Whereas, the reuse of workflows covers their adaptation by workflow modelling the retrieval phase needs to be further explained. In the following, cases in CBR are explained in more detail to give an understanding to the reader how experience is structured in CBR. Then the general retrieval of cases is explained before a form of workflow retrieval is presented.

2.3.2 Case Retrieval

A case represents a real specific experience item. Thereby, it might be structured in several ways. The choice of the structure is depending on the experience management domain and the task that requires experience reuse [Ber02]. In general, a case consists of a problem part that characterizes the experience in a way that allows to assess its reusability in a certain situation and a solution part that provides the solution steps to solve this problem, i.e. the gained experience during the solution process [Ber02].

The problem part of a case describes all relevant facts in order to allow the decision if a case can be reused in a certain situation. Thereby, the formulation of the problem serves as an index of a case in the case base. However, in contrast to a regular index (e.g. the word index of a book) the structure of the problem part may have a complex structure. Recalling that in opposition to general knowledge, experience items include

³⁹The revise and retain phase can be of value on a later development state of social workflows. They come into play when a considerable amount of social workflows has been collected and their value for experience management can be assessed.

the context in which they occurred (cmp. [Kol93]), the definition of the problem must also describe the global context in which the experience occurred. The solution part must contain all required information to solve the (complex) problem definition of the case. It must be detailed enough so that a person can apply the suggested solution to the new problem and if needed to enable the person to adapt the solution.

The representation of the problem statement in cases may vary depending on the needs of the application domain [BKP05]. Traditional representation forms are based on a *feature vector*, a *structured* or *textual* representation⁴⁰. The simplest form to capture a problem and its solution is in a free-text form. A free-text representation is good for domains with a big document corpus with a considerable amount of know-how inside the documents [Ber02]. The intended users can immediately make use of the experience contained in the respective documents. The structured approach represents cases according to a common domain model [Ber02]. Thereby, the structure itself may be represented in different ways. Structured cases are often represented as graphs or in an object-oriented manner expressing “part-of” or “is-a” relationships⁴¹. The structured approach is good for domain models in which the objects – on which experience shall be earned – are composed by further composable objects. An example would be the diagnostic capability of a car engine. In order to gain experience about the operation of a car engine, the engine itself can be described as a composition of the inherit parts (e.g. spark plugs and ignition coil are part of the ignition; the pistons and valves are part of the compression). The feature vector representation represents a case with attribute-value pairs. The set of attributes can be fixed for every case or can differ from case to case. This happens, if some cases are not fully captured and missing values in the attribute set occur. Further, each attribute may have a different range definition for its value.

Independent of the representation of a case, the reuse of experience requires more than just the formalisation of the experience. The central task for experience management is the retrieval of the existing formalised

⁴⁰For more representation forms, the interested reader is referred to [BKP05].

⁴¹Bergmann et al. [BKP05] explicitly emphasize the relational aspect of a structured representation. According to them, structured cases are relations between the kinds of elementary objects that comprise them. A pure attribute-value representation has no relationships and hence it is a self-contained representation.

experience [Ber02]. Therefore, “it requires knowledge about the utility⁴² of an experience item (or case) in a certain problem solving situation” [Ber02, p. 93]. The similarity computation between cases is used in CBR to approximate the utility in a certain situation. A query to a CBR system is a case with a problem part but without the solution part. During the retrieval, the new problem statement is compared to the existing problem statements of the respective cases in the case base. CBR is based on the idea that similar problems have similar solutions and the similarity computation between cases is the central concept of the methodology.

Figure 2.7 illustrates the retrieval phase in CBR based on this assumption. It shows a case base with a collection of cases $c_i = (d_i, l_i)$ with problem description d_i and its solution l_i . The case base is queried with a new problem p . Then, the similarity $\text{sim}(p, d_i)$ between a past problem d_i and p approximates the utility $u = (p, l_i)$ of the solution in case c_i for the new problem p .

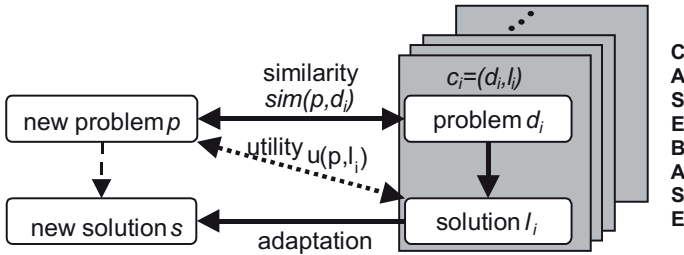


Figure 2.7: Illustration of the CBR retrieval phase [Ber02, p. 94]. The utility is approximated with the similarity.

