Activity-centric Support for Weakly-structured Business Processes

Benedikt Schmidt
SAP Research CEC
Darmstadt
62483 Darmstadt
benedikt.schmidt@sap.com

Todor Stoitsev
SAP Research CEC
Darmstadt
62483 Darmstadt
todor.stoitsev@sap.com

Max Mühlhäuser
Telecooperation Group,
Darmstadt University of
Technology
64283 Darmstadt
max@informatik.tu-darmstadt.de

ABSTRACT
Knowledge-intensive tasks are a blind spot for business process management systems, as these tasks are executed in an unsupervised, highly individual manner. Hence, individual experience is not disseminated and task execution largely depends on implicit knowledge.

In this paper we present a framework, realizing situation-specific and personalized task execution support for knowledge-intensive tasks in business processes. As a core concept we suggest activity scheme: a structure capturing a probabilistic task execution model. Activity schemes seamlessly integrate the organizational business process with the individual task execution process based on personalization and generalization of user interactions in the working applications.

Author Keywords
Human-computer interaction, task execution support, knowledge work support.

ACM Classification Keywords
H.4.1 Office Automation: Workflow Management.

General Terms
Human Factors, Management, Performance

INTRODUCTION
The integration of software applications into organizational work execution highly differs depending on the taken perspective. A process-oriented perspective focuses on coherence: organizing, planning, and controlling work plays a central role. Linear execution sequences are modeled and tracked by a system, e. g. a process-aware information system or a workflow engine [26]. On the other hand, the knowledge work perspective stresses contingence: unforeseeable task execution processes supported by applications to structure personal working tasks and acquire, produce or disseminate information in a context-sensitive manner [24]. In fact, knowledge work reveals as an integral part of business processes due to the importance of inter-personal communication and decision-making tasks in business processes [16, 6]: a close connection of both perspectives.

In this paper, we introduce a concept for supporting the execution of knowledge-intensive human tasks in business processes. The concept comprises the extension of a business process management system by activity schemes as a key concept to realize transparency in the actual task execution for situation-specific and personalized user support. Activity schemes are based on and enhanced by actual user interactions in the working applications, which are captured through desktop sensors and add-in components to track user-system interaction.

The activity scheme approach bridges the individual and organizational perspectives through mechanisms for personalization and generalization. For the organization, a generalized activity scheme can be used for planning and controlling a business task by providing a rough overview of the possible activities that need to be performed (on interaction level) to execute that task. For the individual task executor, a personalized and contextualized version of an activity scheme is generated during process execution to enable user-specific and situation-specific task execution support. We suggest that the presented concept of activity schemes can bridge the gap between organizational business processes and individual task execution processes based on experience management.

The remainder of the paper is structured as follows. In the next section, we discuss background concepts for our approach. Then we model the creation and execution of knowledge-intensive tasks in business processes. This is the foundation for our approach, describing an extension for business process management systems through the integration of activity schemes. A preliminary evaluation based on empirical data follows, which shows the feasibility of our approach. We conclude with a summary and discussion of future work.

EICS'10, June 19–23, 2010, Berlin, Germany.

BACKGROUND

To enable support for knowledge-intensive tasks, a thorough understanding of their inherent structure, their emergence and execution is needed in the first place. Therefore our approach firstly focuses on modeling the creation and execution of knowledge-intensive tasks in business processes. Based on the elaborated conceptual model we identify demands for supporting knowledge-intensive tasks and apply the model for task execution support.

Our model is based on task modeling approaches from the domains of task analysis, information management science and business process management. Thereby, we consider a task as a self-contained unit of work, which can be refined through an arbitrary number of subtasks and aims to achieve a certain goal [2, 21, 13, 4]. Although the adopted definition of a task holds for all mentioned domains, the latter consider different explicit task models due to their different scopes. Task analysis describes task execution as interaction between humans and systems, i.e. human intentions result in activities that are supported by a computer system [19]. Information management is interested in the task execution with respect to information demand [13]. Business process management focuses on the integration of tasks in business processes with respect to business process orchestration and workflow execution [6]. To create a model for the execution of knowledge-intensive tasks in business processes we particularly focus on human tasks with respect to their decomposition.

Task Description and Task Execution Process

Task modeling in the domains of task analysis, information management, and business process management [4, 13] follows two major paradigms: attribute-focused and process-focused, also known as task description and task execution process [2].

Task Description: Task description focuses on the definition of a singular unit of work through attributes. These attributes specify requirements, task goal, optionally methods, and task status [2]. Such task descriptions do not focus on the actual task execution process. They provide abstract task models that are useful for the coordination and control of tasks in complex structures. For example, human tasks in business process models belong to this category.

Task Execution Process: The second view focuses on the task execution process by modeling the manifestation of the task in fine-grained actions. Execution process models can be described for automated tasks in business processes as well as for human tasks, performed by individuals. With respect to human tasks, a task execution process is internal to the performer. This process is defined by the human and is subordinate to his comprehension [18].

Task Execution Process Models

The approach presented in this paper focuses on the modeling of a task execution process as an observable sequence of actions to execute a task. Thereby, generally two types of task execution processes are considered: (i) structured task execution processes, where a task has a strictly defined sequence of actions for its execution; (ii) weakly-structured and unstructured task execution processes, where the same task may be performed differently, i.e. through a different sequence of actions.

