



**Formal Integration of Individual
Knowledge Work and Organizational
Knowledge Work with the Core Ontology
*strukt***

Daniel Eißing
Ansgar Scherp
Steffen Staab

Nr. 10/2011

**Arbeitsberichte aus dem
Fachbereich Informatik**

Die Arbeitsberichte aus dem Fachbereich Informatik dienen der Darstellung vorläufiger Ergebnisse, die in der Regel noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar. Alle Rechte vorbehalten, insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

The “Arbeitsberichte aus dem Fachbereich Informatik“ comprise preliminary results which will usually be revised for subsequent publication. Critical comments are appreciated by the authors. All rights reserved. No part of this report may be reproduced by any means or translated.

Arbeitsberichte des Fachbereichs Informatik

ISSN (Print): 1864-0346

ISSN (Online): 1864-0850

Herausgeber / Edited by:

Der Dekan:

Prof. Dr. Grimm

Die Professoren des Fachbereichs:

Prof. Dr. Bátori, Prof. Dr. Burkhardt, Prof. Dr. Diller, Prof. Dr. Ebert,
Prof. Dr. Furbach, Prof. Dr. Grimm, Prof. Dr. Hampe, Prof. Dr. Harbusch,
jProf. Dr. Kilian, Prof. Dr. von Korflesch, Prof. Dr. Lämmel, Prof. Dr. Lautenbach,
Prof. Dr. Müller, Prof. Dr. Oppermann, Prof. Dr. Paulus, Prof. Dr. Priese,
Prof. Dr. Rosendahl, jProf. Dr. Scherp, Prof. Dr. Schubert, Prof. Dr. Staab, Prof. Dr.
Steigner, Prof. Dr. Sure, Prof. Dr. Troitzsch, Prof. Dr. Walsh, Prof. Dr. Wimmer, Prof.
Dr. Zöbel

Kontakt Daten der Verfasser

Ansgar Scherp, Daniel Eißing, Steffen Staab

Institut WeST

Fachbereich Informatik

Universität Koblenz-Landau

Universitätsstraße 1

D-56070 Koblenz

E-Mail: scherp@uni-koblenz.de, eissing@uni-koblenz.de, staab@uni-koblenz.de

Formal Integration of Individual Knowledge Work and Organizational Knowledge Work with the Core Ontology *strukt*

Ansgar Scherp^{1, 2}, Daniel Eißing¹, and Steffen Staab¹

¹Institute for Web Science and Technologies and

²Institute for Information Systems Research

University of Koblenz-Landau, Germany

{scherp, eissing, staab@uni-koblenz.de}

Abstract. Expert-driven business process management is an established means for improving efficiency of organizational knowledge work. Implicit procedural knowledge in the organization is made explicit by defining processes. This approach is not applicable to individual knowledge work due to its high complexity and variability. However, without explicitly described processes there is no analysis and efficient communication of best practices of individual knowledge work within the organization. In addition, the activities of the individual knowledge work cannot be synchronized with the activities in the organizational knowledge work.

Solution to this problem is the semantic integration of individual knowledge work and organizational knowledge work by means of the pattern-based core ontology *strukt*. The ontology allows for defining and managing the dynamic tasks of individual knowledge work in a formal way and to synchronize them with organizational business processes. Using the *strukt* ontology, we have implemented a prototype application for knowledge workers and have evaluated it at the use case of an architectural firm conducting construction projects.

1 Introduction

There is an increasing interest in investigating means for improving quality and efficiency of knowledge work [10], as knowledge has become a critical factor for the innovation and competitiveness of modern organizations. An established means for improving efficiency of organizational knowledge work is expert-driven business process management [4]. The implicit procedural knowledge found within the organization is made explicit by defining and orchestrating corresponding business processes (cf. [23]). By this, procedural knowledge of the organization is explicitly captured and made accessible for analysis, planing, and optimization. Individual knowledge work is present in domains in which acquiring and applying new knowledge plays a central role. It can be found, e.g., in research, finance, and design [21]. Due to its complexity and variability, individual knowledge work typically is not amenable to planning. In addition, activities

of individual knowledge work that occur only rarely do not justify the effort of business process modeling. However, without explicitly defined processes, it is not possible to analyze and efficiently communicate best practice knowledge within the organization [5]. Nevertheless, it seems to be worthwhile to consider individual knowledge work from the perspective of business process optimization. Even if the activities of individual knowledge work are not entirely accessible to planning, they are often embedded in organizational business processes defining, e.g., some constraints on the activities, deadlines, communication partners, or just provide some documents that need to be processed by the knowledge worker [29]. Individual knowledge work often contains some sub-activities that actually provide a fixed structure and thus can be explicitly planned, e.g., to obtain approval for some activities in a large construction project [29]. These activities of individual knowledge work need to be synchronized with the organizational knowledge work. However, today's models for business processes and weakly structured workflows do not allow for representing such an integration of the activities of individual knowledge work and organizational knowledge work.

Solution to this problem is the semantic integration of individual knowledge work and organizational knowledge work based on the pattern-based core ontology *strukt*¹. The *strukt* ontology allows to model weakly structured workflows of individual knowledge work in a formal and precise way. It allows to decompose the tasks of the individual knowledge work into sub-tasks, which again can be structured along a specific order of execution and dependencies between the tasks. The tasks can be semantically connected with any kinds of documents, information, and tools of a particular domain. In addition, the *strukt* ontology provides for modeling structured workflows of organizational knowledge work and combing the weakly structured workflows with the structured ones. The ontology is used in the *strukt* application that allows knowledge workers to collaboratively create, modify, and execute the dynamic tasks of individual knowledge work and to synchronize them with organizational business processes. Goal of the *strukt* application is to capture specific procedural knowledge in the organization in order to effectively and efficiently communicate it with others. In addition, the captured individual knowledge work is also used for oneself to recall the knowledge in cases where it is not often needed.

The remainder of this paper is organized as follows: The need for integrating individual and organizational knowledge work is motivated by a scenario of an architecture firm in Section 2. An introduction into organizational knowledge work and individual knowledge work and a discussion of their characteristics is provided in Section 3. Based on the scenario and the discussion on knowledge work, the requirements to the *strukt* ontology are derived in Section 4. In Section 5, existing models and languages for business process modeling and weakly structured workflows are compared to the requirements. The pattern-based design of the core ontology *strukt* is described in Section 6. An example application of the ontology design patterns defined in *strukt* is provided in Section 7. The use of the core ontology in the *strukt* application for collaboratively executing

¹ *strukt* comes from the German word *Struktur* and means structure in English.

tasks of individual knowledge work and synchronizing it with business processes is presented in Section 8 at the architecture use case, before we conclude the paper.

2 Scenario

The scenario is based on a real architecture firm conducting projects for apartment construction, industrial construction, and commercial construction. We describe both typical activities for organizational knowledge work in terms of traditional business processes and activities of individual knowledge work. The work in the architecture firm is highly knowledge oriented as the acquisition and application of knowledge plays a crucial role in planing and conducting construction projects. New planning techniques, changed regulations and framework directives, new tools, continuously changing project partners, high complexity of the tasks, and the large amount of information to be processed requires (re-)evaluation of the course of action for each new construction project. Nevertheless, one finds some organizational business processes that are central to the business operations of the architecture firm and that are repeated with each project. Figure 1 depicts in Business Process Modeling Notation (BPMN) [25] an excerpt of typical steps in the process of planing an apartment construction. Subsequently to the activity **Initiate construction project** (a) are the activities **Prepare building application** (b) and **File building application** (c). The activities are strictly separated from each other and are executed in a determined, sequential order. The resource (d) defines the input and output documents of an activity. In the case of the activity **Prepare building application** these documents are, e.g., the building application form and all required attachments. The activity **Prepare building application** is associated with the role **Construction draftsman** (e), whereas the other activities are conducted by roles like **Construction manager**, **Structural engineer**, or **Planner**. Branches are used to represent parallel activities (f) and conditions (g). Besides the processes within the company also the communication with external project partners is explicitly captured (h).

