Chapter XIV

A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

Thomas Hädrich
Martin-Luther-University Halle-Wittenberg, Germany

Torsten Priebe
University of Regensburg, Germany

ABSTRACT

Knowledge work can be characterized by a high degree of variety and exceptions, strong communication needs, weakly structured processes, networks and communities, and as requiring a high level of skill and expertise as well as a number of specific practices. Process-oriented knowledge management suggests to focus on enhancing efficiency of knowledge work in the context of business processes. Portals are an enabling technology for knowledge management by providing users with a consolidated, personalized interface that allows accessing various types of structured and unstructured information. However, the design of portals still needs concepts and frameworks to guide their alignment with the context of persons consigned with knowledge-intensive tasks. In this context the concept of knowledge stance is a promising starting point. This paper discusses how knowledge stances can be applied and detailed to model knowledge work and support to support it with semantic context-based portals. We present the results from implementing a portal prototype that deploys Semantic Web technologies to integrate various information sources and applications on a semantic level and discuss extensions to this portal for the support of knowledge stances.
INTRODUCTION

Knowledge work can be characterized by a high degree of variety and exceptions, strong communication needs, weakly structured processes, networks and communities, and as requiring a high level of skill and expertise as well as a number of specific practices (Schulze, 2003). Process-oriented knowledge management (KM) suggests to focus on enhancing efficiency of knowledge work in the context of business processes and by this way to link KM efforts the value chains of organizations (Edwards & Kidd, 2003; Maier & Remus, 2003). Various types of information and communication technologies (ICT) are deployed to support knowledge work, ideally forming an enterprise-wide knowledge infrastructure (EKI) (Maier, Hädrich, & Peinl, 2005). Portals are an important part of the EKI since they provide users with a consolidated, personalized interface that allows accessing various types of structured and unstructured information as well as applications simultaneously.

Models are a foundation to design supporting ICT in general and portals in particular. However, process-oriented KM lacks ways to model knowledge work in the context of business processes, especially the knowledge-oriented actions connected to the tasks accomplished in business processes. Here, the concept of knowledge stance can be seen as a promising starting point (Hädrich & Maier, 2004). This paper has the goals to (a) discuss how knowledge stances can be applied and detailed to model knowledge work and to support it with semantic portals, (b) present results from implementing a portal prototype that deploys Semantic Web technologies to integrate various information sources on a semantic level (Priebe, 2004; Priebe & Pernul, 2003), and (c) discuss extensions to this portal to support knowledge stances.

The remainder of this paper is organized as follows: The concept of knowledge stance is outlined in section 2 together with its conceptual foundations. Section 3 provides a framework for context information and relates knowledge stances to it. Section 4 presents the INWISS knowledge portal prototype and how it applies Semantic Web technologies to provide a context-based portlet integration. Section 5 proposes extensions to the portal based on knowledge stances and discusses how these can be implemented. Section 6 concludes the paper and gives an outlook on future research.

MODELING KNOWLEDGE WORK

Modeling approaches applied in KM can be classified according to the concepts that they primarily emphasize into four categories: (1) person (e.g., communication relationships and structural organization), (2) process (e.g., business processes and tasks), (3) topic (e.g., knowledge structure defined by an ontology) and (4) tool (e.g., software architecture and interaction of components) (Maier, 2004). From the view of KM, particularly the interconnections between concepts in these categories are of interest, e.g., “Markus Schmidt” (person) is experienced in “project management” (topic). When choosing a process-oriented KM approach, the relationships between the categories process and topic are of primary interest, i.e. the link between functions and tasks accomplished in business processes and the knowledge applied and created in this context. This section describes two perspectives on knowledge work that correspond to these two categories: a process-oriented and an activity-oriented perspective. The concept of knowledge stance is one possible way to connect these perspectives.

Process Modeling vs. Activity Modeling

Examples for traditional process modeling approaches are ADONIS (Junginger, Kühn, Strobl, & Karagiannis, 2000), ARIS (Scheer, 2001), IEM
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

(Spur, Mertins, & Jochem, 1996), MEMO (Frank, 2002), PROMET (Österle, 1995), SOM (Ferstl & Sinz, 1994), UML-based process modeling (Österle, Weiss, Schröder, Weilkiens, & Lenhard, 2003) and IDEF. Examples for approaches that extend process modeling for KM are ARIS with extensions (Allweyer, 1998), PROMET®I-NET (Bach & Österle, 2000), GPO WM (Heisig, 2002), KMDL (Gronau, 2003), Knowledge MEMO (Schauer, 2004) and PROMOTE (Karagiannis & Woitsch, 2003). The main extensions are the introduction of additional object types like knowledge object, i.e. topics of interest, documented knowledge, individual employee, and skill as well as the introduction of model types like knowledge structure diagram and communication diagram. Even though the added concepts describe a portion of the context of knowledge work, they are not suited to model the often unstructured and creative learning and knowledge practices in knowledge work and particularly their link to business processes.