Structured Task Execution Processes: Structured task execution can be modeled through various task analysis approaches. A common feature of these approaches is the decomposition of tasks into subtasks as a transition from abstract objectives to concrete, executable objectives. One task analysis approach, focusing on the structured decomposition of tasks, is Hierarchical Task Analysis (HTA) [4]. The temporal relation between the subtasks is modeled using plans. The Concur Task Trees (CTT) [19] approach is also based on hierarchical task decomposition. It further allows task classification in different categories (abstract tasks, user tasks, interaction tasks, and application tasks) which directly hint towards a further need of task decomposition or towards the responsible stakeholder for task execution. This approach also allows modeling of constraints like task dependencies and temporal relationships.

Structured task execution can be also modeled by approaches like petri-nets or business process modeling notation (BPMN) from the domain of business process and workflow management [26]. These approaches generally model work through process or workflow diagrams by only including descriptions of system and human tasks, without describing the low-level interactive activities, needed to perform a human task. To realize situation-specific adaptation of the task execution process, change patterns and change support techniques have been introduced [27]. Business rules are another promising approach towards agile adaptation of task execution processes [5]. Current approaches focus on situational adaptation based on influence factors external to the task executor. They do not reflect individual aspects, like information and competence demand, as e.g. modeled by KMDL (Knowledge Modeler Description Language) [8]. The idealized assumption of structured task execution processes is especially useful for system analysis and for organization of production-like processes, which are not subject to frequent change. Flexibilization of these processes is an important topic, requiring high modeling effort.

Unstructured Task Execution Processes: Unstructured task execution processes relate to knowledge-intensive human tasks. Models for such task execution processes need to reflect situation-specific and task performer specific information. Therefore, the elaboration of such models is very complex and they can be considered valid only at very abstract or probabilistic level.

Two highly abstract approaches for modeling such task execution processes are GOMS (Goals, Operators, Methods, and Selection rules) [4] and activity theory [18]. GOMS reflects the task performer specific information for the execution process by specifying the procedural knowledge that a user must have in order to carry out tasks on a device or system. Activity theory considers the context in which task execution is performed [18]. Activities as motives are set into an execution context of influencing factors. The context, which sets the subject in relation to its objective, is formed by elements such as rules, community, division of labor, and instruments. The context connects the task performer with the situation and the execution process towards a given goal.

A similar reflection of the task performer and the situation, in which a task is executed, exists also in information science. Byström [2] describes the unstructured execution as cyclic process of task construction, task performance, and task completion, influenced by individual and situational attributes. Due to the complexity, realized systems generally simplify the model. Many systems for unstructured task execution support consider tasks on system level as a collection of resources for recommendation. One example for such an approach is activity-centric computing [17], where task-cases can be directly reused for recommendation using case-based reasoning [24]. Approaches like Task Pattern for task management systems only model task execution for specific parts of a process [22]. To include the process and provide variability, many approaches consider the process as probabilistic model of connected resources and executions, which is appropriate for machine learning [20, 14, 24].

Ontologies for Unstructured Task Execution Processes

None of the approaches discussed above provides a taxonomy or a model to describe the actual actions in the task execution process in a reusable form. We follow the knowledge work description of [21] as a semiotic activity of information creation, transformation, and consumption. Knowledge work often is executed using computer desktop applications, which deliver a restricted set of functionalities [19]. For this understanding of knowledge work, Hädrich identifies a set of eight reoccurring knowledge work actions as fragments of work [9]. Each knowledge action is an abstraction from the actual execution process. These knowledge actions are: authoring, co-authoring, training, acquisition, update, feedback, expert search, and invitation. Each knowledge action contains similarities which are independent from the specific task context, e.g. authoring always involves the creation or transformation of symbols. An individual knowledge-intensive task is executed as a combination of knowledge actions.

We suggest that a knowledge-intensive task execution process can be described by means of knowledge actions. Thereby, each knowledge action contains a collection of small workflows of fine-grained operations on interaction level. We model such an operation as a triangle of activities, resources, and applications: Activities are performed on resources by using applications. Applications are software tools or software services. Resources are objects which are used during the execution of knowledge work. Examples for resources are documents, websites, e-mails or an address book. To describe the connection between activities, applications and resources we use the following taxonomies and ontologies:

- Task, knowledge action, activity taxonomy: A hierarchy of verbs for task decomposition.
- Resource ontology: Classified resources which occur in office work and their relation.
- Application taxonomy: Classes of software applications.

Connections between these taxonomies describe which activities can be performed on which resources by which applications. In the following we will give an overview of this knowledge base. Currently we have an initial version of the taxonomies and ontologies, which undergoes a validation process.

Task, Knowledge-Action and Activity Taxonomy: The task, knowledge-action and activity taxonomy provides a vocabulary of verbs to describe task execution. Generally taxonomies to describe task execution are collections of verbs describing very specific activities [10]. Although, different taxonomies for reuse beyond a specific domain exist (c.f. [1]), we see knowledge work task taxonomies as an open topic. We assume that knowledge work tasks are made up of a discrete amount of recurring execution steps which are adopted in a situation-specific manner. We describe a knowledge base for the modeling of task execution processes on three abstraction levels: (i) the abstraction level of a business process task; (ii) the abstraction level of knowledge actions; (iii) the abstraction level of actual interactive activities. These abstraction levels are interconnected in the sense of hierarchical task decomposition, as described in the context of structured task execution processes. For example the task "set up meeting" can be considered as abstract activity description, which is realized by a sequence of knowledge actions (e.g. "organize", "communicate") that can be directly realized by operations on different applications (e.g. the actual booking of a room or the writing of invitation emails).