The organizational knowledge work is already well described on the level of business processes. However, the core area of the architecture firm's activities is insufficiently captured. For example, activities such as **Prepare building application** or **Draw construction** consist of a large number of sub-activities and are usually collaboratively executed by multiple persons. These activities of individual knowledge work are characterized by high complexity and variability when executing the tasks. As an example, we consider the business process **Prepare building application** of Figure 1 in more detail: For preparing a building application one has to fill a corresponding application form. This form requires some attachments such as ground plan, vertical plan, site plan, specifications of the building, and others that are used to prepare the administrative permit for the construction project. Depending on the type of building construction, however, different attachments are needed. In addition, the construction projects may have specific characteristics that need to be considered such as providing a ter-

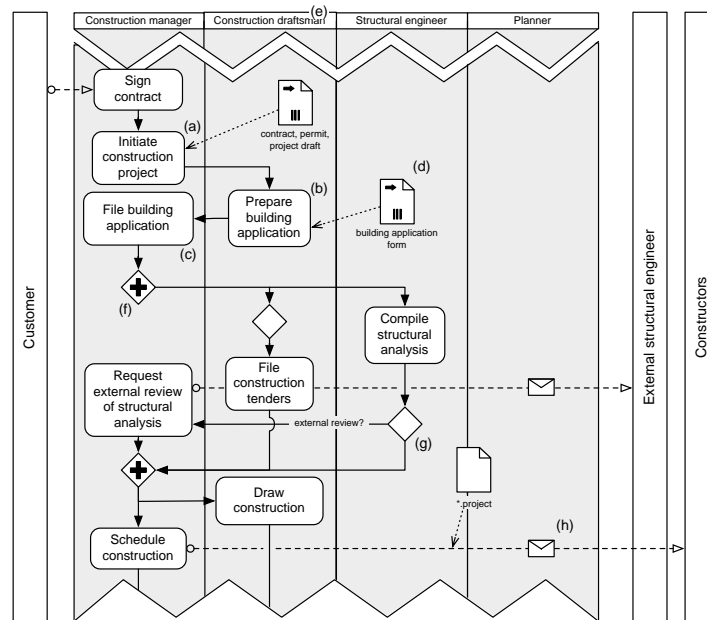


Fig. 1: Example Business Process of the Architecture Firm

restrial heat, timber construction, accessibility, and others. Finally, the location where the building is constructed and customer requirements influence the steps to be conducted for preparing a building application. For example, in some cases a complete structural engineering calculation has to be conducted at application time whereas this is not required in other cases.

3 Knowledge Work

One of the most frequent definitions of knowledge [2] is its distinction from data and information [27]. Whereas data are the raw signals like digits and characters, information enriches this data with some meaning, i.e., some metadata. Knowledge adds a purpose to information in order to achieve a specific goal within the organization. Knowledge is to be continuously revised within the organization. Thus, knowledge work is considered the acquisition and application of new knowledge.

3.1 Organizational Knowledge Work

Organizational knowledge work considers knowledge from the perspective of the company or organization. A central role plays here business processes modeling. It aims to make implicit procedural knowledge of the organization explicit

by defining processes and thus amenable to formal analysis and optimization (cf. [23]). A business process describes a set of organizational activities that uses one or more resources as input and produces a value for the customers of the organization [17]. Business processes consider the organizational activities in a top-down view and aim at improving the use of resources within the organization by an explicit analysis and planning of the organizational activities.

Business process modeling differentiates between the *definition phase* and *execution phase* [32]. In the definition phase, existing business processes are captured and orchestrated into a (semi-)formal business process model. In the execution phase, the previously created process model is implemented. To this end, a *workflow* is defined as supporting the operative execution of a business process in parts or total [32]. Workflows can be distinguished into *workflow models* and *workflow instances* [29]. The workflow model describes a generic definition of a workflow under consideration of activities, alternatives, and roles participating in the processes. Workflow instances are instantiations of a workflow model under consideration of current process parameters. Adapting the workflow instance after its instantiation from the model is not foreseen [29]. If at any time during the execution of a workflow instance the workflow turns out to be inflexible or inappropriate, one needs to adapt the workflow model. This implies that all running workflow instances need to be adapted accordingly. A more flexible behavior of the workflow instances and less dependencies with the workflow model is desired.

3.2 Individual Knowledge Work

Individual knowledge work describes knowledge work from the perspective of the individuals, i.e., the knowledge workers. It is characterized by a high complexity, a low formal structuring, and high degree of variability in its execution [29]. In the following, we discuss the characteristics of individual knowledge work [9, 21, 29]:

Complexity and Variability: Individual knowledge work typically comprises activities that are difficult to plan due to their high complexity and variability. A huge amount of input information and often changing communication partners require frequent revisions of one's procedures in conducting the required activities. Many activities are also conducted rarely or the first time. In order to integrate the varying tasks into a busy and fragmented work schedule [18], the knowledge worker constantly needs to evaluate the different, concurrent activities and coordinate them in an integrated work flow.

Self-organization: The high complexity and variability of individual knowledge work makes it impossible to define a-priori an exact procedure and the resources needed for a task. Thus, it is typically the knowledge worker's responsibility to plan and manage the efforts for conducting a task in terms of time and resources.

Interdisciplinary: In contrast to the traditional skilled engineering worker, knowledge workers are often not specialized and typically cannot specialize to one particular field of business activity. In lieu, knowledge workers have to be

experts in many fields of activity. They must be able to quickly acquire new skills when required and play different roles within the organization.

Communication-orientation: Knowledge work often requires an active communication with other participants. For example, there is a high need for communication with colleagues and the management. However, there is also an increased communication need with external participants such as customers, suppliers, and the administration office.

4 Requirements to *strukt* Ontology

We have derived the requirements to the *strukt* ontology from the scenario in Section 2 and the discussion of knowledge work in Section 3. We briefly discuss each requirement and provide a number *REQ- $\langle number \rangle$* for reference.

Weakly Structured Workflows (*REQ-1*): Individual knowledge work is characterized by a high complexity and variability (see Section 3.2). Resources and activities for conducting tasks are often not known a priori (see Section 2). A support for representing weakly structured workflows is needed that can be adapted during execution time without violating the consistency of other running processes.

Support for Structured Workflows (*REQ-2*): Despite the high flexibility of individual knowledge work, there are also some organizational requirements and framework directives that need to be strictly followed (see scenario in Section 2). Thus, support is needed to represent structured workflows in the sense of traditional business process management (see Section 3.1).

Integrating Weakly Structured and Structured Workflows (*REQ-3*): Within an organization there is typically a need to represent both weakly structured workflows and structured workflows (see Sections 2 and 3)(see Section 2). Today's models and systems, however, lack in formally integrating weakly structured workflows and structured workflows and thus cannot benefit from this integration. In order to leverage the strength of both weakly structured and structured workflows, an appropriate model must be able to formally integrate and synchronize them into a common workflow.

Workflow Models and Instances (*REQ-4*): Distinguishing workflow models and workflow instances is a common feature of traditional business process models (see Section 3.1). In individual knowledge work, however, such a distinction often not conducted as the individual knowledge work is high in complexity and variability (see Section 3.2). However, also from the execution of weakly structured workflows one can learn some generic procedural knowledge. Thus, also for weakly structured workflows the distinction between instance and model should be made.

Descriptive Workflow Information (*REQ-5*): Structured workflows and weakly structured workflows are characterized by the resources involved. A core ontology for integrating individual and organizational knowledge work should therefore support describing the necessary information for the workflow execution, like resources used, processed, or created (which is a central aspect in particular

for individual knowledge work [23]), the tools applied, the status of the workflow execution, as well as scheduling information (cf. Section 2).

5 Comparing Models for Organizational and Individual Knowledge Work

We analyze and evaluate existing models for structured workflows and weakly structured workflows with respect to the requirements introduced in Section 4.

5.1 Traditional Business Process Models

Models for describing traditional business processes such as the Business Process Modelling Notation (BPMN) [25], extended Event-driven Process Chain (eEPC) [26], and Action Port Model (APM) [7] have a rigorously determined process execution flow and separate the business process modeling from its execution (see Section 3.1). Thus, they are less suitable with respect to the requirements stated in this work. Only APM suggests an adaptation of the business process model at runtime. However, due to its lack of formality and low flexibility of the concepts it defines, APM is not suitable to serve as basis for our work.

From the field of knowledge management, we find models like the Knowledge Business Process Improvement Framework (KBPI) [8] and Knowledge Modeling and Description Language (KMDL) [15]. Besides describing traditional business processes, these models provide approaches for planning of knowledge activities in the organizations. However, these models focus on capturing and planing of organizational business processes for the purpose of management. Support for modeling individual knowledge work is not provided by these models.

The Business Process Execution Language (BPEL) [3] allows for the technical interface specification of (automatable) web services. Planing and modeling individual knowledge work in form of weakly structured workflows, is not part of BPEL. The extensions BPEL4People [22] and HumanTask [1] allow in principle for describing activities of individual knowledge work and model them as *black-boxes* within the organizational business processes. However, a specification of the black-box such as partitioning a task into sub-activities is conducted outside of the system. As such, the BPEL4People and HumanTask extensions are de-facto the same as the traditional business process modeling and less appropriate for representing individual knowledge work.