Activity theory has been proposed to guide the analysis of knowledge work (Blackler, 1995) and the design of information systems (Clases & Wehner, 2002; Kuutti, 1997; Sachs, 1995). The underlying thesis is that knowledge is not an object, a passive unit. Rather, the processes of knowing and doing take place in so-called activity systems (Blackler, 1995) which are the basic unit of analysis (not to be confused with activities in Porter’s value chain and activities in UML). The core idea of activity theory is that human activity is a dialectic relationship between individuals (called agents or subjects) and objects (the purpose of human activity) that is mediated (a) by tools and instruments like cultural signs, language and technologies and (b) by communities of people that are involved within the transformation process of the activity (see Figure 1). The relation between subject and community is determined by implicit or explicit social rules. A division of labor (e.g., role system) defines the relation of the community to the object of the activity system. The outcomes of the activities’ transformation process are intended or unintended results.

Another important feature of activity theory is that activities have a hierarchical structure: (1) The activity is driven by a common motive which reflects a collective need and the reason why the activity exists (Engeström, 1999). (2) It is accomplished by actions directed to goals coupled to the motive of the activity. Actions consist of an orientation and an execution phase: the first comprises the planning for action, the latter its execution by a chain of operations (Kuutti, 1997). Repeated exercise leads to better planning of the action that then can be conducted more successfully. Due to learning and reutilization, the planning phase can become obsolete and actions collapse into operations. (3) Operations are executed under certain conditions. They are clearly structured and easy to automate. These levels are characterized by a dynamic relationship: Elements of higher levels collapse to constructs of lower levels if learning takes place. They unfold to higher levels if changes occur and learning is necessary. Activity modeling comprises identification of activity systems together with their context and history. It emphasizes analysis of the mediating relationships and tensions between their constituting components and other activity systems.

Figure 1. Socially-distributed activity system (Blackler, 1995; Kuutti, 1997)
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

Compared to process modeling, activity theory contributes the concept of mediation, consideration of individual and group motives, the notion of communities and ways to conceptualize learning by routinization. The concepts provided by activity theory are well suited to analyze the creative, unstructured and learning-oriented practices of knowledge work. However, activities primarily aim at the joint creation of knowledge (exploration of knowledge). They lack integration with the value chain and it is not ensured that they are oriented towards creating customer value (exploitation of knowledge). Therefore, concepts of process and of activity modeling have to be combined in order to get a more comprehensive picture of knowledge work in a business context.

The Concept of Knowledge Stance

As we have seen, activity modeling differentiates between the levels motives, goals and conditions. Approaches for process modeling distinguish between three corresponding levels of granularity: (1) Value chains arrange value-adding activities (Porter, 1985), (2) business processes connect functions and (3) workflows orchestrate tasks. Figure 2 contrasts both perspectives. An important difference is that in the process-oriented perspective, a change from a higher to a lower level corresponds to refinement whereas in the activity-oriented perspective this is associated with routinization. We propose to connect both perspectives on the level of goals by the concept of knowledge stance.

A knowledge stance is a class of recurring situations in knowledge work defined by occasion, context, and mode resulting in knowledge-oriented actions (Hädrich & Maier, 2004). It describes a situation in which a person can, should, or must switch from a business-oriented function to a knowledge-oriented action. In a process-oriented perspective, an employee accomplishes functions on the level of goals that belong to a value chain on the level of motives by fulfilling a sequence of tasks on the level of conditions. Simultane-

---

**Figure 2. Concept of knowledge stance**

![Figure 2](image-url)
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

Table 1. Components of the knowledge stance

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occasion</td>
<td>is a type of opportunity to learn and to generate knowledge related to the (core) competencies of the organization within the function of a business process.</td>
</tr>
<tr>
<td>Context</td>
<td>describes the current work situation, i.e. process context, activity context, and person-related information.</td>
</tr>
<tr>
<td>Mode</td>
<td>classifies knowledge-oriented actions into expressing, monitoring, translating and networking.</td>
</tr>
<tr>
<td>Action</td>
<td>refers to an unstructured knowledge-oriented action and is specified by occasion, context and mode.</td>
</tr>
</tbody>
</table>

...ously, she can be involved in an activity framing knowledge-oriented actions and corresponding operations. It can (a) be focused on the business process or (b) pursue a motive not related to the business process (e.g., an effort to build competencies) and thus may make a direct or indirect contribution to the process goal.