The three levels of the taxonomy have different foundations. The highest level for tasks is based on a taxonomy of activity verbs used to describe business process tasks [15]. [16] succeeded in assigning 20.000 business tasks to this taxonomy. The second level uses the knowledge actions by [9] in a reworked and extended form, e.g. using [12]. The lowest activity level models activities of user-system interaction. The resulting ontology classifies activity verbs on different abstraction levels. Verbs standing for desktop work activities are connected with knowledge actions and those are connected with verbs for business process task descriptions. The resulting structure is comparable to a structured decomposition of a task with respect to verbs standing for generic activities.

Resource ontology: The resource classification is a structured net of information objects which represent the knowledge space that can be accessed during the work of an individual. Activities are executed by consuming, transforming or creating information objects. Our main goal has been the structured description of individual and organizational information objects used for work execution. A Personal Information Model (PIMO) concept has been introduced in [23], which includes all the data that is needed by an individual to perform knowledge work; independently from the way the data is accessed. Hence, all different resource types, respectively the concepts they stand for, can be modeled using PIMO.

We have extended the structure of PIMO to reflect the different abstraction levels included in the task, knowledge action, activity taxonomy. We use a construct called placeholders as variables for resources. Placeholders enable information reuse based on generalization from specific situations. For example, a placeholder can be people involved in the process which is instantiated based on the process runtime data. By connecting elements of the task, knowledge-action and activity taxonomy with elements in the resource ontology it becomes visible which activities can be performed on which objects.

Application taxonomy: Applications play a major role in the described data model. A worker has access to certain applications to execute his working tasks. A classification of applications allows the reusable assignment of activities to the applications they are performed with. There is no integrated application which covers all working needs. Work activities are generally spread over numerous applications. Therefore, it is necessary to specify which activity can be performed by which application. Recently [7] proposed a software type taxonomy which is the foundation of our work. We have selected applications which are used in office work settings. This includes application classes like authoring applications (word processing, spread sheet, etc.), communication applications, business information systems and web applications. The application classes have been connected to the activities they provide and the resources they are capable to create or transform.

A TASK EXECUTION MODEL FOR KNOWLEDGE-INTENSIVE TASKS IN BUSINESS PROCESSES

Our work focuses on knowledge-intensive tasks in business processes, by considering the organizational business process management, as well as the individual task execution. The discussed model is shown in Fig. 1 and addresses human tasks that are integrated in business processes. Thereby, task descriptions based on attributes are created during the composition of the business process models. At process runtime, tasks from the business process model are instantiated and delegated to given roles (Fig. 1, Organizational Level). The delegation initiates the personal task execution process (Fig. 1, Individual Level).

The delegation triggers individual choice for executing a task as commitment by intention [3]. The commitment leads to a task execution process following Byströms [2] execution model. Within this process individual task decomposition is triggered and new task execution processes are generated. We model the execution of a task and all generated subtasks through knowledge actions. Thereby we explicitly describe the execution of a knowledge action as a sequence of activities on resources using applications. Activities are the execution of functionalities on applications, i.e. these are interactive activities on software application level. Each knowledge action includes a different amount of singular activities

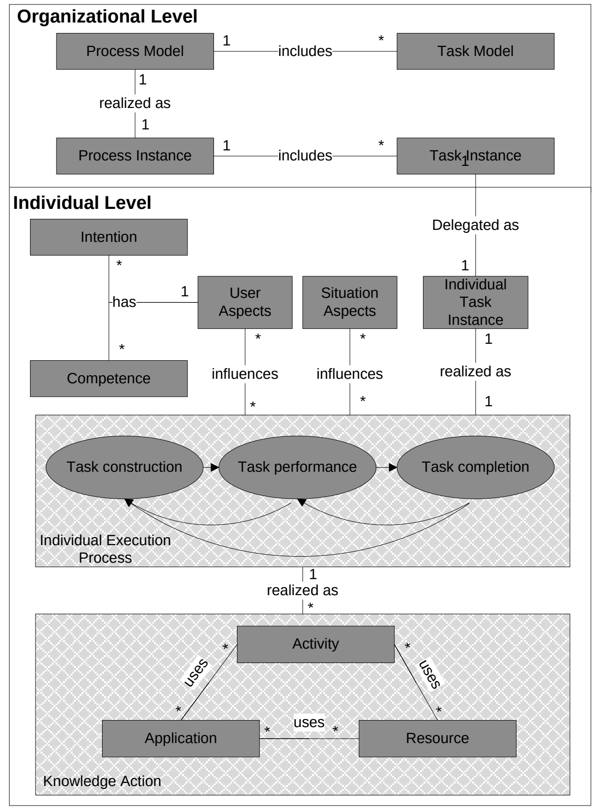

Figure 1. Model for Cross-Application Tasks

which are executed on different applications and different resources. These are influenced by the user environment, the user profile, and situation-specific information. The model shows that support for knowledge-intensive tasks needs to consider the transition from task descriptions to a weakly structured task execution processes influenced by individual and situational factors. This meta-model allows us to understand the interplay between structured organizational business processes and individual task execution processes. It can be used for the conceptual design of business process management systems, as it extends human task definitions with entities that support the execution of such tasks at system interaction level. In the following, we work on the question, how an appropriate externalization of tasks might look like, how it reflects individual and situational influence on the task and how instances of such knowledge can be collected in a format, enabling work execution support.