5.2 Semantic Models

The traditional business process models like BPMN and EPC are available as semantic models in form of the sBPMN [19] and sEPC [19] ontologies. However, they still lack support for representing weakly structured workflows and thus are less applicable to our problem. Other models like the Toronto Virtual Enterprise Ontology (TOVE) [11], Enterprise Ontologie (EO) [30], Knowledge Management Ontology (KMO) [20], and Core Organizational Knowledge Entities

Ontology (COKE) [16] share the structural characteristics of traditional business process models such as a strictly determined business process flow and an explicit separation of modeling and execution of the processes (see Section 3.1). Because of this, they are less suitable to model weakly structured workflows. Also the degree of formalization of the ontologies differs. Instead of providing a formalization using description logics, often only a taxonomy of terms in a specific domain is used.

The DOLCE+DnS Plan Ontology (DDPO) [13] provides a rich axiomatization and formal precision. It obtains its high level of formal precision from the foundational ontology DOLCE+DnS Ultralight (DUL) [6]. In addition, it is in principle possible to represent both traditional workflows as well as weakly structured workflows using the DDPO. However, the DDPO is very generic and abstract and thus does not fulfill all requirements. Nevertheless, due to its high formality and using the foundational ontology DUL as basis, the DDPO is well suited for extensions and to serve as basis for our work.

5.3 Models for Weakly Structured Workflows

Ad-hoc and weakly structured models like the Process Meta-Model (PMM) [5] and the Task-Concept-Ontology (TCO) [28] do not require a strictly determined process flow like the traditional business process models. As such, they are at a first glance suitable to represent individual knowledge work. However, the lack of formal precision and missing integration with traditional business processes hinder their reuse.

5.4 Summary

In conclusion, one can say that none of the existing models fulfill all requirements stated to *strukt*. Traditional, organizational business process models capture the processes from the organization's point of view. However, they miss representing the weakly structured workflows of the individual knowledge work. On contrary, weakly structured workflows are in principle enabled to represent the activities of individual knowledge work. However, they lack the formal precision required and do not allow for an integration with traditional business process models. The DDPO model differs from the other models insofar as it in principle allows for modeling both organizational business processes and activities of individual knowledge work. In addition, it enables integration with other systems due to its formal nature. Thus, it is used as basis in our work and will be adapted and extended towards the requirements stated in Section 4.

6 Pattern-based Core Ontology *strukt*

An overview of our ontological support for representing and integrating individual knowledge work and organizational knowledge work is shown in Figure 2. The pattern-based foundational ontology DOLCE+DnS Ultralight (DUL) [6]

serves as basis for the core ontology strukt. Foundational ontologies like DUL provide a highly axiomatized representation of the very basic and general concepts and relations that make up the world [24]. As such, foundational ontologies are applicable to a wide variety of different fields.

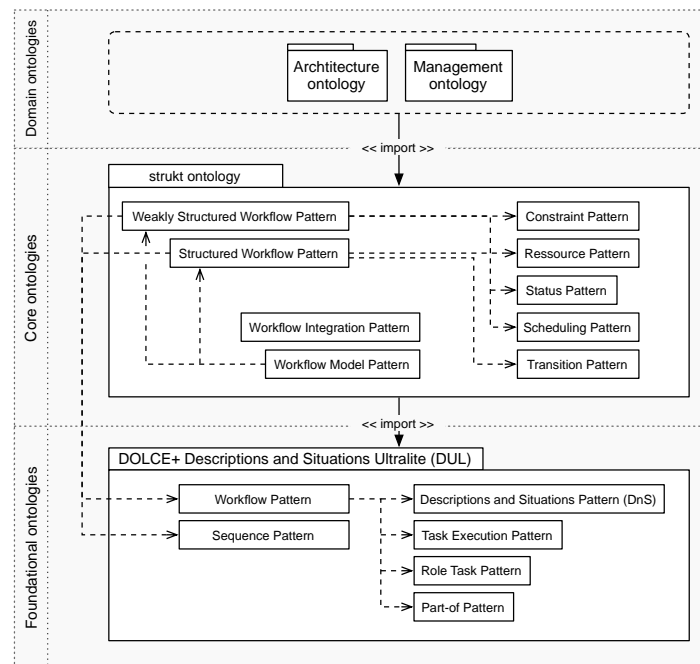


Fig. 2: Overview of the strukt Ontology

Foundational ontologies like DUL follow a pattern-oriented design. Ontology design patterns provide a generic modeling solution for a recurring modeling problem [14]. They allow to arrange the functionality of an ontology in smaller, modular, and reusable units. Thus, ontology design patterns are similar to design patterns in software engineering [12]. In the following, we briefly introduce the patterns of DUL that are of particular interest in this work:² The Descriptions and Situations Pattern provides a formal specification of context [24]. The Description concept formalizes the description of a context by using roles, parameters, and other concepts. The Situation represents an observable excerpt of the real world that satisfies the Description. By using the Descriptions and Situations Pattern, different views onto the same entities can be formally described. The Workflow Pattern is a specialization of the DDPO (see Section 5). It provides generic concepts for planning activities and extends the Descriptions and Situations Pattern. Central entity is the Workflow concept that specializes the

² For a detailed description we refer to <http://ontologydesignpatterns.org/>

Description concept and formalizes the planning of processes. The `WorkflowExecution` concept represents the concrete execution of a workflow instance and is a specialization of the `Situation` concept. The Task Execution Pattern formalizes the processing of tasks in activities. The Role Task Pattern enables association of roles to tasks. The Part-of Pattern represents the (de-)composition of entities into wholes and parts [31]. The Sequence Pattern describes the order of entities through the relations `precedes`, `follows`, `directlyPrecedes`, and `directlyFollows`.

Core ontologies like *strukt* provide a rich axiomatization and pattern-oriented design, which is inherited from the foundational ontology used as basis. A core ontology refines a foundational ontology towards a particular field by adding detailed concepts and relations [24]. As depicted in Figure 2, the core ontology *strukt* reuses and specializes different ontology design patterns that DUL offers. Central patterns the core ontology *strukt* provides are the Weakly Structured Workflow Pattern (*REQ-1*), the Structured Workflow Pattern in combination with the Transition Pattern (*REQ-2*), the Workflow Integration Pattern to integrate weakly structured workflows and structured workflows (*REQ-3*), and the Workflow Model Pattern for differentiating workflow models and workflow instances (*REQ-4*). Weakly structured workflows and structured workflows can be further described by applying *strukt*'s Condition Pattern, Resource Pattern, Status Pattern, and Scheduling Pattern (*REQ-5*). Finally, the core ontology *strukt* is extended by domain ontologies such as an architectural ontology or financial administration ontology. In the following, we describe the patterns of the *strukt* ontology.

6.1 Weakly Structured Workflow Pattern

The Weakly Structured Workflow Pattern depicted in Figure 3 refines the generic Workflow Pattern of DUL. The concept `WeaklyStructuredWorkflow` specializes the `Workflow` concept of DUL's Workflow Pattern. Using the `defines` relation, different `Roles` and `Tasks` are defined. `Roles` abstract from the characteristics, skills, or procedures relevant for the execution of a specific task and allow to differentiate `Agents` and `Objects` participating in activities (see Role Task Pattern of DUL). The `classifies` relation allocates the `Role` an `Object` plays within the context of a specific workflow. The concept `Agent` is a specialization of the `Object` concept and describes the entity acting such as a person. `Objects` and `Agents` are defined as participants of an `Action` by using the `hasParticipant` relation. `Tasks` are used to sequence activities [13]. They structure a workflow into different sub-tasks relevant for the workflow execution and can be hierarchically ordered (see Task Execution Pattern of DUL). `Tasks` are associated to `Actions` using the relation `isExecutedIn`. `Action` is a specialization of DUL's `Event` and describes the actual processing of a task. `Tasks` can be ordered using the `precedes` relation. The order of tasks may be underspecified, i.e., the actual sequence of processing may only be determined on a short-term basis and day-to-day requirements when executing the workflow. Thus, a strict order of processing the tasks is not enforced and the order may even change during execution time.

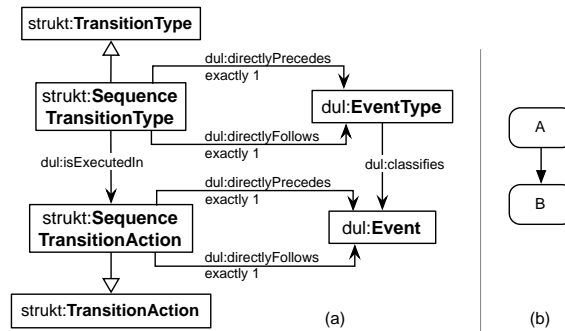


Fig. 5: Sequence Transition Pattern

der of execution of two EventTypes. Thus, the SequenceTransitionAction connects exactly two concrete business processes represented as Events.