A business process offers several occasions to learn and to generate knowledge related to the core competencies of the organization. Occasions trigger knowledge stances and are associated with the functions of the business process by offering the opportunity or the need for knowledge-related actions. A knowledge stance is not limited to the generation of knowledge, but may also include the translation and application of knowledge created outside the knowledge stance.

The context includes all dimensions suitable to describe the current situation of the person. It comprises the process context consisting of elements such as involved organizational units, roles, and resources as well as elements of the activity context, e.g., contacts to persons that are member of a community dealing with creation of related knowledge. We will discuss types and dimensions of context information in detail in section 3.1.

The mode classifies what actions can be performed and refers to four informing practices (Schulze, 2000; Schulze, 2003): (a) expressing is the practice of self-reflexive conversion of individual knowledge and subjective insights into informational objects that are independent of the person, (b) monitoring describes continuous non-focused scanning of the environment and gathering of useful “just in case”-information, (c) translating involves creation of information by ferrying it across different contexts until a coherent meaning emerges, and (d) networking is the practice of building and maintaining relationships with people inside and outside the organization.

During the process of modeling, context, mode and occasion are means to specify a set of available, allowed or required knowledge-oriented actions. Examples for actions are evaluate source, indicate level of certitude, compare sources, link content, relate to prior information, add metadata, notify and alert, ask questions, and offer interaction (Eppler, 2003). In contrast to the clearly defined sequences of functions in the process-oriented perspective, there is no predetermined flow of actions. They are accomplished by executing operations suited to serve the goals of the action. Table 1 summarizes the components of a knowledge stance.

An example for a knowledge stance is “learning about product features”, which is related by the occasion “product introduced by vendor” to the procurement process of a company that sells home electronics. It is linked to an activity that aims at gathering knowledge about relevant products and their features and thus strongly related to the core competency “offering the right product at superior prices to the customer”.

Table 1. Components of the knowledge stance
Shop assistants involved in the sales process and consigned with the tasks to consult customers are part of this activity. The knowledge stance thus links multiple processes, activities and people in support of learning and generation of new knowledge. Examples for knowledge-related actions triggered by the knowledge stance are “contact a shop assistant” and “look-up information about product features”.

CONTEXT-BASED SUPPORT

Knowledge comprises observations that have been meaningfully organized and embedded in a context through experience, communication, or inference that an actor uses to interpret situations and to accomplish tasks (Maier, 2004). It is tightly coupled to and embedded in its context of creation and application. In a broad definition, context is “any information that can be used to characterize the situation of an entity; an entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” (Dey & Abowd, 1999). Note that our idea of a user context is different from the notion of context in work on context mediation and interchange for data integration (Tan, Madnick, & Tan, 2004) which rather considers a user-independent application context.

To effectively support knowledge work, it is crucial to provide specifically those functions and knowledge artifacts that the user needs in his current context. Main challenges are to identify and model the relevant context dimensions, to automatically recognize the user’s current context, to detect transitions between contexts, to keep context models up-to-date and last but not least to (dynamically) decide which functions and elements are suited best to support users in a certain context. Another challenge is that users may not apply a system because context-based filtering mechanisms are experienced as being too rigid by hiding information that may be relevant (Dey & Abowd, 1999; Klemke, 2000).

In the last years, many approaches and prototypes for context-based support of system users were introduced, particularly in the area of knowledge management and information retrieval, but also in computer supported co-operative work (CSCW), workflow, and more recently in mobile computing. For example, the PreBIS approach (Böhm & Härtwig, 2005) provides a situated information delivery by attuning a so-called contextualized information system to an organization in a pre-build (modeling) phase. Similarly, the KnowMore project (Abecker, Bernardi, Maus, Sintek, & Wenzel, 2000; Maus, 2001) proposes the use of the context information that can be derived from a workflow management system. Information on the current task context is taken from the workflow and organizational model and transferred into a query on a document base. The DECOR project (Abecker et al., 2001) tries to extend the ideas form the KnowMore project, that were originally developed for strongly structured processes, to weakly structured processes. Finally, Henrich & Morgenroth (2002) (see also below) propose a high-level architecture for retrieving documents relevant to the current user context by including plug-ins in client applications that communicate the context to a search engine. In more recent publications the authors have concentrated on the particular use within the software development environment (Henrich & Morgenroth, 2003).