APPROACH

We propose a concept to support knowledge-intensive tasks in business processes. Our concept is based on the task execution model discussed in the previous section (see Fig.1).

System Concept

Our approach integrates the creation and enhancement of reusable task execution knowledge into business process execution. Thereby, the approach reflects the weak structure of the tasks in focus, as well as situational and personal influence on the task execution process. The structure to support the execution of knowledge-intensive tasks in business processes is called activity schemes. We propose the integration of activity schemes into a generic business process management system landscape, including process modeling, process controlling, process runtime, and process execution environment. Through this we intend to blend task delegation and user support: e.g. delegation of a task to decide on creditworthiness transfers additionally an activity scheme which hints to the actual task execution process without prescribing this process in a strict manner. For enabling the use of activity schemes a process execution environment needs to offer functionalities for activity tracking, reasoning, and user model generation. A process runtime needs to be extended through delivery of task execution information in form of activity schemes. On the other hand, the process modeling environment needs to provide functionalities to assign activity schemes to tasks and to access activity scheme information.

Activity Schemes

Actual task execution activities are captured by reusable activity schemes. In the following, we give an overview of the structure of activity schemes and describe processes of activity scheme generalization and activity scheme personalization to enable reuse of these schemes in the community of task performers.

Activity Scheme Structure

An activity scheme is a directed graph of nodes, which stand for knowledge actions. Recall that we consider a knowledge action as a sequence of activities on resources using applications. Knowledge actions can be included several times, e.g. if information search for different topics is necessary. The knowledge action nodes include the relation between activity, application and resource.

- Activities: They can be directly executable or can consist of a small workflow of minor activities.
- Applications: A list of involved applications. Thereby, an abstract category of an application (e.g. word processing software) or a direct instance (e.g. Word) can be included.
- Resources: Information objects which stand for documents, e-mail contacts, etc. Next to instances of such objects, a predefined set of placeholders exists (e.g. "persons involved in process instance"). The placeholders stand for information objects which are only accessible during the actual execution of a task. When a task is delegated the placeholders shall be replaced by actual information objects.
- Edges: the directed graph edges of an activity scheme are labeled with the transition probability from one objective to another objective, i.e. from one knowledge action to other knowledge action. Each activity scheme is connected to a human task.

This structure uses the initially discussed taxonomies. As activity schemes have a personal and an organizational facet, the resource and application taxonomy exist in a personal and in a shared version. Once personal information is shared, the respective taxonomy instances get included in the shared taxonomy, if necessary.

Activity Schemes for Working Tasks

We use business process tasks as carriers for generic and reusable task execution knowledge in the form of activity schemes. An activity scheme is a generic model of the activities involved in the execution of a task (e.g. a task "create report" includes the creation of an object and the search for information). A scheme is no strict plan but a generic description of involved activities. In fact, an activity scheme is a graph with weighted edges, connecting activities with classes and instances of resources and applications. An activity scheme enables two main capabilities:

- For an organization activity schemes give access to the relevant execution data on performed activities, unfolding information demand, involved information, and execution times. Thus, activity schemes enable mechanisms for organizational governance, increase task execution transparency and enable detailed best-practice management in the existing business process management infrastructure.
- For the individual task execution activity schemes are combined with observed user activities, user model, taxonomies of resources activities and applications, and used to generate situation-specific task execution support.

Activity Scheme Lifecycle

The lifecycle of activity schemes involves the following phases: creation, update, transformation, and use. These phases are shown in Fig. 2 and discussed in the following. Later on an example is given.

Creation: Currently activity schemes are identified and created based on activity logs generated by desktop sensors. This means, involved activities, applications, and resources are clustered to knowledge actions and integrated into a scheme, based on the transition probability between the different knowledge actions. This is performed through domain experts with IT background by manual data analysis. The manual creation process is intended to prove the feasibility of activity schemes in the first place. As a next step, we want to create activity schemes automatically, using machine learning algorithms. Based on a training phase, an engine will generate an activity scheme for a task.

Transformation: Once an activity scheme is used, it is personalized based on information from the process instance and the user model. This is described in more detail in the next section.

Use: The activity stream of the user is captured using desktop sensors as described in [14, 20]. Based on the working activities, the system tries to identify the active activity scheme that guides the work. Therefore, a distance-measure between the working activities and the scheme is calculated. This distance is based on the activities performed using applications and on analysis of the texts visible on the computer screen. Once an activity scheme is activated, the current position within the scheme gets identified (e.g. the knowl-

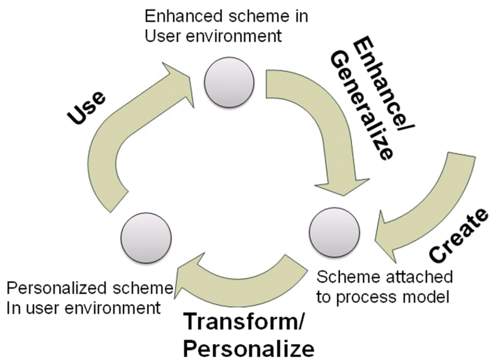

Figure 2. Activity Scheme Lifecycle

edge action "authoring"). Subsequently, the transition probabilities are used to generate the execution proposals for the user (e.g. a knowledge action "dissemination" to distribute the authored product). Depending on the user activities, the position in the scheme gets permanently recalculated and proposals are updated. As long as a scheme is activated, it can also be extended with new nodes based on the user activities.