6.3 Condition Pattern, Resource Pattern, Status Pattern, and Scheduling Pattern

The workflows specified using the Weakly Structured Workflow Pattern and Structured Workflow Pattern can be further described with information about the conditions, resources, status, and scheduling of activities. Information about conditions for executing an activity are added by combining the Weakly Structured Workflow Pattern or Structured Workflow Pattern with the Condition Pattern. The Condition Pattern allows to define some preconditions and post-conditions such as that a document needs to be signed. Using the Resource Pattern, one can define if an activity produces a resource (create), uses a resource (without exactly knowing if the resource is modified or not), views a resource (without modifying it, i.e., read), edits a resource (update), consumes a resource (delete), or locks a resource. The status of activities and processes can be set to active, inactive, or finished using the Status Pattern. The pattern can be extended to domain specific requirements such as initiated, suspended, and failed. Activities may have to be executed until a specific time and/or place. This can be represented using the Scheduling Pattern. The pattern diagrams are omitted for reasons of brevity.

6.4 Workflow Integration Pattern

The integration of individual knowledge work and organizational knowledge work is conducted using the Workflow Integration Pattern specialized from DUL's Workflow Pattern and is depicted in Figure 6. The alignment of the concepts defined in the Weakly Structured Workflow Pattern and the Structured Workflow Pattern is supported by using the Workflow Pattern of DUL as common modeling basis. As described in Section 6.1 and Section 6.2, it is possible to associate

a Goal to each Task using the *relatesTo* relation. The Goal concept is connected to the workflow via the *hasComponent* relation. Using the Goal concept, a formal mapping of weakly structured workflows and structured workflows can be conducted. It bases on the assumption that if some individual knowledge work is carried out in the context of an organizational business process or vice versa, they share a common Goal. Finally, the association between concrete activities carried out in the individual knowledge work and organizational knowledge work is established through the *relatesTo* relation that connects the Goals with Tasks in the Weakly Structured Workflow Pattern and the Structured Workflow Pattern.

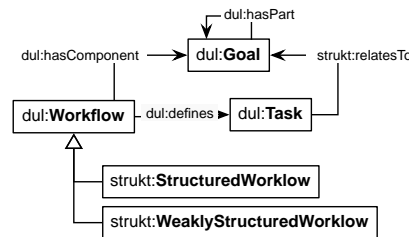


Fig. 6: Workflow Integration Pattern

6.5 Workflow Model Pattern

The Workflow Model Pattern allows to explicitly distinguish workflow models and workflow instances for both, weakly structured workflows and structured workflows. To create a workflow model, the Workflow Model Pattern is able to represent on a *generic*, i.e., conceptual level, the flow of Tasks, their dependencies, and the resources required. In contrast to the traditional business process modeling (see Section 3.1), however, the workflow instances created from a workflow model do not need to strictly cohere with their model. This is in particular important for weakly structured workflows that can be adapted to the requirements of a concrete execution situation.

Figure 7 depicts the Workflow Model Pattern. It consists of two parts, one for the *WeaklyStructuredWorkflowModel* and one for the *StructuredWorkflowModel*. In the case of the *StructuredWorkflowModel*, subclasses of *Role*, *Task*, and *TransitionType* are defined as valid components of the workflow model definition. For weakly structured workflow models, only *Roles* and *Tasks* can be defined.

7 Example Application of the *strukt* Core Ontology

The application of the *strukt* core ontology is shown at the example of an apartment construction by the architecture firm introduced in Section 2. Figure 8 (bottom part) depicts the application of the Weakly Structured Workflow Pattern

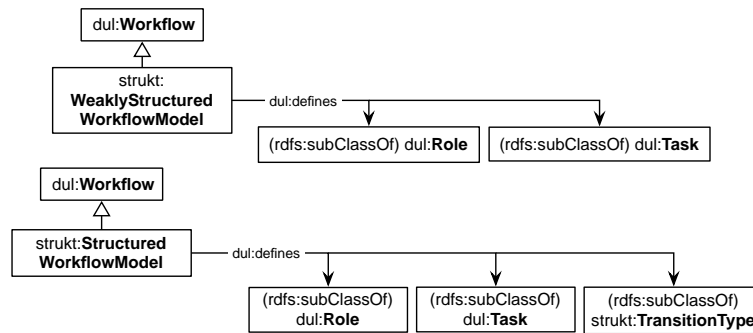


Fig. 7: Workflow Model Pattern

wsw-prepare-building-application-1 for preparing a building application. It defines the Tasks *t-compute-statics-1* and *t-create-groundplan-1*. The relation *isExecutedIn* classifies the individuals *a-compute-statics-1* and *a-create-groundplan-1* as Actions execution of the tasks. The *isTaskOf* relation specifies that the task *t-compute-statics-1* has to be conducted by an agent playing the role of a *StructuralEngineer* *r-structural-engineer-1*, in the example the *NaturalPerson* *tmueller-1*. The *NaturalPerson* *tmueller-1* is specified as participant of the Action *a-compute-statics-1*. The participant of the Action *a-create-groundplan-1* is not specified.

The weakly structured workflow belongs to the organizational business process depicted in Figure 8 (top part) using the Structured Workflow Pattern. It models an excerpt of the business process shown in Figure 1 of the scenario in Section 2. The *StructuredWorkflow* *sw-residential-object-1* defines the Tasks *t-prepare-building-application-1* and *t-submit-application-1*, the *SequenceTransitionType* *tt-sequence-1*, and the role *r-draftsman-1*. The tasks *t-prepare-building-application-1* and *t-submit-application-1* are connected in a sequence using the relations *directlyPrecedes* and *directlyFollows* of *tt-sequence-1*. The Role *r-draftsman-1* is connected using the *isTaskOf* relation with the Task *t-prepare-building-application-1*. In the context of this workflow, the *NaturalPerson* *tmueller-1* acts as *r-draftsman-1*. The Actions *a-prepare-building-application-1*, *a-sequence-1*, and *a-submit-application-1* constitute the execution of the Tasks and *SequenceTransitionType*, respectively. The integration of the weakly structured workflow *wsw-prepare-building-application-1* and structured workflow *sw-construction-project-1* is conducted by defining *g-prepare-building-application-1* as Goal of the *t-prepare-building-application-1* task. The Goal *g-prepare-building-application-1* is then connected with the Weakly Structured Workflow *wsw-prepare-building-application-1* using the *hasComponent* relation. As described above, the *wsw-prepare-building-application-1* captures the individual activities, concrete sub-tasks, and roles involved in actually writing the building application. In contrary to traditional business process modeling, all activities in conducting the individual knowledge work are formally captured using the *strukt* core ontology's patterns.