This approximation of the utility function u is crucial to CBR, because the function u cannot be fully formalized in a complex application domain [Ber02]. Normally, we can only obtain partial knowledge about a certain domain, which enables us to decide, if the utility of a solution

⁴²Utility can be expressed in different ways and is depending on the current situation and preferences of the user. The interested reader is referred to [Ber02].

is higher, equal or less than the utility of another solution [Ber02]. In fact, the retrieval in CBR is based on this preference relation between cases and not on the absolute numerical value of the similarity computation. The similarity function *sim* maps the similarity of two cases in an interval between [0;1] showing total dissimilarity with 0 and equality with 1. The absolute computed similarity between a new problem *p* and a case *c_i* relates to the position of the case in the preference relation of the approximated utility.

Depending on the representation form of the cases, different similarity measures can be applied. If cases have only a textual description, a simple text matching approach (e.g. the Levenshtein distance) can be applied. Such an approach regards the syntactical similarity between two texts and does not include semantics. If the case representation includes a domain model then the semantics of the model can also be included in a similarity measure. Such semantics can be expressed by the case representation itself. An hierarchical composition in a structured case representation is a semantic information that offers insights about dependencies between parts of the objects. However, this is not sufficient for most application domains and semantics are usually expressed in an ontology that provides a conceptualization for the case objects. “An ontology is an explicit specification of a conceptualization“ [Gru93, p. 199]. A conceptualization is a goal-oriented, simplified model of the world [Gru93]. The ontology for objects and the relationships among them determine what “exists” in a domain model and thereby it represents the general knowledge in a CBR system for the contextualization of the experiential knowledge. The larger and more complex the domain, the more effort must be put in the development of the ontology. However, for pragmatic considerations a similarity measure should be modelled with a reasonable effort [Ber02]. It is a trade-off between the accuracy of the domain model and the improvement of the similarity computation between cases.

As an alternative to traditional ontology engineering, folksonomies have been proven to successfully implement semantic searches (e.g. YouTube, Delicious, Flickr or Twitter). Shirky [Shi05] even states that traditional “ontologies are overrated” and he assumes that instead of ontologies much more organic ways of organizing information are com-

ing by social tagging⁴³. Gruber [Gru07] agrees to Shirky's statement that ontologies and their taxonomies are too static. Gruber regards the forced categorization of things and their hierarchical relationship as a flaw of traditional ontologies. Instead, he proposes the possibility to create an ontology based on folksonomies: Tag annotations for a piece of information should be shared across different applications and commonalities between tags must be captured (what again requires a kind of conceptualization).

In the end, the impact of the domain model on the similarity between a new problem and a case in the case base is determined by its weighting in the similarity function *sim*. It is likely that the inclusion of a semantic domain model improves the utility approximation, because it extends syntical information without neglecting them. Social workflows are assumed to be a complex application domain serving different purposes and private goals. Accordingly, a domain model would have to cover many different application areas, too. Traditional, static ontologies are probably not able to deal with a domain with many application areas because they need to be adapted to every new scenario. In such domains a folksonomy is preferable to a traditional ontology even though folksonomies are patchy at the beginning of an online community. Their quality is expected to increase with the number of users who participate willingly in a community. To avoid this bootstrapping problem, the next section deals with generic methods from PO-CBR for the retrieval of workflows. They allow a similarity computation between workflows with semantic annotations which may be incomplete. The domain model might be changed later on in order to improve the similarity computation.

2.3.3 Workflow Retrieval

A PO-CBR workflow retrieval requires two parts: a case representation able to include workflow graphs and a retrieval mechanism that computes the similarity between two (workflow) cases. Cases which contain workflow graphs clearly belong to a class of highly structured case representations [BG11] and hence there is considerable computational effort for the similarity computation. In PO-CBR applications a case describes

⁴³Shirky criticises the way ontologies are formalized and the effort for developing them. His critic does not aim at ontologies in general.

a workflow prototype, a workflow instance that has been executed successfully or it records the deviation that have been applied on a workflow [MMR14]. Regarding the fact that in traditional CBR a case provides an index structure for the retrieval, the computational complexity in PO-CBR increases with the size of the case base but also with the arbitrary complexity of the workflows inside the cases. Therefore, it is not surprising that the efficient index computation is a current topic in the PO-CBR community [BMW13; BS13; BG14; KL14; MB14a].