Enhancement: During the execution process, the locally activated activity scheme gets extended by the new data from the task execution process. Thus, during the use, an enhancement of the local scheme is realized. Once the user finalizes a task, the system asks the user if the changed local activity scheme including information about the execution activities, execution time, etc. can be sent to the central system in an anonymized form. Based on such execution data, the activity scheme attached to the human task in the process model gets enhanced. Thereby, explicit data gets generalized and placeholders for specific resources are generated.

Lifecycle end: The lifecycle assumes the stabilization of activity schemes over time, a process which can be observed in many structures that are based on user generated input, e.g. tags. This assumption needs to be confirmed. If a continued execution of the lifecycle is undesired, activity schemes can be integrated into a governance framework to deny changes of mature models through an organizational entity.

Generalization and Personalization of Activity Schemes

The generalization and personalization of activity schemes is a core demand for their situation- and person-specific utilization. Therefore two mechanisms exist. Activity schemes are transformed based on situation- and person- specific information, once a task is delegated to a user. Transformation means that resources and applications assigned to knowledge actions are transformed based on a placeholder concept:

- Resource-Placeholders: Resources involved in an activity can be summarized through placeholders, which enable abstraction from situation-specific or individual information. Placeholders stand for data which is only known at the execution time, e.g. "output of previous process task".
- Application-Placeholders: Classes of applications are applicable to similar activities. Thus, application-placeholders abstract from actual applications, by e.g. using "Word Processing Software" instead of Word.

The transformation of activity schemes is based on the generation and delegation of tasks in the business process execution environment, resulting in three different activity scheme states. The process is visible in Fig. 3. In the following, we introduce the different activity schemes and their positions in the personalization and generalization process:

- *Activity scheme models* are attached to a task in a process model. The activity scheme model has the highest abstraction degree. Those elements of the scheme, which depend on task instance data and on individual preferences, are covered by placeholders. In the personalization process the activity scheme model is the starting point (A1 to A2). On the other hand, in the generalization process this model is the end point, resulting after process instance data is removed from the activity scheme (B2 to B3).
- *Activity scheme instances* result from activity scheme transformation based on process instance data. In the personalization process placeholders for process instance data get replaced by specific information objects (A2 to A3). In the generalization process, the instance data is identified and removed and a respective placeholder gets attached to the activity scheme (B2 to B1).
- *Activity scheme personalization* in the personalization process results from activity scheme transformation based on the PIMO and on mapping the user application landscape to the application taxonomy. The PIMO includes information on the resources, the user can access. The application taxonomy for the user specifies the applications which can be used to execute the activities (A3). In the generalization process specific applications are replaced by generic applications (Powerpoint is replaced by Presentation Software) and individual user information is removed or a respective placeholder is created.

Activity Scheme Example

In the following we will give an example for a simple activity scheme. In a knowledge transfer process the task "Create one slide presentation on generics in java" exists. An activity scheme for this task includes four knowledge actions: Authoring, Dissemination, Data Search, Expert Search. The activity scheme is visible in Fig. 4. The nodes stand for the involved knowledge actions, specifying the triangle of activity (ACT) performed with application (APP) on resource (RES). For each node the respective transition probabilities to the other knowledge action are visible.

Initially, the generic activity scheme attached to the task in the process model is transformed, based on process instance information and on the user model. We want to discuss this transformation process for one resource and one application attached to the knowledge action "dissemination" included in the activity scheme (see Table 1). The "dissemination" stands for delivery of a presentation authored during the task

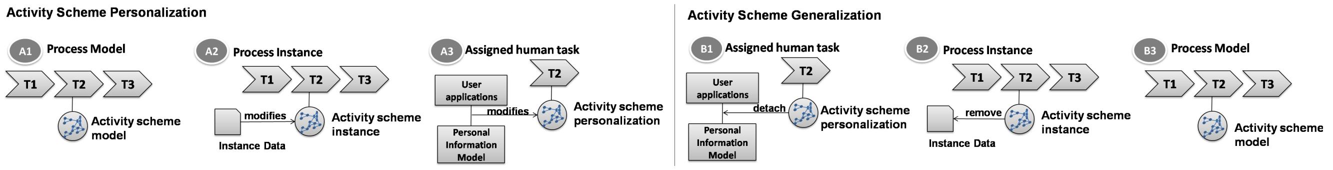

Figure 3. Activity Scheme Transformation

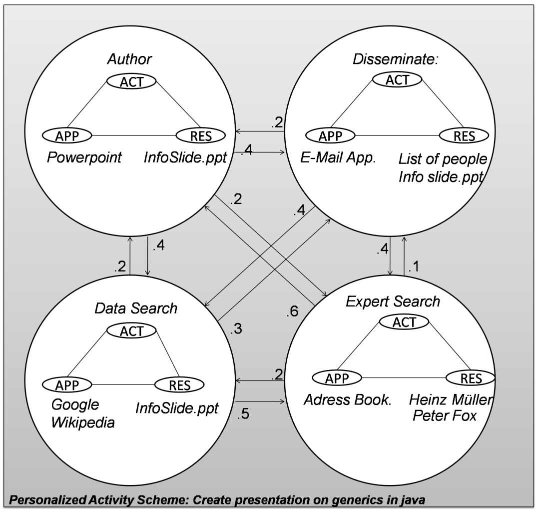

Figure 4. Activity Scheme for "Create presentation on generics in Java"

execution process to all people involved in the overarching knowledge transfer process. In the activity scheme model, the respective resource is realized by a placeholder for all people involved in the process and the specification of an e-mail application. Once the activity scheme instance is generated, the placeholder for all people involved in the process can be instantiated based on the actually assigned people. Once the personalized activity scheme gets created, the e-mail application can be changed based on the user system which integrates Outlook as e-mail application.