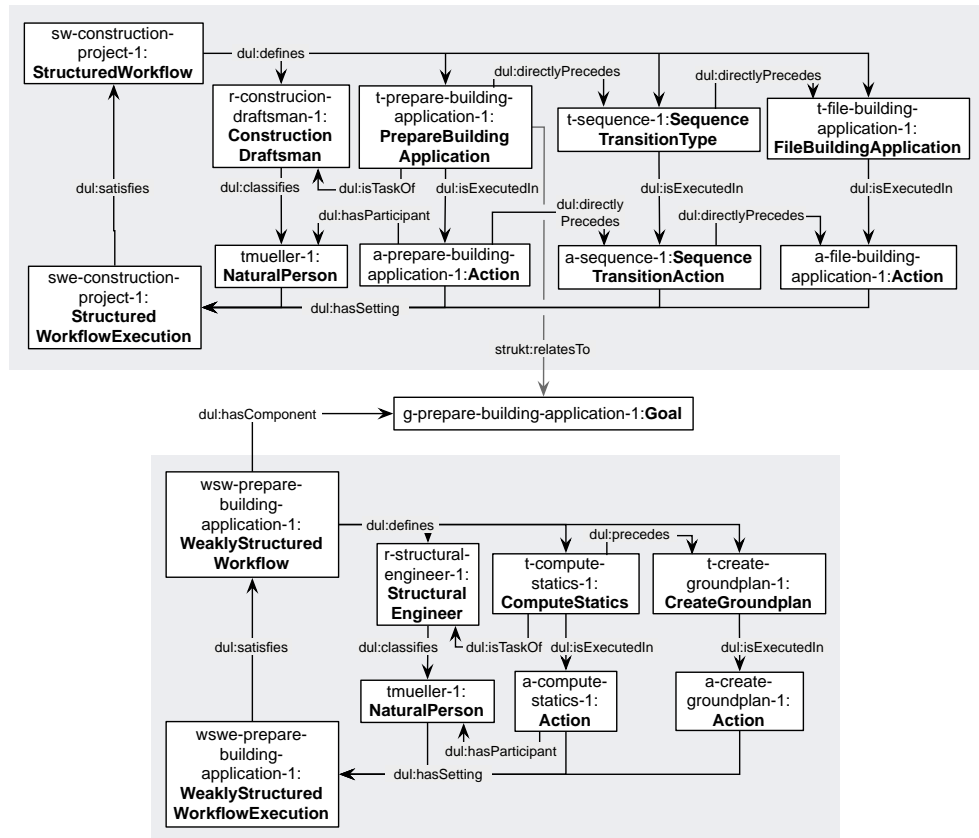


Fig. 8: Example integration of weakly structured and structured workflow

An instance of a workflow such as the example of the weakly structured workflow `wsw-prepare-building-application-1` in Figure 8 (bottom part) can be abstracted to a workflow model using the Workflow Model Pattern. As shown in Figure 9, the abstraction from a workflow instance to a model is basically the upper part of the Descriptions and Situations Pattern of DUL. In our example, the `WeaklyStructuredWorkflowModel wswm-prepare-building-application-1` consists of the domain-specific concepts of the role `StructuralEngineer`, the two tasks `ComputeStatics` and `CreateGroundplan`, and the relations.

When creating a model from a workflow instance, conflicts must be detected and potential solutions offered. As an example, we consider adding another `isTaskOf` relation that connects the task `ApproveStatics` with the role `Architect`. When adding this relation, two cases must be considered: If the `isTaskOf` relation does yet not exist, it may be added or the user creating the workflow model may decide to reject adding this relation. In the case there is already a `isTaskOf` relation defined in the workflow model, e.g., to the `StructuralEngineer`, the following options are possible: The existing relation is kept and the new relation

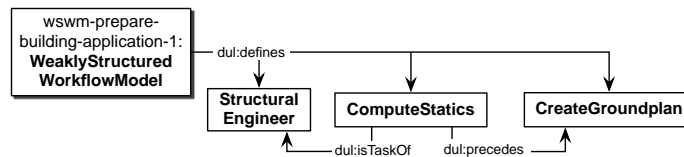


Fig. 9: Application of the Workflow Model Pattern

dismissed (a), the existing relation is deleted and replaced by the new one (b), or both relations are removed (c). In addition, the two relations can be combined using the logical operators AND (d) or OR (e).

8 Prototype Application

8.1 Individual Knowledge Worker Application

The prototype application implementing the strukt core ontology supports the individual knowledge workers in carrying out their tasks. It consists of a *task space* depicted in Figure 10 for managing the weakly structured workflows with their tasks and sub-tasks. The task space allows to receive details of a task, create new tasks, modify tasks, save a workflow instance as workflow model, instantiating a workflow model, and deleting tasks and workflows, respectively. The *work space* contains an overview of the tools and resources provided to the individual knowledge worker for processing a task (no screenshot provided).

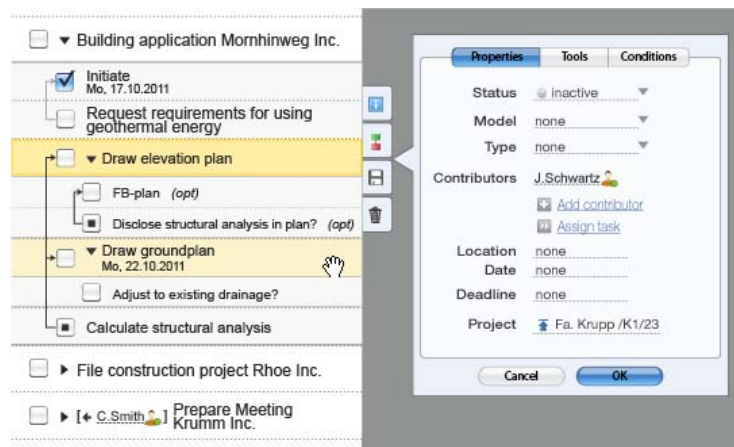


Fig. 10: Task Space of the strukt Prototype

The left hand side of the screenshot depicted in Figure 10 shows example weakly structured workflows from the architecture scenario presented in Sec-

tion 2. The tasks and subtasks of a weakly structured workflow can be shown by clicking on the small triangle symbol next to the task as shown in the figure at the example of the task **Building application Mornhinweg Inc.** Important details of a task are shown on the right hand side of the screenshot such as deadlines, appointments, and others. Tasks can be marked as finished by clicking on the checkbox on the left to the task name. When there is a lock symbol in the checkbox (indicated as small box), the task cannot be accomplished due to unfulfilled dependencies (indicated by the arrows). For example, the task **Calculate structural analysis** cannot be processed as the tasks **Draw elevation plan** and **Draw ground plan** are not completed. Optional tasks are indicated with the keyword (*opt*). The order of the tasks in a weakly structured workflow can be changed by simple drag and drop interaction.

The right hand side of the screenshot in Figure 10 provides details of a task such as its status and the responsible agent. Additional agents can be added as well as the responsibility of tasks can be forwarded. Thus, the strukt prototype enables a collaborative execution of a weakly structured workflow by multiple knowledge workers. Further details can be investigated using the tab **Tools** showing the tools used to process a task and the tab **Conditions** showing detailed information about the conditions associated with the task, e.g., when a specific role needs to sign a specific document.

8.2 Creating a Workflow Model from a Workflow Instance

In order to abstract a workflow model from a workflow instance, the strukt application provides the *workflow transformation menu* as depicted in Figure 11. The workflow task menu allows to define the components of the workflow model. To this end, all components of the workflow instance to be transformed are depicted in a table. Each row of the table represents a task of the weakly structured workflow. Subtasks are indicated by indentions. Concrete details of the task instances such as scheduling date and time and the responsible agent are discarded prior to the transformation. The columns **Task**, **Conditions**, **Optional**, **Role conditions**, **Documents**, and **Tools** show the details of the tasks relevant for creating a workflow model. Tasks can be removed from the workflow at this point of the transformation process. In addition, new tasks can be added and the order of the tasks can be changed by drag and drop interaction. Clicking on the **OK** button creates the workflow model.

8.3 Integration with Structured Workflows

We have implemented a simple workflow management system in our strukt prototype. The workflow management component provides a test environment for synchronizing the activities in the weakly structured workflows and some predefined business processes of the architecture firm. Thus, it serves as proof of concept that weakly structured workflows can be connected with structured ones, i.e., that individual knowledge work can be integrated with organizational knowledge work using the core ontology strukt. We have implemented different

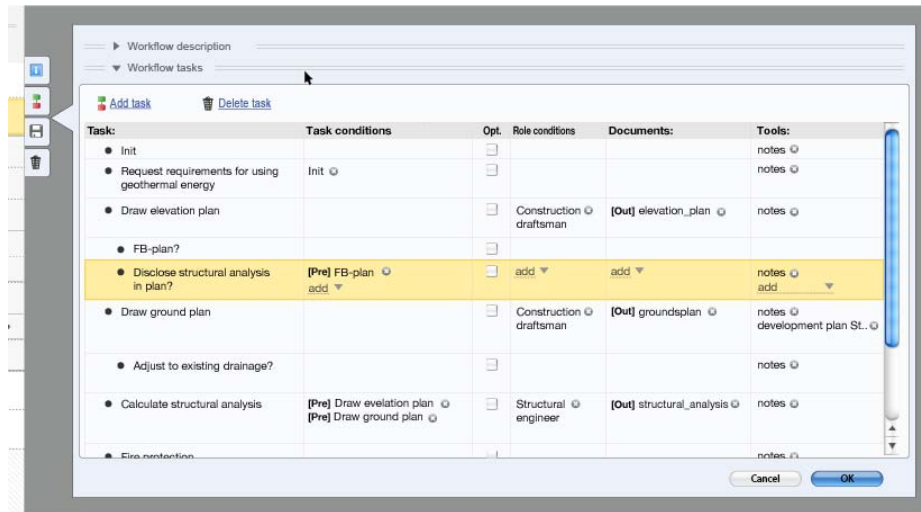


Fig. 11: Transformation Menu for Creating a Workflow Model from an Instance

test cases: For example, a task is assigned from a business process to an agent of a weakly structured workflow. In another example, the status of a task in a weakly structured workflow is observed by a business process. The business process of the structured workflow is only finished when the task in the weakly structured workflow is marked as accomplished.