Existing approaches for a similarity computation between workflows can be usually classified as being either annotation-based or structure-based. Each of these approaches has its particular strengths and weaknesses [Sta+14]. Structure-based approaches only rely on the syntactical information contained in its graph structure, but they do not need human-provided knowledge to model ontological knowledge. Bergmann and Gil [BG14] and Starlinger et al. [Sta+14] summarize structure-based approaches that only regard the syntactical information of a workflow⁴⁴ [Kap+10; ML12; God08; MTB07; DJ08; DDG09; Sil+11; San+08; FR10; XM07]. The algorithms of these approaches deal with the edit-distance between graphs, sub-graph isomorphisms, or maximal common sub-graphs [San+08; BG14] causing problems due to a high computational effort. Annotation-based approaches are independent of the graph structure and the WfMS that executes the workflows. This allows to compare workflows across different repositories. Yet, they require an active community that fosters a folksonomy and it cannot be ensured that every workflow in a public repository has assigned tags. For scientific workflows effort has been put in clustering algorithms that use semantic annotations to categorize workflows in order to easily re-use them [Cos+12; STD10]. Gil et al. [Gil+09] do not use human-provided annotations but use semantically annotated graphs which have constraints about the contained workflow components.

Currently, there are only few approaches that allow a structure-based retrieval including semantics. The SAI framework allows a domain-specific similarity assessment with complex structures like workflows [KL11] and Bergmann and Gil [BG11; BG14] present an approach that

⁴⁴Thereby, the different approaches aim at different application domains. Mainly at business processes and scientific workflows. The interested reader might refer to these citations as they link to many more approaches dealing with structure-based similarity computation between graphs.

allows a semantic workflow search including the workflow structure and semantic annotations. In the following of this section the approach of Bergmann and Gil is summarized. It has been selected because:

- It provides a generic case representation for workflows that can be used for different workflow formats.
- It allows a fast workflow retrieval even in large workflow repositories. The generic workflow structure allows a distributed similarity computation.
- Nodes and edges in workflows may be annotated with semantic descriptions like tags or ontologies.

In their work they use a workflow representation based on semantically labeled directed graphs. Such a graph is represented by a 4-tuple $W = (N, E, S, T)$. N is a set of nodes and $E \subseteq N \times N$ is a set of edges. All nodes and edges have types assigned by the function T . Typically, these are activity types or data types which occur in a workflow. Additionally, all nodes and edges can have a semantic description S from a domain model. These “NEST”-graphs are a generic structure to represent different kinds of workflows⁴⁵. Further, they are specialized for the representation of workflows by typing nodes and edges with a workflow specific semantic.

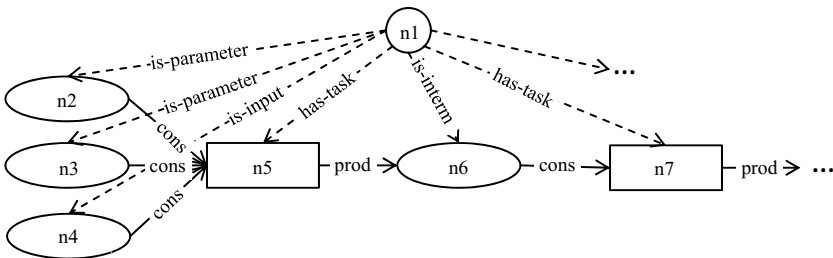


Figure 2.8: Illustration of a workflow in the NEST-graph representation [BG11, p. 121].

⁴⁵They are applicable for business, scientific, and cooking workflows [BG11; MB14b].

Figure 2.8 shows an example for a workflow in the NEST-graph representation. Each NEST-graph has exactly one workflow node (“n1” in the illustration). It stands “on top” of the workflow and every other node is linked with the workflow node in a particular part-of relation (e.g. a “has-task” for every activity, “is-parameter” for input data object or “has-control” for control-flow elements). Each activity in a workflow is represented by a task node (rectangular in the illustration). Each data object is represented by a data node (oval in the illustration). Each control flow element in a workflow, such as loop blocks or split/join elements (cmp. section 2.1.3) are represented as a control-flow node. The data-flow is illustrated with producer and consumer relationships between data and task nodes. In the illustration the task nodes “n5” and “n7” are connected via the data node “n6”. “n5” produces “n6” and “n7” consumes “n6” for its execution. The control-flow among activities is represented by control-flow edges. An control-flow edge from a task node A to a task node B indicates that A is executed prior to B. Control-flow edges correspond to a sequence in a workflow.