Context Types and Dimensions

In the following, we present a framework suited to structure relevant context dimensions. We distinguish two general categories of context types: (a) The context of a person dealing with a concrete task in a specific situations and (b) the context of knowledge elements (to be) stored in a system (see Figure 3).
The situation of a person can be described by the following types of contexts (Henrich & Morgenroth, 2003):

- The *user context* represents the physical context (location, time) of the user, her personal profile including her interests and skills in selected topics as well as the organizational position (leading e.g., to specific user privileges).
- The *working context* characterizes the current tasks and actions a user needs to accomplish in a process-oriented and in an activity-oriented perspective (see section 2.1) with regard to available material or immaterial tools (e.g., information systems, methods, language), roles (e.g., role of the user in a business process, role in a community), rules (e.g., access rights, organizational and social rules), partners (e.g., communication partner, external contact persons, experts in a specific domain), topics (e.g., information and knowledge relevant for completion of a task) and related functions or knowledge-oriented actions (e.g., functions of another business process that applies the results of the current function).
- The user’s *interaction context* reflects past and current interactions with the application system, e.g., selected menus or dialogues and protocols about the systems usage.

The context of knowledge elements can be described by an *environmental context* consisting of the six dimensions person (e.g., author, target group, expert), related topics (e.g., knowledge domains), geographical location (e.g., of organizational unit, of creation), type of knowledge (e.g., technical format, conceptual structure), time (e.g., creation time, modified date) and processes/activities where the knowledge element is applied or was created (Maier & Sametinger, 2002). These dimensions render the static part of the context directly related to the content of the knowledge element.

Additionally, the *transitional context* links the knowledge element to the processes of creation and application of knowledge:
The dimension level of commitment refers to the life cycle of knowledge that distinguishes different stages of organizational legitimation. Individual ideas, opinions and suggestions have a low level of legitimation while good or best practices shared between a large number of members of an organization have been repeatedly evaluated and thus have a high level of commitment and legitimation. New and unproven knowledge usually has low levels of legitimation and is strongly linked to exploration of knowledge in an activity-oriented perspective. On the other hand, application of proven knowledge is a predominant target in business processes and thus links knowledge to the process-oriented perspective.

Orientation refers to whether the context reflects the circumstances of creation, e.g., author’s name, creation date and assumptions or purpose of creation, or whether it relates to retrieval and application of knowledge, e.g., by categorizing knowledge, relating it to other knowledge elements or domains, describing access rights, usage restrictions and circumstances as well as feedback from its re-use (Barry & Schamber, 1998; Eppler, 2003).

The level of transferability indicates whether the knowledge element represents generalized knowledge that can be more or less easily be transferred to other contexts (e.g., guidelines applicable in multiple different processes) or reflects a particular case (e.g., experience about a particular product configuration in a special setting).

Applying Knowledge Stances for Context-Based Support

Compared to previous approaches of context-based support, our conceptualization of context is broader as it includes elements related to knowledge creation and thus to learning. This section clarifies how to apply the concept of knowledge stance to guide the design of context-based information systems. Firstly, knowledge stances need to be modeled by accomplishing the following steps:

1. Activities are identified by analyzing the core competencies of the organization and identifying groups and communities concerned with developing knowledge related to them.
2. Selected business processes are detailed and their functions are analyzed with regard to occasions to learn or to generate new knowledge relevant to develop these core competencies. Here, knowledge stances are linked to the process.
3. The context of each knowledge stance is defined based on elements of the working context elements for both, the process-oriented and activity-oriented perspective.
4. Knowledge-oriented actions suited to accomplish the function are defined and linked to the knowledge stance.

Knowledge stances are defined on type-level during build-time. At this stage, only those elements can be assigned that are valid for all instances of related processes and activities. Examples are guidelines, checklists and good or best practices (Goesmann & Herrmann, 2001). During run-time, the context of a knowledge stance can also comprise instance-level information, e.g., documents or functions used for the last execution of the function, contact information about employees that recently answered questions about a related topic or entries about new documented experiences. These elements will be part of the interaction context. Additionally, the user context can be combined with the working and interaction context to further filter the system’s output.