9 Conclusions

Business process management is an established means for improving efficiency in organizational knowledge work by capturing implicit procedural knowledge in the organization and making it explicit through the specification of processes. On the contrary, we have seen that individual knowledge work is characterized by a high degree of complexity and variability, which is difficult to capture in an information system. This hinders adoption and communication of best practices in individual knowledge work within the organization and its synchronization with organizational knowledge work.

As solution to this problem, we have presented in this paper an approach for formally integrating individual knowledge work and organizational knowledge work by means of the pattern-based core ontology *strukt*. The core ontology *strukt* defines several ontology design patterns for capturing weakly structured workflows of individual knowledge work and structured workflows in the context of organizational knowledge work. A formal alignment and synchronization of the activities in individual knowledge work and organizational knowledge work is established by using the Workflow Pattern of the foundational ontology DOLCE+DnS Ultralight. Concrete instances of weakly structured workflows can

be transformed into generic workflow models, enabling reuse of procedural knowledge. On basis of the strukt ontology, we have developed a prototypical software system for the collaborative planning and execution of weakly structured workflows and applied it to the use case of an architecture firm. The strukt prototype connects with a simple workflow management system to synchronize the flexible, individual knowledge work with the strict execution of business processes.

References

1. A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M. Zeller. Web services human task (WS-HumanTask), version 1.0, June 2007. http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf.
2. M. Alavi and D. Leidner. Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. *MIS quarterly*, 25(1):107–136, 2001.
3. A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, Y. Goland, A. Guzar, and N. Kartha. Web services business process execution language, 2007. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
4. J. Becker, B. Weiss, and A. Winkelmann. Developing a Business Process Modeling Language for the Banking Sector - A Design Science Approach. In *Americas Conference on Information Systems*, pages 1–12, 2009.
5. J. Bolinger, G. Horvath, J. Ramanathan, and R. Ramnath. Collaborative workflow assistant for organizational effectiveness. In *Applied Computing*. ACM, 2009.
6. S. Borgo and C. Masolo. *Handbook on Ontologies*, chapter Foundational choices in DOLCE. Springer, 2nd edition, 2009.
7. S. Carlsen. Action port model: A mixed paradigm conceptual workflow modeling language. In *Cooperative Information Systems*, pages 300–309. IEEE, 1998.
8. P. Dalmaris, E. Tsui, B. Hall, and B. Smith. A Framework for the Improvement of Knowledge-Intensive Business Processes. *Business Process Management Journal*, 13(2):279–305, 2007.
9. G. Davis. Anytime/Anyplace Computing and the Future of Knowledge Work. *Communications ACM*, 45(12):67–73, 2002.
10. P. Drucker. Knowledge-Worker Productivity: The Biggest Challenge. *California Management Review*, 41(2):79–94, 1999.
11. M. Fox, M. Barbuceanu, M. Grüninger, and J. Lin. An organization ontology for enterprise modelling. *Simulating Organizations: Computational Models of Institutions and Groups*, pages 131–152, 1998.
12. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Programmer’s Choice. Addison-Wesley, July 2004.
13. A. Gangemi, S. Borgo, C. Catenacci, and J. Lehmann. Task Taxonomies for Knowledge Content. *METOKIS Deliverable, D7:20–42*, 2004. http://www.loa-cnr.it/Papers/D07_v21a.pdf.
14. A. Gangemi and V. Presutti. *Handbook on Ontologies*, chapter Ontology Design Patterns. Springer, 2009.

15. N. Gronau, U. Palmer, K. Schulte, and T. Winkler. Modellierung von wissensintensiven Geschäftsprozessen mit der Beschreibungssprache K-Modeler. In *Professionelles Wissensmanagement - Erfahrungen und Visionen*, pages 315–322, 2003.
16. A. Gualtieri and M. Ruffolo. An Ontology-based Framework for Representing Organizational Knowledge, 2005.
17. M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Paperbacks, 2003.
18. W. Hart-Davidson, C. Spinuzzi, and M. Zachry. Capturing & Visualizing Knowledge Work: Results & Implications of a Pilot Study of Proposal Writing Activity. In *Design of Communication*, pages 113–119. ACM, 2007.
19. M. Hepp, R. Belecheanu, J. Domingue, A. FilipowskaG, M. Kaczmarek, T. Kaczmarek, J. Nitzsche, B. Norton, C. Pedrinaci, D. Roman, et al. Business Process Modelling Ontology and Mapping to WSMO. Technical report, SUPER Project IST-026850, 2006.
20. C. Holsapple and K. Joshi. A Formal Knowledge Management Ontology: Conduct, Activities, Resources, and Influences: Research Articles. *Journal of the American Society for Information Science and Technology*, 55(7):612, 2004.
21. A. Kidd. The marks are on the knowledge worker. In *Human Factors in Computing Systems*, pages 186–191. ACM, 1994.
22. M. Kloppmann, D. König, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic. WS-BPEL Extension for People—BPEL4People, July 2005. <http://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/cfab6fdd-0501-0010-bc82-f5c2414080ed>.
23. I. Nonaka. The Knowledge-Creating Company. *Harvard Business Review*, 69(6):96–104, 1991.
24. D. Oberle. *Semantic Management of Middleware*. Springer, 2006.
25. OMG. Business process model and notation (BPMN), version 2.0 beta 2, 2010. <http://www.omg.org/cgi-bin/doc?dtdc/10-06-04.pdf>.
26. A. Scheer. *Aris-Business Process Frameworks*. Springer, 1998.
27. G. Schreiber. *Knowledge engineering and management: the CommonKADS methodology*. MIT Press, 2000.
28. S. Schwarz. Task-Konzepte: Struktur und Semantik für Workflows. In *Professionelles Wissensmanagement; Luzern, Switzerland*. GI e.V., 2003.
29. S. Schwarz, A. Abecker, H. Maus, and M. Sintek. Anforderungen an die Workflow-Unterstützung für Wissensintensive Geschäftsprozesse. In *Professionelles Wissensmanagement; Baden-Baden, Germany*, 2001.
30. M. Uschold, M. King, S. Moralee, and Y. Zorgios. The Enterprise Ontology. *The Knowledge Engineering Review*, 13(1):31–89, 1998.
31. A. Varzi. Parts, Wholes, and Part-Whole Relations: The Prospects of Mereotopology. *Data & Knowledge Engineering*, 20(3):259–286, 1996.
32. WFMC. Terminology & Glossary. Technical Report WFMC-TC-1011, Version 3.0, Workflow Management Coalition, 1999. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.

Bisher erschienen

Arbeitsberichte aus dem Fachbereich Informatik

(<http://www.uni-koblenz-landau.de/koblenz/fb4/publications/Reports/arbeitsberichte>)

Daniel Eißing, Ansgar Scherp, Steffen Staab, Formal Integration of Individual Knowledge Work and Organizational Knowledge Work with the Core Ontology *strukt*, Arbeitsberichte aus dem Fachbereich Informatik 10/2011

Bernhard Reinert, Martin Schumann, Stefan Müller, Combined Non-Linear Pose Estimation from Points and Lines, Arbeitsberichte aus dem Fachbereich Informatik 9/2011

Tina Walber, Ansgar Scherp, Steffen Staab, Towards the Understanding of Image Semantics by Gaze-based Tag-to-Region Assignments, Arbeitsberichte aus dem Fachbereich Informatik 8/2011

Alexander Kleinen, Ansgar Scherp, Steffen Staab, Mobile Facets – Faceted Search and Exploration of Open Social Media Data on a Touchscreen Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 7/2011

Anna Lantsberg, Klaus G. Troitzsch, Towards A Methodology of Developing Models of E-Service Quality Assessment in Healthcare, Arbeitsberichte aus dem Fachbereich Informatik 6/2011

Ansgar Scherp, Carsten Saathoff, Thomas Franz, Steffen Staab, Designing Core Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 5/2011

Oleg V. Kryuchin, Alexander A. Arzamastsev, Klaus G. Troitzsch, The prediction of currency exchange rates using artificial neural networks, Arbeitsberichte aus dem Fachbereich Informatik 4/2011

Klaus G. Troitzsch, Anna Lantsberg, Requirements for Health Care Related Websites in Russia: Results from an Analysis of American, British and German Examples, Arbeitsberichte aus dem Fachbereich Informatik 3/2011

Klaus G. Troitzsch, Oleg Kryuchin, Alexander Arzamastsev, A universal simulator based on artificial neural networks for computer clusters, Arbeitsberichte aus dem Fachbereich Informatik 2/2011

Klaus G. Troitzsch, Natalia Zenkova, Alexander Arzamastsev, Development of a technology of designing intelligent information systems for the estimation of social objects, Arbeitsberichte aus dem Fachbereich Informatik 1/2011

Kurt Lautenbach, A Petri Net Approach for Propagating Probabilities and Mass Functions, Arbeitsberichte aus dem Fachbereich Informatik 13/2010