For the retrieval of workflow cases the similarity between the query workflow QW and each case workflow CW must be assessed. The similarity computation for NEST-graphs can be considered as an enhancement of the local/global principle for structural and feature vector cases [BR01; Ber02]. The local/global principle says that in order to compute the similarity between two objects, the overall similarity can be decomposed in regard to the composition of the objects. This means for the retrieval of complex case objects that each part of an object has its local similarity measure. The overall/global similarity is composed of the results for the local similarity values. The global similarity $sim(QW, CW)$ between a query workflow QW and a case workflow CW is obtained by aggregating local similarity values of nodes and edges within a graph mapping process. The mapping process is defined with the mapping function $m : N_q \cup E_q \rightarrow N_c \cup E_c$. The legal mapping m is a partial, injective mapping function of the nodes and edges in QW to the ones in CW preserving the types of the nodes and edges, i.e. the similarity is only computed between same types. Since the mapping can be partial, not all nodes and edges of the query workflow must be mapped. An edge can only be mapped if the nodes that the edge connects are also mapped to the respective nodes which are linked by the mapped edge. The injective property of m ensures that each node and edge of CW is

mapped only once. This mapping process is an optimization problem with the aim to find a mapping configuration with the highest possible similarity between QW and CW .

The best legal mapping m – resulting in the highest possible similarity – is used to compute the aggregated overall workflow similarity $sim_m(QW, CW)$. Finally, the found maximum similarity expresses how well the query workflow is covered by the case workflow. If the overall similarity is 1, then the query workflow is exactly included in the case workflow.

2.3.4 Conclusion

This section dealt with experience management. Therefore, PO-CBR was introduced as a branch of Artificial Intelligence targeting at the reuse of procedural knowledge. PO-CBR was explained in the context of traditional CBR which follows the same basic idea. In contrast to knowledge management, PO-CBR gathers specific experiential knowledge dealing with the reuse and adaptation of workflows. The aim is to apply the collected experiential knowledge on new problems that may occur in the future. For the foundations of a social workflow platform the retrieval step in PO-CBR could be relevant. The inclusion of a workflow retrieval enhances the concepts for sharing and reuse of social workflows. Sharing of tagged social workflows would already allow users to browse in a catalogue of social workflows and even to search for workflows with a semantic annotation. However, the semantic knowledge of tags is human-provided and thus the community members must proactively foster tagging of existing social workflows. Moreover, the information contained in the structure of a workflow would be omitted. Therefore, an approach was described that allows a workflow retrieval based on the structure of a workflow and on semantic annotations that can be linked to all parts of a workflow. This allows a similarity-based semantic search for social workflows which is more sophisticated than pure tag-based or structure-based approaches.

2.4 Related Approaches

Until today, the terms “social workflow” and “social workflow platform” are not defined in the literature. An online search using Google⁴⁶ for these terms turns out that they are not frequently used. The idea of social workflows, presented here, is closely related to different research areas and application domains and therefore these different views are regarded. This consideration serves on the one hand as delimitation of the term ‘social workflows’ from ‘personal’ or ‘private workflow’. On the other hand it shows the intersections of this work with different related research areas.

The idea to employ workflows in a non-business context is not a new one. Evidence is given in research as well as in commercial systems. These approaches will be presented and their differences are discussed. The focus of this work lies on the creation, management and reuse of procedural knowledge for private individuals and communities. In this context, research has just started to describe Social BPM (S-BPM) which tries to merge traditional BPM with collaborative social software. Hence, related work in S-BPM and workflow systems and management is regarded first.

2.4.1 Research

Social Business Process Management

Along with the development of the idea of “Social Business” or “Business 2.0”, companies start to develop social business software which integrates social aspects. The aim is to foster flat hierarchies and a stronger collaboration across all levels in a company, e.g. by means of communication tools and common social software - lend from the Web 2.0 - like wikis. This is an actual topic in BPM research [OSV13] where social BPM (S-BPM) is sometimes called Subject-Oriented Business Process Management or Social Business Process Management. In the end S-BPM means that business process management itself becomes a collaborative process, including all stakeholders (employees, suppliers, partners, etc.). The knowledge and the experience of the stakeholders shall contribute

⁴⁶The term “Social Workflow” resulted in 7.690 entries and “Social Workflow platform” in 10 entries. Performed in November 2014.

to the overall quality of a business process. Two examples of S-BPM research are now considered.