Knowledge stances can be supported at different levels and by different means, e.g., by portals or workspaces that bundle KM functions and filter...
contents for knowledge stances, by user agents that guide through an action, by workflows that routinize parts of actions, by functions that enable communication and collaboration between individuals that is triggered by the knowledge stance.

Ideally, an enterprise knowledge portal provides a platform with advanced knowledge services for publication, discovery, collaboration and learning, which brings together the various heterogeneous data and information sources and applications of the organization (Priebe & Pernul, 2003). Semantic Web technologies aim at structuring, describing, translating, reasoning about and securely accessing metadata and provide promising starting points for developing and implementing systems that support knowledge stances. Ontologies help to organize and link knowledge elements from multiple systems on a semantic level, represent the semantics of the organizational knowledge base and to structure the context of the knowledge elements. We will now turn to a prototypical implementation of a portal that deploys Semantic Web technologies and discuss extensions to it for supporting knowledge stances.

**INWISS—AN INTEGRATIVE ENTERPRISE KNOWLEDGE PORTAL**

Using Web-based technologies, knowledge portals are an emerging approach to provide a single point of access to various information sources and applications. Today’s portal systems allow combining different portal components, so-called portlets, side by side on a single portal webpage (Wege, 2002). However, there is only little interaction between those portlets, which means that the user needs to manually transfer the context. Earlier, we presented an approach for integrative knowledge portals, communicating the user context among portlets using Semantic Web technologies (Priebe & Pernul, 2003). For example, the query context of a reporting portlet, i.e. the information shown within a certain Online Analytical Processing (OLAP) report (Chaudhuri & Dayal, 1997), can be used by a search portlet to automatically provide the user with related intranet articles or documents. The approach is implemented within the INWISS knowledge portal prototype (Priebe, 2004).

The use of Semantic Web technologies within knowledge portals has also been proposed in other works such as OntoViews (Mäkelä, Hyvönen, Saarelä, & Viljanen, 2004), ODESeW (Corcho, Gómez-Pérez, López-Cima, López-García, & Suárez-Figueroa), and SEAL (Stojanovic, Maedche, Staab, Studer, & Sure, 2001). There however, metadata and ontologies are mainly used for content management and searching. Within INWISS we use a semantic representation of the user context to allow portlets to communicate with each other.

**Context-Based Portlet Integration and Retrieval**

Current portal systems provide only limited inter-portlet communication capabilities. If they are offered at all, they require extensive individual programming and are not suitable for portlets that are supposed to be deployed as standard software components. The IBM WebSphere Portal provides a concept called Click-to-Action (C2A) and Cooperative Portlets which add advanced capabilities for managing portlet messaging. The communication paths between portlets no longer have to be explicitly coded but can be bound dynamically, i.e. the communication targets do not need to be known when the portlets are developed. However, the interpretation of messages and back-end integration are not addressed. The SAP Enterprise Portal provides a technology called Drag&Relate. It allows dragging objects from a portlet onto a navigation panel invoking certain navigation actions. Drag&Relate only works for
special Unifier iViews (portlets), which can be used to access (and combine) information from structured data sources such as relational databases or legacy systems. It handles the backend integration by means of a Unification Server. However, it can not be used to integrate third party portlets.

Our generic portlet integration approach within INWISS is based on communicating the user context among portlets, utilizing Semantic Web technologies for the context representation and back-end integration. Usually portlets only provide their portlet content for rendering the user interface (Wege, 2002). In addition, we introduce a context management service, where portlets can publish their current context, i.e. a semantic representation of what the user sees. Other portlets can pick that context up and use it to display related information. Figure 4 shows the overall architecture of our context-based portlet integration.

In order to be able to map the semantics of context elements between portlets, we base our approach on Semantic Web standards and technologies. The main idea is to use the Resource Description Framework (RDF) (W3C, 2004) to represent the context, i.e. portlets should annotate their content with RDF metadata. For example, if a user displays an OLAP report, the context can be represented as the set of elements such as product categories shown on the report (see Figure 5), or a portlet representing a customer relationship management (CRM) system displaying information about a certain customer can point to a customer object to represent its context.