Claudia Schon, Linkless Normal Form for ALC Concepts, Arbeitsberichte aus dem Fachbereich Informatik 12/2010

Alexander Hug, Informatik hautnah erleben, Arbeitsberichte aus dem Fachbereich Informatik 11/2010

Marc Santos, Harald F.O. von Kortzfleisch, Shared Annotation Model – Ein Datenmodell für kollaborative Annotationen, Arbeitsberichte aus dem Fachbereich Informatik 10/2010

Gerd Gröner, Steffen Staab, Categorization and Recognition of Ontology Refactoring Pattern, Arbeitsberichte aus dem Fachbereich Informatik 9/2010

Daniel Eißing, Ansgar Scherp, Carsten Saathoff, Integration of Existing Multimedia Metadata Formats and Metadata Standards in the M3O, Arbeitsberichte aus dem Fachbereich Informatik 8/2010

Stefan Scheglmann, Ansgar Scherp, Steffen Staab, Model-driven Generation of APIs for OWL-based Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 7/2010

Daniel Schmeiß, Ansgar Scherp, Steffen Staab, Integrated Mobile Visualization and Interaction of Events and POIs, Arbeitsberichte aus dem Fachbereich Informatik 6/2010

Rüdiger Grimm, Daniel Pähler, E-Mail-Forensik – IP-Adressen und ihre Zuordnung zu Internet-Teilnehmern und ihren Standorten, Arbeitsberichte aus dem Fachbereich Informatik 5/2010

Christoph Ringelstein, Steffen Staab, PAPEL: Syntax and Semantics for Provenance-Aware Policy Definition, Arbeitsberichte aus dem Fachbereich Informatik 4/2010

Nadine Lindermann, Sylvia Valcárcel, Harald F.O. von Kortzfleisch, Ein Stufenmodell für kollaborative offene Innovationsprozesse in Netzwerken kleiner und mittlerer Unternehmen mit Web 2.0, Arbeitsberichte aus dem Fachbereich Informatik 3/2010

Maria Wimmer, Dagmar Lück-Schneider, Uwe Brinkhoff, Erich Schweighofer, Siegfried Kaiser, Andreas Wieber, Fachtagung Verwaltungsinformatik FTVI Fachtagung Rechtsinformatik FTRI 2010, Arbeitsberichte aus dem Fachbereich Informatik 2/2010

Max Braun, Ansgar Scherp, Steffen Staab, Collaborative Creation of Semantic Points of Interest as Linked Data on the Mobile Phone, Arbeitsberichte aus dem Fachbereich Informatik 1/2010

Marc Santos, Einsatz von „Shared In-situ Problem Solving“ Annotationen in kollaborativen Lern- und Arbeitsszenarien, Arbeitsberichte aus dem Fachbereich Informatik 20/2009

Carsten Saathoff, Ansgar Scherp, Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology, Arbeitsberichte aus dem Fachbereich Informatik 19/2009

Christoph Kahle, Mario Schaarschmidt, Harald F.O. von Kortzfleisch, Open Innovation: Kundenintegration am Beispiel von IPTV, Arbeitsberichte aus dem Fachbereich Informatik 18/2009

Dietrich Paulus, Lutz Priese, Peter Decker, Frank Schmitt, Pose-Tracking Forschungsbericht, Arbeitsberichte aus dem Fachbereich Informatik 17/2009

Andreas Fuhr, Tassilo Horn, Andreas Winter, Model-Driven Software Migration Extending SOMA, Arbeitsberichte aus dem Fachbereich Informatik 16/2009

Eckhard Großmann, Sascha Strauß, Tassilo Horn, Volker Riediger, Abbildung von grUML nach XSD soamig, Arbeitsberichte aus dem Fachbereich Informatik 15/2009

Kerstin Falkowski, Jürgen Ebert, The STOR Component System Interim Report, Arbeitsberichte aus dem Fachbereich Informatik 14/2009

Sebastian Magnus, Markus Maron, An Empirical Study to Evaluate the Location of Advertisement Panels by Using a Mobile Marketing Tool, Arbeitsberichte aus dem Fachbereich Informatik 13/2009

Sebastian Magnus, Markus Maron, Konzept einer Public Key Infrastruktur in iCity, Arbeitsberichte aus dem Fachbereich Informatik 12/2009

Sebastian Magnus, Markus Maron, A Public Key Infrastructure in Ambient Information and Transaction Systems, Arbeitsberichte aus dem Fachbereich Informatik 11/2009

Ammar Mohammed, Ulrich Furbach, Multi-agent systems: Modeling and Virification using Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 10/2009

Andreas Sprotte, Performance Measurement auf der Basis von Kennzahlen aus betrieblichen Anwendungssystemen: Entwurf eines kennzahlengestützten Informationssystems für einen Logistikdienstleister, Arbeitsberichte aus dem Fachbereich Informatik 9/2009

Gwendolin Garbe, Tobias Hausen, Process Commodities: Entwicklung eines Reifegradmodells als Basis für Outsourcingentscheidungen, Arbeitsberichte aus dem Fachbereich Informatik 8/2009

Petra Schubert et. al., Open-Source-Software für das Enterprise Resource Planning, Arbeitsberichte aus dem Fachbereich Informatik 7/2009

Ammar Mohammed, Frieder Stolzenburg, Using Constraint Logic Programming for Modeling and Verifying Hierarchical Hybrid Automata, Arbeitsberichte aus dem Fachbereich Informatik 6/2009

Tobias Kippert, Anastasia Meletiadou, Rüdiger Grimm, Entwurf eines Common Criteria-Schutzprofils für Router zur Abwehr von Online-Überwachung, Arbeitsberichte aus dem Fachbereich Informatik 5/2009

Hannes Schwarz, Jürgen Ebert, Andreas Winter, Graph-based Traceability – A Comprehensive Approach. Arbeitsberichte aus dem Fachbereich Informatik 4/2009

Anastasia Meletiadou, Simone Müller, Rüdiger Grimm, Anforderungsanalyse für Risk-Management-Informationssysteme (RMIS), Arbeitsberichte aus dem Fachbereich Informatik 3/2009

Ansgar Scherp, Thomas Franz, Carsten Saathoff, Steffen Staab, A Model of Events based on a Foundational Ontology, Arbeitsberichte aus dem Fachbereich Informatik 2/2009

Frank Bohdanovicz, Harald Dickel, Christoph Steigner, Avoidance of Routing Loops, Arbeitsberichte aus dem Fachbereich Informatik 1/2009

Stefan Ameling, Stephan Wirth, Dietrich Paulus, Methods for Polyp Detection in Colonoscopy Videos: A Review, Arbeitsberichte aus dem Fachbereich Informatik 14/2008

Tassilo Horn, Jürgen Ebert, Ein Referenzschema für die Sprachen der IEC 61131-3, Arbeitsberichte aus dem Fachbereich Informatik 13/2008

Thomas Franz, Ansgar Scherp, Steffen Staab, Does a Semantic Web Facilitate Your Daily Tasks?, Arbeitsberichte aus dem Fachbereich Informatik 12/2008

Norbert Frick, Künftige Anfordeungen an ERP-Systeme: Deutsche Anbieter im Fokus, Arbeitsberichte aus dem Fachbereich Informatik 11/2008

Jürgen Ebert, Rüdiger Grimm, Alexander Hug, Lehramtsbezogene Bachelor- und Masterstudiengänge im Fach Informatik an der Universität Koblenz-Landau, Campus Koblenz, Arbeitsberichte aus dem Fachbereich Informatik 10/2008

Mario Schaarschmidt, Harald von Kortzfleisch, Social Networking Platforms as Creativity Fostering Systems: Research Model and Exploratory Study, Arbeitsberichte aus dem Fachbereich Informatik 9/2008

Bernhard Schueler, Sergej Sizov, Steffen Staab, Querying for Meta Knowledge, Arbeitsberichte aus dem Fachbereich Informatik 8/2008

Stefan Stein, Entwicklung einer Architektur für komplexe kontextbezogene Dienste im mobilen Umfeld, Arbeitsberichte aus dem Fachbereich Informatik 7/2008

Matthias Bohnen, Lina Brühl, Sebastian Bzdak, RoboCup 2008 Mixed Reality League Team Description, Arbeitsberichte aus dem Fachbereich Informatik 6/2008

Bernhard Beckert, Reiner Hähnle, Tests and Proofs: Papers Presented at the Second International Conference, TAP 2008, Prato, Italy, April 2008, Arbeitsberichte aus dem Fachbereich Informatik 5/2008