In 2010, Rito-Silva et al. [Sil+10] propose a human-centered approach following the principles of agile software development, called AGILIPO. Their research aim is to define an agile business process methodology that shall overcome weaknesses of traditional BPM, especially the lack of stakeholder involvement. By including different stakeholder groups different perspectives can influence the process design. Managers have an abstract top-level view while normal employees have a rich detailed view on the same processes. Their idea is to empower all community members to participate in the design of new processes. This way, the delay between process design and enactment shall be shortened because organizations and business environments often change faster than the available processes. Agility in AGILIPO is performed by allowing under-specified process models with “generic” activities which can be specified during the run-time of an instance. When an activity has been completed, its executor (workflow participant) may annotate the activity with tags to reuse the instance in a next evolution of this process. AGILIPO promotes a social software environment in which the folksonomy of tagged activities can be discussed and rated. So far, Rito-Silva et al. [Sil+10] only sketch the idea of AGILIPO but an implementation of the proposed features has not been described.

In 2012, Schönthaler et al. [Sch+12] presented the Horus methodology. Horus provides comprehensive support for the BPM life cycle including all stakeholders of the business community. A core component of Horus is the “Social BPM Lab” [Cap+13] that strives for a re-thinking how BPM is done. During a business process many stakeholders are involved in strategic, tactical, and operational issues, sharing their knowledge in a networked structure. Horus fosters knowledge management via an integrated social business network so that all involved people can work collaboratively. Horus is a mature BPM tool that allows to model, simulate, and execute workflows in a cloud-based environment. However, knowledge management in Horus does not address process reuse and learning from experience of others.

Workflow Management Systems

In a business context, workflow management systems are employed for the creation, management, and reuse of business processes. One of the first articles elaborating the idea that individuals have private workflows is written by Gary et al. (1997) [Gar+97]. They state that organizational workflows are overlapping with the workspace of an employee and with the implicit personal workflows of a person. They claim that these personal workflows are little regarded considering the value for the organization. Therefore they want to enable employees to develop their own productive work practices in support of an organization's processes. They postulate a more integrated view for the processes of the organization and the personal workflows of the employees.

In 2002, Keller et al. [KSM02] described a personal workflow system. Here a personal workflow management system is a means for an individual to organize the participation in multiple projects. When working in different projects, a person may be confronted with the problem of having to perform several tasks concurrently, while she/he might only work on one task at a time. Social workflows support private persons in a similar way as they can organize parallel executed tasks in a sequential order. In parallel running tasks for private persons can occur if a persons pursues several complex goals within one or several social workflows.

In contrast to the works above, Hwang and Chen (2003) [HC03] define personal processes as processes that constitute personal tasks and data for private individuals. A personal workflow management system (PW-FMS) is a system that coordinates personal processes. Their work is the first, we know of, that explicitly shifts the use of a workflow management system to a non-profit private scenario. In contrast to a traditional workflow management system, which enforces task execution, the main objective of their PWFMS is to remind or provide suggestions to private individuals. In the evaluation of their work Hwang and Chen describe a simple data-driven workflow system to execute personal workflows. Their PWFMS only regards individuals and how they can be supported in their everyday life. Therefore groups or collaboration within communities are not within the scope of their work.

The work of Hwang and Chen is partly closer related to pervasive/ubiquitous computing than to workflow management because not the struc-

ture of workflows is in the focus but the idea of a pervasive/ubiquitous computing environment which is supported by a workflow system.

Pervasive Computing / Ubiquitous Computing

Mark Weiser coined the term ‘ubiquitous computing’ in 1991 [Wei91]. He envisioned a scenario in which the environment proactively interacts with a human and the presence of computers becomes invisible. Pervasive computing is more pragmatic and does not try to hide the presence of computers but rather tries to embed technology or intelligence in small and easy-to-use devices, like smartphones. Therefore the term pervasive computing is more often used by the industry [LM03]. Taking into account these aspects of pervasive computing, it is reasonable that current research on pervasive/ubiquitous computing are up to further include social interaction in their systems [BC09; Jie+12; End+11].