Knowledge Action: Dissemination	**Activity Scheme Model**	**Activity Scheme Instance**	**Personalized Activity Scheme**
Resource	Placeholder: Persons Involved in Process	Helmut Gartner, Sina Telde	Helmut Gartner, Sina Telde
Application	Placeholder: E-Mail Software	Placeholder: E-Mail Software	MS Outlook

Table 1. Transforming Activity Scheme Knowledge Action

The user works with the personalized activity scheme. This means that a distance between his activities and the given activity scheme is calculated during the work process. After undergoing a threshold the scheme gets activated. If the user has opened Outlook, after previously "authoring" a Powerpoint file, having keywords extracted similar to the topic of the authoring knowledge action, the dissemination" knowledge action gets active. Considering this position in the graph, activity recommendations are calculated based on the transition probabilities to other nodes.

Characteristics, Threats and Challenges of the approach

In the following we compare our approach with existing systems and, point out the specific focus of our work, compared to related literature. Additionally, we show weaknesses of our approach and methods to deal with these weaknesses.

Structured task execution approaches: Our approach presupposes the existence of a business process definition, including knowledge-intensive human tasks. Thereby, a strictly prescribed execution process for knowledge-intensive tasks is not in focus of the presented approach and it does not aim at streamlining knowledge-intensive tasks in a strictly structured manner. This distinguishes our work from the work on change patterns [27] or business rules [5] which rely on a higher organizational control and intervention in the task execution process and consequently require higher modeling effort. The task execution environment exists in the perspective of our approach as diverse applications on a user desktop, thus having generalization on application level in focus. This is different from solutions which directly provide user interfaces for business processes based on task analysis [25].

Weakly-structured task execution approaches: Our work focuses on weakly structured task execution. Approaches for unobtrusive information detection based on desktop sensors as discussed in [20, 14, 24] are highly relevant in this research area. These approaches follow the idea of plan recognition [11] based on the user activities. The user goal is deduced and user support is generated based on existing plans. Plans generally contain the actual activities. In contrast we have introduced knowledge actions as an additional decomposition layer which enables generalization and personalization of task execution knowledge. Our approach extends the business process-centric perspective on work execution and suggests infrastructural extensions for business process management systems to support user-centric assistance.

Threats and challenges: Generalization and personalization of task execution processes are complex system tasks which heavily rely on a rich and appropriate knowledge base. For the generalization we use knowledge actions with placeholder representations based on our resource and application taxonomy. We cover those elements in the knowledge base we consider essential for generic knowledge work. Still, extensions of the knowledge base probably will be necessary. Another difficult aspect is the generalization and personalization process: we try to segregate individual from collective facets of work. The necessary information, again, is element of our taxonomies to describe user competency and the user's system environment. Here, again, the knowledge base limits the functionalities. We have addressed these challenges by founding our work on existing studies in the addressed research field and will further improve the knowledge base in iterative tests to mitigate the threats. Apart from these, rather system specific, aspects our approach and the founding model rely on several major assumptions. In the following, we highlight those assumptions and show their justification in an evaluation study.

PRELIMINARY EVALUATION

The Activity Scheme approach focuses on the creation of interpersonal, reusable descriptions of task execution processes. Activity Schemes are composed of connected knowledge actions, which enable generalization from execution sequences of interactive activities for interpersonal exchange of task execution knowledge and personalization for user-specific task execution support. This approach contains certain assumptions which require validation beyond the initially given model. The assumptions can be summarized as follows:

Assumption 1: Knowledge work tasks, executed by different people result in different execution sequences of interactive activities. In other words, we need to show that giving the same task to different people results in different task execution processes.

If that holds true, discussing eligibility of Activity Schemes needs to focus on validation of implicit assumptions concerning knowledge actions:

Assumption 2: Knowledge actions exist as self-contained units of work in work processes. Thus, they make the task execution processes of similar tasks comparable. In other words, we need to show that abstraction from resources and applications used in the work process is necessary to make the process comparable by preserving significance. As a result, we need to show that the same task, executed by different individuals always contains the same knowledge actions.

Assumption 3: Work sequences of similar tasks can be modeled as transitions of knowledge actions. In other words, we need to show that the differences of the execution sequences can be modeled as a recombination of knowledge actions.

Setting

The task execution processes were accessed as activity logs, which have been created through desktop sensors, capturing user interaction with a desktop computer system and the included applications. We have used the Matthäus1 data set [20] which includes activity logs of 20 users executing 12 different knowledge work tasks, leading to 240 logs. The included tasks differ with respect to complexity. Examples for tasks are e.g.: "Create UML diagram.", "Prepare a handout on a given topic." or "Develop a Hello World Java prototype.". The tasks were executed on a similar desktop system.

Data Evaluation

Our evaluation focused on the three aspects described: similarities in the execution processes of similar tasks. To identify this, we compared the respective 20 activity logs for a task with respect to different aspects as visible in Fig. 5. As we rely on operations we use the term sequence for the observed activities.