The anonymous RDF description of the context represents the elements shown on the report by identifying them with URIs. Web Ontology Language (OWL) subclassing and concept mapping (W3C, 2004) (e.g., “owl:equivalentClass” and “owl:sameAs”) and an inference engine can be used to map these to business objects from an enterprise ontology (see below). Hence, the portlets can use their own “language” to represent and interpret the context. The advantage of using the Semantic Web standards RDF and OWL over other logic languages is the already large (and emerging) support by standard software tools for storage and reasoning (e.g. Sesame and Jena).

We have identified different context integration scenarios, distinguished by the dimensions shown in the morphological box in Figure 6.

The first dimension is the communication paradigm. Context integration can follow a push or a pull principle. An example for a context pull is a search engine that uses the context of the other

---

Figure 4. Architecture for context integration
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

Figure 5. Sample portlet context

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mstr="http://www.microstrategy.com/terms/">
  <rdf:Description>
  </rdf:Description>
</rdf:RDF>
```

Figure 6. Morphological box of context integration scenarios

<table>
<thead>
<tr>
<th>communication paradigm</th>
<th>context push</th>
<th>context pull</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggering event</td>
<td>explicit</td>
<td>implicit</td>
</tr>
<tr>
<td>transmission method</td>
<td>unicast</td>
<td>multicast</td>
</tr>
<tr>
<td>published context</td>
<td>full</td>
<td>selective</td>
</tr>
</tbody>
</table>

portlets to enhance the precision of the search results. For context pushes, the *triggering event* can be explicit or implicit. Explicit context pushes require the user to explicitly invoke the event, e.g. by clicking a “find related” button. Implicit context pushes are triggered automatically by other events, e.g., by user navigation within a portlet. An example for an implicit context push would be a topic navigation portlet that publishes the selected topic after every browsing event. Context push messages can be transmitted as *unicast* with a single portlet as the destination, as *multicast* with a set of target portlets, or as *broadcast* to all other portlets. Finally, the context published with a context push can be *full* (i.e. cover all information that is shown within the portlet) or *selective*. A selective context push can only be explicit, as it requires the user to select the part of the context that should be published. An example for a selective context push would be a “find related” button next to a customer name in a customer list that triggers a CRM portlet to display related customer information.

Note that not all combinations of all characteristics are possible. The INWISS prototype realizes different context integration scenarios. A major application is to provide implicit searches based on the current user context. A “find related” button is provided in reporting and content portlets that triggers a search portlet to search for related documents by means of an *explicit unicast context push*. In order to be able to perform context-based
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

searches and due to semantics that can be used, we use metadata queries rather than full-text searches by utilizing an enterprise ontology. For example, Figure 7 shows the Dublin Core metadata for two documents: a product experience report (linked to the Freeplay Solar Radio product) and a procurement guidelines document. In addition to ontology elements a simple topic taxonomy is used for the annotation.

A search initiated by the portlet context shown in Figure 5 should also find the above experience report as being related if the described product belongs to one of the subcategories in the context. Firstly, the concepts used by the context provider (in this case a business intelligence system) need to be mapped to the ones used by the search engine. For example, the product category identified by the URI “http://www.microstrategy.com/elements/Subcategory_1” needs be mapped to something like “http://www.inwiss.org/ontology#Audio”.

Secondly, inference rules need to be used to provide that documents annotated with products belonging to “audio” are also annotated with the category. Finally, the property “mstr:element” needs to be considered as semantically identical with “dc:coverage”. Note that this can be achieved by means of OWL or a similar ontology language, combined with an inference engine, requiring no modification of the portlets themselves.

Besides ontological concept mapping, such implicit queries require a fuzzy retrieval approach. Current metadata querying techniques, however, do not support vague queries. Hence, we developed a metadata-based information retrieval approach similar to classical retrieval models like the Vector Space Model (VSM) (Baeza-Yates & Ribeiro-Neto, 1999). It is based on the similarity of RDF descriptions: Both, the query and the resources are represented as RDF descriptions and the ranking of the search results is done

---

Figure 7. Sample document metadata

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
>
    <dc:creator rdf:resource="ldap://cn=Tina Techwriter,ou=Sales,o=MyCompany"/>
    <dc:date>1998-04-05</dc:date>
    <dc:format>application/pdf</dc:format>
    <dc:description>The Freeplay(TM) Solar Radio never needs batteries -- the crank-up radio that runs for an hour on a single crank up. Solar power provides additional play time.</dc:description>
    <dc:subject rdf:resource="http://www.inwiss.org/topics/Sales"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.inwiss.org/documents/ProcurementGuidelines.pdf">
    <dc:title>Procurement Guidelines</dc:title>
    ...
    <dc:subject rdf:resource="http://www.inwiss.org/topics/Procurement"/>
  </rdf:Description>
</rdf:RDF>
```
using a similarity measure (Priebe, Schläger, & Pernul, 2004).