Klaas Dellschaft, Steffen Staab, Unterstützung und Dokumentation kollaborativer Entwurfs- und Entscheidungsprozesse, Arbeitsberichte aus dem Fachbereich Informatik 4/2008

Rüdiger Grimm: IT-Sicherheitsmodelle, Arbeitsberichte aus dem Fachbereich Informatik 3/2008

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik 2/2008

Markus Maron, Kevin Read, Michael Schulze: CAMPUS NEWS – Artificial Intelligence Methods Combined for an Intelligent Information Network, Arbeitsberichte aus dem Fachbereich Informatik 1/2008

Lutz Priese, Frank Schmitt, Patrick Sturm, Haojun Wang: BMBF-Verbundprojekt 3D-RETISEG Abschlussbericht des Labors Bilderkennen der Universität Koblenz-Landau, Arbeitsberichte aus dem Fachbereich Informatik 26/2007

Stephan Philippi, Alexander Pinl: Proceedings 14. Workshop 20.-21. September 2007 Algorithmen und Werkzeuge für Petrinetze, Arbeitsberichte aus dem Fachbereich Informatik 25/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS – an Intelligent Bluetooth-based Mobile Information Network, Arbeitsberichte aus dem Fachbereich Informatik 24/2007

Ulrich Furbach, Markus Maron, Kevin Read: CAMPUS NEWS - an Information Network for Pervasive Universities, Arbeitsberichte aus dem Fachbereich Informatik 23/2007

Lutz Priese: Finite Automata on Unranked and Unordered DAGs Extended Version, Arbeitsberichte aus dem Fachbereich Informatik 22/2007

Mario Schaarschmidt, Harald F.O. von Kortzfleisch: Modularität als alternative Technologie- und Innovationsstrategie, Arbeitsberichte aus dem Fachbereich Informatik 21/2007

Kurt Lautenbach, Alexander Pinl: Probability Propagation Nets, Arbeitsberichte aus dem Fachbereich Informatik 20/2007

Rüdiger Grimm, Farid Mehr, Anastasia Meletiadou, Daniel Pähler, Ilka Uerz: SOA-Security, Arbeitsberichte aus dem Fachbereich Informatik 19/2007

Christoph Wernhard: Tableaux Between Proving, Projection and Compilation, Arbeitsberichte aus dem Fachbereich Informatik 18/2007

Ulrich Furbach, Claudia Obermaier: Knowledge Compilation for Description Logics, Arbeitsberichte aus dem Fachbereich Informatik 17/2007

Fernando Silva Parreiras, Steffen Staab, Andreas Winter: TwoUse: Integrating UML Models and OWL Ontologies, Arbeitsberichte aus dem Fachbereich Informatik 16/2007

Rüdiger Grimm, Anastasia Meletiadou: Rollenbasierte Zugriffskontrolle (RBAC) im Gesundheitswesen, Arbeitsberichte aus dem Fachbereich Informatik 15/2007

Ulrich Furbach, Jan Murray, Falk Schmidsberger, Frieder Stolzenburg: Hybrid Multiagent Systems with Timed Synchronization-Specification and Model Checking, Arbeitsberichte aus dem Fachbereich Informatik 14/2007

Björn Pelzer, Christoph Wernhard: System Description: "E-KRHyper", Arbeitsberichte aus dem Fachbereich Informatik, 13/2007

Ulrich Furbach, Peter Baumgartner, Björn Pelzer: Hyper Tableaux with Equality, Arbeitsberichte aus dem Fachbereich Informatik, 12/2007

Ulrich Furbach, Markus Maron, Kevin Read: Location based Information systems, Arbeitsberichte aus dem Fachbereich Informatik, 11/2007

Philipp Schaer, Marco Thum: State-of-the-Art: Interaktion in erweiterten Realitäten, Arbeitsberichte aus dem Fachbereich Informatik, 10/2007

Ulrich Furbach, Claudia Obermaier: Applications of Automated Reasoning, Arbeitsberichte aus dem Fachbereich Informatik, 9/2007

Jürgen Ebert, Kerstin Falkowski: A First Proposal for an Overall Structure of an Enhanced Reality Framework, Arbeitsberichte aus dem Fachbereich Informatik, 8/2007

Lutz Priebe, Frank Schmitt, Paul Lemke: Automatische See-Through Kalibrierung, Arbeitsberichte aus dem Fachbereich Informatik, 7/2007

Rüdiger Grimm, Robert Krimmer, Nils Meißner, Kai Reinhard, Melanie Volkamer, Marcel Weinand, Jörg Helbach: Security Requirements for Non-political Internet Voting, Arbeitsberichte aus dem Fachbereich Informatik, 6/2007

Daniel Bildhauer, Volker Riediger, Hannes Schwarz, Sascha Strauß, „grUML – Eine UML-basierte Modellierungssprache für T-Graphen“, Arbeitsberichte aus dem Fachbereich Informatik, 5/2007

Richard Arndt, Steffen Staab, Raphaël Troncy, Lynda Hardman: Adding Formal Semantics to MPEG-7: Designing a Well Founded Multimedia Ontology for the Web, Arbeitsberichte aus dem Fachbereich Informatik, 4/2007

Simon Schenk, Steffen Staab: Networked RDF Graphs, Arbeitsberichte aus dem Fachbereich Informatik, 3/2007

Rüdiger Grimm, Helge Hundacker, Anastasia Meletiadou: Anwendungsbeispiele für Kryptographie, Arbeitsberichte aus dem Fachbereich Informatik, 2/2007

Anastasia Meletiadou, J. Felix Hampe: Begriffsbestimmung und erwartete Trends im IT-Risk-Management, Arbeitsberichte aus dem Fachbereich Informatik, 1/2007

„Gelbe Reihe“

(<http://www.uni-koblenz.de/fb4/publikationen/gelbereihe>)

Lutz Priebe: Some Examples of Semi-rational and Non-semi-rational DAG Languages. Extended Version, Fachberichte Informatik 3-2006

Kurt Lautenbach, Stephan Philippi, and Alexander Pinl: Bayesian Networks and Petri Nets, Fachberichte Informatik 2-2006

Rainer Gimnich and Andreas Winter: Workshop Software-Reengineering und Services, Fachberichte Informatik 1-2006

Kurt Lautenbach and Alexander Pinl: Probability Propagation in Petri Nets, Fachberichte Informatik 16-2005

Rainer Gimnich, Uwe Kaiser, and Andreas Winter: 2. Workshop "Reengineering Prozesse" – Software Migration, Fachberichte Informatik 15-2005

Jan Murray, Frieder Stolzenburg, and Toshiaki Arai: Hybrid State Machines with Timed Synchronization for Multi-Robot System Specification, Fachberichte Informatik 14-2005

Reinhold Letz: FTP 2005 – Fifth International Workshop on First-Order Theorem Proving, Fachberichte Informatik 13-2005

Bernhard Beckert: TABLEAUX 2005 – Position Papers and Tutorial Descriptions, Fachberichte Informatik 12-2005

Dietrich Paulus and Detlev Droege: Mixed-reality as a challenge to image understanding and artificial intelligence, Fachberichte Informatik 11-2005

Jürgen Sauer: 19. Workshop Planen, Scheduling und Konfigurieren / Entwerfen, Fachberichte Informatik 10-2005

Pascal Hitzler, Carsten Lutz, and Gerd Stumme: Foundational Aspects of Ontologies, Fachberichte Informatik 9-2005

Joachim Baumeister and Dietmar Seipel: Knowledge Engineering and Software Engineering, Fachberichte Informatik 8-2005

Benno Stein and Sven Meier zu Eißel: Proceedings of the Second International Workshop on Text-Based Information Retrieval, Fachberichte Informatik 7-2005

Andreas Winter and Jürgen Ebert: Metamodel-driven Service Interoperability, Fachberichte Informatik 6-2005

Joschka Boedecker, Norbert Michael Mayer, Masaki Ogino, Rodrigo da Silva Guerra, Masaaki Kikuchi, and Minoru Asada: Getting closer: How Simulation and Humanoid League can benefit from each other, Fachberichte Informatik 5-2005

Torsten Gipp and Jürgen Ebert: Web Engineering does profit from a Functional Approach, Fachberichte Informatik 4-2005

Oliver Obst, Anita Maas, and Joschka Boedecker: HTN Planning for Flexible Coordination Of Multiagent Team Behavior, Fachberichte Informatik 3-2005

Andreas von Hessling, Thomas Kleemann, and Alex Sinner: Semantic User Profiles and their Applications in a Mobile Environment, Fachberichte Informatik 2-2005

Heni Ben Amor and Achim Rettinger: Intelligent Exploration for Genetic Algorithms – Using Self-Organizing Maps in Evolutionary Computation, Fachberichte Informatik 1-2005