Similar task execution process: (checks assumption 1) Similarity means exactly the same task execution process, as a sequence of similar interactive activities on similar resources with similar applications. Simple tasks, like "Do Inventory Update" or "Visualize research results as graph" were executed very straightforward and many task execution processes were similar (Fig. 5, topmost graph). For tasks of higher complexity, only few similar processes could be identified. For instance, this holds true for the task "Translate executive summary" which was highly dependent on the individual language skill and preferred application for translations (e.g. Babylon , Dict.Leo or Babel Fish[1]). The task "Do budget calculation" showed two different execution sequences for persons who preferred to calculate in Excel or such, who would combine a word processing software with a calculator application.

In a nutshell, the results show that knowledge-intensive tasks are executed using more than one application and with information demand that strongly depends on the actual task performer, which finally results in different task execution processes.

Similar Knowledge Actions: (checks assumption 2) We compared task execution processes without respecting the sequence of interactive activities. For each task, we counted the number of executions which used exactly the same resources and exactly the same applications, independently from the sequence of interactive activities (Fig. 5, middle graph). Only few tasks included similar resources: generally people used different resources, though the content was comparable (e.g. the task "Create UML diagram" used resources about UML by Wikipedia[2] or by an online published UML book). The number of similar applications was high, as the tasks were executed using a similar desktop environment. Interestingly the participants used different applications for the tasks "Do budget calculation" and "Create UML diagram". This was due to the lack of a shared application purpose understanding and varying technical background (e.g. UML diagrams were created using Paint, Eclipse and Powerpoint). We see that resources and applications can be different, although they are used for similar work. In a next step we clustered the activities on resources using knowledge actions. Knowledge actions generalize from resources and applications based on the knowledge base and the similarity of resources. The review of applications and resources hinted to a shared understanding of a task on activity level. Knowledge actions deliver the required abstraction from the activities. A knowledge action "information search" captures the information search independent from

[1]http://www.babylon.com/,http://dict.leo.org/,http://babelfish.yahoo.com/
[2]http://www.wikipedia.org/

a use of Google or Yahoo search. On the other hand the different applications like Paint, Eclipse and Powerpoint are all described in a knowledge action “graphical authoring” . Clustering the activities based on knowledge actions leads to an increased equivalence of tasks, as they include similar knowledge actions. In fact, only three of the evaluated twelve tasks differed with respect to the included knowledge actions. Thereby, from our point of view, knowledge actions do not result in a loss of explanatory power, as they describe a task execution in an easy to grasp manner.

Similar Knowledge Action Sequence: (checks assumption 3) As knowledge actions showed an overlap between different task executions, we subsequently focused on the knowledge action sequence. In our test set, only 50% of the knowledge actions were executed in exactly the same sequence of applied knowledge actions (Fig. 5, lower graph). Direct comparison of the knowledge action sequences shows that often task entry points are different. E.g. “Create presentation on generics in java” is executed by many persons though an initial “information search”, followed by an “authoring” knowledge action. Others prefer to initially author, switch to “information search” and go back to “authoring”.

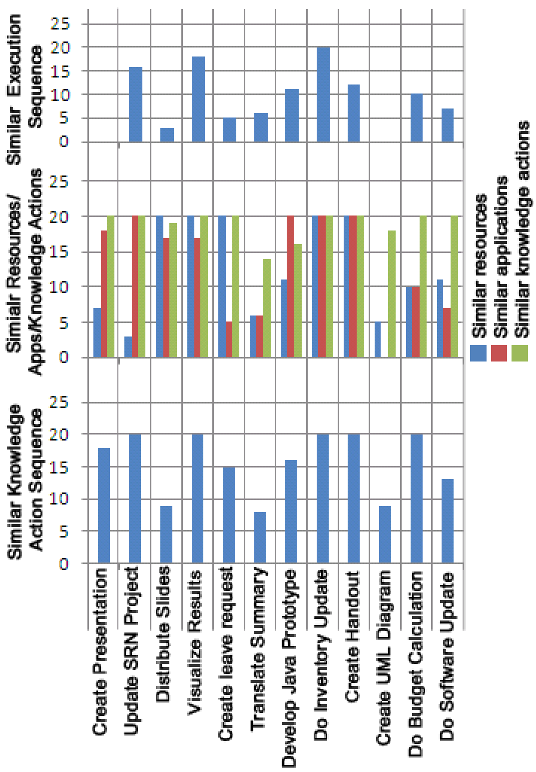

Figure 5. Comparing Similar Elements in Tasks

Evaluation Result

The evaluation of the data set showed that the execution sequences of interactive activities within knowledge-intensive tasks highly vary. The results show that most tasks involve similar knowledge actions, independently from the task performer. This shows that knowledge actions provide a useful abstraction from individual task execution processes, as they remedy individual and situational factors in the generalization process. As the sequences of knowledge actions for a similar task varies among the participants, a formalized process is inadequate. Activity Schemes, creating sequences between knowledge actions based on probabilities, enable dynamic task execution support. We suggest an approach for enhanced user support and organizational transparency and dissemination of task execution knowledge through activity schemes as probabilistic structures based on an evolving set of knowledge actions. The outcome of the study underlines the feasibility of this approach.

The assessment of knowledge actions focused on the generalization process to identify similar core execution structures for knowledge-intensive tasks. An open topic remains the personalization of the resulting activity schemes based on user models.