Such a metadata-based search engine will of course only work if the documents are properly annotated. This requires a certain critical mass of metadata-enriched documents. Users will only manually annotate documents, if they see a significant benefit from it. An extension to INWISS approaches the problem of metadata creation by means of text mining and (semi) automated annotation (Priebe, Kiss, & Kolter, 2005).

Prototype

Figure 8 shows a screenshot of the INWISS prototype (Priebe, 2004). At this point we provide four portlets: One is responsible for displaying intranet articles. A second one provides reporting access to a data warehouse. The navigation portlet represents a taxonomy-based topic browser. Finally, a fourth portlet is responsible for metadata-based searches.

As mentioned above, INWISS demonstrates different context integration scenarios. The navigation portlet publishes its topic to the other portlets, triggered implicitly by browse events (context push). The search portlet accepts context messages from the content and the OLAP portlet. In this case the context push is triggered explicitly when the user clicks a “find related” control in the portlet title bar. Finally, when checking to use the portal context in the search portlet, the search engine will query the context of the other portlets and add it to the user query (context pull).

The context management is implemented as an extension to the Apache Jetspeed Portal platform. For the data warehouse access we use the MicroStrategy 7i business intelligence system. The open source Sesame RDF Framework (Broekstra, Kampman, & van Harmelen, 2002)
is used as a repository for resource metadata, a taxonomy, and an ontology.

**Evaluation**

The main contribution of INWISS is the proposal for a context-based portlet integration in enterprise portals. It enables the portal to allow implicit searches for unstructured documents (based on their metadata) using the context of an external application with access to structured database data. As an example we use a business intelligence system that views OLAP reports. The approach can, however, also be generalized to other scenarios.

The results are very promising. A major—although hardly measurable—success is the generic practicability of the context integration approach. The mentioned reporting portlet has been built completely independently, i.e. it contains no specific code that considers the existence of any of the other portlets. This ensures an applicability of the approach also for portlets that are provided as third party software components.

The semantic search engine (Priebe et al., 2004) is only a byproduct of this main proposal. However, its evaluation is more tangible. We have run the search engine against an RDF repository with 46,608 triples in total. The repository contains metadata for 1,322 resources. The ontology consists of 23 classes and 2,421 object instances. We used 10 sample queries for the evaluation; five of them were dynamically created from a user context. Although we still use generated test data and an imaginary company as a scenario, we can state that it achieves a significant gain in recall compared to approaches that do not consider semantic links within ontology-based metadata. In terms of query speed, we achieved acceptable performance on standard PC hardware, which is sufficient for our prototype. Future performance improvements will be possible by directly applying the Vector Space Model (VSM), e.g. utilizing the Jakarta Lucene information retrieval framework\(^1\).

**SUPPORTING KNOWLEDGE STANCES**

User actions can be supported by various functions and services of the ICT infrastructure presented within the portlets of the portal. This comprises services for operative tasks and specifically those services part of a knowledge portal targeting support of knowledge-related actions and learning connected to the current function of the business process. These are (a) publication services to create, store and edit documents and to complement them with metadata, (b) discovery services to navigate the system, retrieve documented knowledge and discover subject matter experts, (c) collaboration services that allow for knowledge exchange over various media and cooperation between users, and (d) learning services that facilitate creation and use of electronic courses or evaluations (Maier, 2004; Maier et al., 2005).

In section 3 we have discussed dimensions to structure the user context as well as the context of knowledge elements. If one and the same ontology is used to represent the context of the user as well as of knowledge elements or if the concepts are properly mapped, the system can proactively search for resources related to the current user context. Section 4 presented INWISS, a portal that employs Semantic Web technologies (in particular RDF and OWL) for this purpose.

INWISS so far concentrates on interaction context elements. In addition, a static user context can easily be defined in RDF and queried together with the interaction context. A major improvement would be to regard the working context within the portal and to transfer this context to services that support (parts of) knowledge-oriented actions. As an extension, which we are currently implementing, we propose to add the idea of
knowledge stances explicitly to the system, i.e. a knowledge stance model is used to represent the working context.