Ben Mokhtar and Capra (2009) [BC09] argue that pervasive computing is unavoidably moving towards “Pervasive Social Computing”, with the pervasiveness of handheld devices and the popularity of social network sites. For them, pervasive social computing aims to take advantage of human social relationships, expressed as social networks, to enable the fulfilment of users’ tasks. Their aim is to assist private individuals in realizing their daily tasks by reasoning about their social links and recommending them to other users that share similar interests. The pursuit of complex private goals is not considered and so there is no flow of activities and there are no dependencies among them.

In 2004 Ranganathan and McFaddin [RM04] stated that private individuals perform different kinds of tasks in order to achieve their goals. They point out that one problem in pervasive computing environments is the discovery and coordination of different web services for achieving the user’s goals. They conceive pervasive computing environments in which persons walk into and the environment interacts with the persons to attain their goals. The environment will have a number of web-services performing various tasks like providing location or map information, giving product or exhibit information, receiving payment, etc. Their approach of private workflows aims at E-commerce scenarios which use predefined knowledge implemented in web services to guide a customer.

Montagut and Molva (2005) [MM05] suggest a distributed/pervasive computation paradigm in which mobile users do not act merely as clients but can also actively participate in the execution of the overall services. They outline technical issues for such a pervasive execution environment like overall execution control and dynamic task assignment. However, their approach omits synchronization issues for parallel executed tasks and constraints for the task execution for reasons of simplicity. The pervasive execution environment has a distributed control, so no single device is in charge of managing the workflow execution and the overall control is assured through the collaboration of the devices involved in the execution of the workflow. Although the authors consider collaboration, no human activities are regarded in their work.

Kramer et al. (2007) [KMH08] developed a context-driven mobile tourist guide. They pursue the idea to have a local guide, who understands the individual interests and timeframe, knows the local situation and gives a personal tour to each tourist. Their system is able to interpret personal interests according to a user profile and has a dynamic knowledge base which uses web services to collect information about sights and places. If a user wants the system to compute a personal tour, the system computes a sequence of attractive places to be visited according to the user's interests, the current GPS position, and a given timeframe. Their approach is similar to the Personal Workflows of Hwang and Chen (2003) [HC03] because the flow of activities is dynamically computed by provided mobile data. However, the scope of this work is limited to travels and tour guides. A further interesting aspect of their work is the evidence that concrete interests or goals of private individuals in the scenario of travels are not predetermined and that tourists need to be inspired to get aware of their own interests.

Urbanski et al. (2009) [Urb+09] propose a distributed workflow system with dynamic, adaptable processes to enact personal workflows (PerFlows) for private individuals. The concept of the PerFlows system is built upon the described scenario of Mark Weiser. A PerFlow allows specifying conditions, e.g. based on location that cause tasks to start, to skip or to abort. Like in many of the above described approaches, the data that is the cause for the execution of the workflow is delivered by mobile devices. PerFlows provides the infrastructure with additional information about the context which triggers the individual tasks. However, the focus of PerFlows is not the procedural knowledge of workflows,

but the creation of an ubiquitous environment backed by a distributed workflow system. This can also be seen in the granularity of the tasks in the example workflow, like ‘Make Coffee’ or ‘Go Home’, which cannot be considered as a complex goal or activity. PerFlows focuses on supporting individuals and not communities or groups. Hence every user needs to model her/his own workflows as collaboration is not considered.

Personal Information/Knowledge Management

Another research field of interest is Personal Information Management (PIM) and Personal Knowledge Management (PKM). This is evident, because PIM aims to support people to have the right information at the right time, in the right place, in the right form, to perform their current activity [JBD01], whereas PKM is a collection of processes that a person uses to gather, classify, store, search, retrieve, and share knowledge in her/his daily activities [Gru06]. PKM is more about E-Learning and about taking personal responsibility in an organization. Hence, a person is enabled to work better with knowledge [Völ10]. It is a bottom-up approach to traditional knowledge management directed at the needs of individual employees [Pol08]. Surprisingly, in none of these two research fields, related work could be found which is applied in a non-profit scenario using procedural knowledge.