CONCLUSION AND FUTURE WORK

We have proposed an approach for supporting users in the execution of knowledge-intensive tasks in business processes and for creating and disseminating information about this execution on organizational level. Thereby, the task execution process has been characterized as structured by knowledge actions which are realized by activities on interaction level in different applications. Activity schemes are the central structure to capture and support task execution processes. Activity schemes enable generalization of task execution knowledge in an organizational community on the one hand and personalization for user support, which is tailored to the individual preferences, on the other.

We have described the process of activity scheme utilization, the involved taxonomies and ontology, and the structure of activity schemes. Activity schemes bring additional controlling capabilities into the organizational infrastructure for business process management and enable community-based task execution support through activity recommendation. Currently, we investigate the applicability of activity schemes to a variety of knowledge-intensive tasks of higher complexity. During this process we further validate and extend the taxonomies and ontology that we use for the activity scheme transformation.

In our future work, we will integrate the support mechanisms based on activity schemes in an environment for task execution in the context of organizational business processes. We further want to evaluate the possibility to generate initial activity schemes based on existing activity schemes and the information stored in process models.

ACKNOWLEDGMENTS

The work presented in this paper was developed in the context of the project *Allianz Digitaler Warenfluss* (ADiWa) that is funded by the German Federal Ministry of Education and Research. Support code: 01IA08006.

REFERENCES

1. A. Becks, C. Seeling, and S. Birlinghoven. A Task-Model for Text Corpus Analysis in Knowledge Management. *Machine Learning*, (July), 2001.
2. K. Byström and P. Hansen. Conceptual framework for tasks in information studies. *Journal of the American Society for Information Science and Technology*, 56(10):1050–1061, 2005.
3. P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261, 1990.
4. D. Diaper. *Understanding task analysis for human-computer interaction*, chapter 1, pages 5–47. Laurence Erlbaum Associates, London, 2004.
5. T. V. Eijndhoven, M.-E. Iacob, and M. L. Ponisio. Achieving Business Process Flexibility with Business Rules. *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, pages 95–104, 2008.
6. M. J. Eppler, P. M. Seifried, and A. Röpnack. Improving knowledge intensive processes through an enterprise knowledge medium. *Proceedings of the 1999 ACM SIGCPR conference on Computer personnel research - SIGCPR '99*, pages 222–230, 1999.
7. A. Forward and T. C. Lethbridge. A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering. *ACM*, pages 1–13, 2008.
8. N. Gronau, R. Korf, and C. Müller. KMDL-Capturing, Analysing and Improving Knowledge-Intensive Business Processes. *Journal of Computer Science*, 4(4):452–472, 2005.
9. T. Hädrich. Situation-oriented Provision of Knowledge Services. *Information Systems*, 2008.
10. D. Hu and Q. Yang. CIGAR: Concurrent and interleaving goal and activity recognition. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)*, volume pages, pages 1363–1368, 2008.
11. H. Kautz. A formal theory of plan recognition and its implementation. *Reasoning about plans*, 125:1–67, 1991.
12. B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. The University Press of Chicago, Chicago, 1993.
13. Y. Li and N. Belkin. A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management*, 44(6):1822–1837, 2008.
14. R. Lokaiczyk and M. Goertz. Extending Low Level Context Events by Data Aggregation. *Proceedings of I-KNOW 08 and I-MEDIA '08*, pages 118–125, 2008.
15. T. Malone, K. Crowston, and G. Herman. *Organizing Business Knowledge*. MIT Press Cambridge.
16. J. Mendling and J. Recker. Towards systematic usage of labels and icons in business process models. In *Proc. of EMMSAD*, pages 1–13, 2008.
17. T. Moran. Unified activity management: explicitly representing activity in work-support systems. In *Workshop on Activity: From a Theoretical to a Computational Construct*, volume 1, pages 1–10. Citeseer, 2005.
18. D. Mwanza. Where theory meets practice: A case for an Activity Theory based methodology to guide computer system design. In *Proceedings of INTERACT*, pages 9–13, 2001.
19. P. Pyöriä. The concept of knowledge work revisited. *Journal of Knowledge Management*, 9(3):116–127, 2005.
20. A. S. Rath. UICO: An ontology-based user interaction context model for Automatic Task Detection on the Computer Desktop. *CIAO '09: Proc. 1st Workshop on Context, Information and Ontologies*, pages 1–10, 2009.
21. R. B. Reich. *The work of nations - Preparing Ourselves for 21st Centurv Capitalism*. Vintage Books, 1992.
22. U. Riss, B. Schmidt, and T. Stoitsev. Task Patterns in Collaborative Semantic Task Management as Means of Corporate Experience Preservation. In *Workshop on Knowledge and Experience Management*, pages 40–47, 2009.
23. L. Sauermann, L. Van Elst, and A. Dengel. Pimo-a framework for representing personal information models. *Proceedings of I-Semantics*, 7:270–277, 2007.
24. S. Schwarz. A context model for personal knowledge management. *LNCS*, 3946:18–33, 2006.
25. K. Sousa, H. Mendonca, and J. Vanderdonckt. Addressing the impact of business process changes on software user interfaces. *2008 3rd IEEE/IFIP International Workshop on Business-driven IT Management*, pages 11–20, April 2008.
26. W. van der Aalst, A. Hofstede, and M. Weske. Business process management: A survey. *Lecture Notes in Computer Science*, pages 1–12, 2003.
27. B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support featuresenhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, 66(3):438–466, 2008.