Non-Technical Systems

Detached from a certain research area, Dorn et al. (2012) [DTD12] describe flexible social workflows. They address collaborative problem solving and its integration with process-support systems as an architecture comprising human components and connectors. They state that human collaboration consists of human components and human connectors, e.g. a secretary (connector) might respond on behalf of the principal (component). With this assumption they model a process which reflects three collaboration patterns (social networking, wiki creation, and crowdsourcing) based on Web 2.0 applications in order to pursue a complex goal (Writing a scientific project proposal). However, the authors do not propose an execution service or an executable workflow representation. The procedural knowledge of a social workflow is not

regarded but only the collaboration within a dedicated group of persons to pursue a goal.

2.4.2 Workflow Communities

The current state of research has shown that the idea of social workflows, as stated in this work, has not been pursued yet. Of course, there are intersections and similarities to different works and research areas. Especially, knowledge sharing and the idea of building a tied community, linked by their interests, were left beyond. In the following three selected applications are described which have proven a successful implementation of these aspects concerning scientific workflows. The Taverna website⁴⁷ says that “Scientific workflows are widely recognised as a useful paradigm to describe, manage, and share complex scientific analyses”. They are also a means to provide unambiguously outcomes of experiments because the whole experimental setup is repeatable and the workflow serves as execution protocol [Gil+07]. The following three systems show that there is an interest to collect and organize procedural knowledge within a community in a non-business context.

The WINGS⁴⁸ (Workflow Instance Generation and Specialization) [Gil+11] platform is used to create and execute workflows which represent computational experiments. It was initially developed to create large-scale workflows for physics-based simulations to construct seismic hazard maps for the prediction of earthquakes. Additionally, it has been used in several scientific applications including biomedical image processing and genomics. WINGS aims at communities of researchers, so they can use, create, and execute data mining workflows. To assist its users, WINGS uses Artificial Intelligence (AI) planning and semantic reasoners to support users in creating workflows while validating that the software components are properly used and the requirements are fulfilled to execute the components. Besides that, WINGS supports its community in sharing and reuse of their workflows. Therefore, it implements a discovery and exploration service to find suitable experiments.

⁴⁷For more information please refer to <http://www.taverna.org.uk/introduction/why-use-workflows/>.

⁴⁸For more information please refer to <http://wings-workflows.org/>.

CloudFlows⁴⁹ [KPL12] is a cloud-based workflow management system that supports the creation, execution, and sharing of data mining workflows among a community. It is built as a service-oriented architecture that allows integrating third-party services in workflows at run-time. The service discovery of CloudFlows offers a wide range of data mining and utility components for data processing. All user interfaces are web based and also intended to run on browsers of mobile devices. As a social aspect, every registered user may share and reuse workflows of other users by copying a public workflow in her/his private repository.

MyExperiment⁵⁰ [GR07] is a social networking environment for sharing research objects and especially scientific workflows. It offers a collection of scientific workflows which consists most notably of Taverna⁵¹ workflows, which are executable in the Taverna workflow workbench. Taverna was designed from the outset to assist bioinformaticians on their laboratory work. The Taverna workflow system should release bioinformaticians from the burden to repeatedly perform tedious tasks of plumbing together data resources and analytical tools. Nevertheless, the MyExperiment environment supports workflows of various workflow engines. For the community of MyExperiment the workflow execution engine does not matter but the knowledge in the workflows and the used services. For this reason, it offers a market place where users can share, reuse, and repurpose scientific workflows to avoid unnecessary reinvention. To foster collaboration and the community feeling it also integrates communication means to enable users in discussing workflows and their related scientific objects.

2.4.3 Conclusion

This section gave an overview of related work that intersect with the idea social workflows. For this purpose, related work was divided in related research approaches and workflow communities which already employ workflow technology in a non-business context. Works in the field of S-BPM showed that there are endeavours to benefit from the Web 2.0 attitude of people in order to collaboratively shape business processes of higher quality and in shorter time. However, workflow systems mainly

⁴⁹For more information please refer to <http://www.clowdflows.com/>.

⁵⁰For more information please refer to <http://www.myexperiment.org/>.

⁵¹For more information please refer to <http://www.taverna.org.uk/>.

reside in the business (besides the work of Hwang and Chen [HC03]). In research, work on pervasive/ubiquitous computing aim more at private individuals although often driven by commercial interests. The fact that workflow technology is not only employed by companies is underpinned by presenting running communities for scientific workflows. Section 3.4 will summarize the related work presented here and give an overview how each approach intersects with the idea of social workflows.

Foundations for a Social Workflow Platform

Görg, M.S.E.K.

2016, XIII, 295 p. 48 illus., 10 illus. in color., Softcover

ISBN: 978-3-658-13532-4