**Representing Knowledge Stances**

Figure 9 shows the schema of the knowledge stance and context model on which we base our extensions to INWISS (which represents a subset of the dimensions discussed in section 3.1). The part that already existed in the prototype before is the knowledge element perspective. Knowledge elements (resources, e.g., documents) are annotated with Dublin Core metadata pointing to a type, taxonomy topics, and ontology objects (e.g., products). Recall the examples from Figure 7 in section 4.1.

In addition, we introduce a person perspective, i.e. persons can also be described by metadata. The main concept that is used for annotation is role, which will also be used to define responsibilities in the knowledge stance model. For example, Figure 10 shows the metadata for the employee “Michael Bates” who has the role “shop assistant” and is involved in the “sales” process. Annotating persons (and roles) with other context elements (topics and ontology objects) may also provide a user context that can be used together with the interaction context for contest pulls as sketched above. In addition, persons may also be considered as elements that are found by a semantic search engine as they may be contacted for collaboration purposes.

We capture the working context by means of a knowledge stance model, represented by the elements explained in section 2.2: process, occasion, activity, action, and task. The link between the process- and activity-oriented perspectives is realized by associating occasions with activities. The knowledge stance elements can be annotated

---

**Figure 9. Schema of the knowledge stance model and context elements**

[Diagram of the knowledge stance model and context elements]

---
with the same concepts as the knowledge elements (depicted by the dotted lines in Figure 9), i.e. an activity can, for example, be related to a certain topic, as well as roles to define responsibilities.

Figure 11 and Figure 12 show an example of a knowledge stance model in RDF. Figure 11 defines two occasions “new product introduced by vendor” and “new product demand by customer”
linked to the procurement and sales process respectively. These occasions link the process to the shared activity “develop knowledge about products”. This activity is defined in Figure 12 by a set of actions and corresponding tasks. On each level, context information can be incorporated into the definition. For example, the processes are related to the corresponding topics of the organization’s taxonomy. The action “lookup experiences with related products” relates to documents of the product experience report type. Additionally, persons of the role “purchaser” or “shop assistant” are related to the action (i.e. they might be worth being contacted). Activity and process-related contexts thus represent parts of the working contexts defining situations that are triggered by the two occasions.

Utilizing the Working Context

Ideally, the portal would automatically recognize occasions by processing information of the current interaction context. It could then notify the user about occasions and present supportive contents and functions. Challenges are whether the portal contains enough information to conclude to the user’s working context and to define rules that allow concluding from interaction context to working context elements. Since users are only confronted with a manageable number of occasions (usually five to ten, depending on their tasks), a straightforward way is to let the user manually choose from a list of occasions.

It is desirable to guide or even automate the operations that execute knowledge-oriented

Figure 12. Activity-oriented part of the knowledge stance model

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:inwiss="http://www.inwiss.org/schema#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <inwiss:actions rdf:parseType="Collection">
      <inwiss:Action rdf:resource="http://www.inwiss.org/actions/LookupRelatedProductExperiences">
        ...<inwiss:tasks rdf:parseType="Collection">
          ...
        </inwiss:tasks>
      </inwiss:Action>
    </inwiss:actions>
  </inwiss:Activity>
  <!-- Activity-related context -->
  <inwiss:role rdf:resource="http://www.inwiss.org/roles/Purchaser"/>
  <inwiss:role rdf:resource="http://www.inwiss.org/roles/ShopAssistant"/>
</rdf:RDF>
```
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

actions. Hence, an expedient extension to the INWISS portal would be a “My Work” portlet as shown in Figure 13. Current process, occasion, and activity are presented in this portlet and may be changed by the user. Depending on his choices, appropriate actions are shown in the list below. If a corresponding workflow is defined, the user can select from the available workflow tasks and is taken to a supporting portlet, activating the desired application function and presenting appropriate content based on the current context.

The workflows are modeled in advance at the time when knowledge stances and corresponding actions are defined. Knowledge elements (e.g., topics, objects, roles) related to the knowledge stance can be linked to them.

Figure 14 shows a possible working context as provided by the “My Work” portlet. The working context comprises knowledge stance elements as well as context elements that can be inferred from the knowledge stance model in Figure 11 and Figure 12. Everything with the subject “procurement” is considered as possibly relevant to the current knowledge stance. In addition, the activity definition includes a link to the “purchaser” and “shop assistant” roles which might guide the user to find other persons that are worth being contacted. Finally, the current action reveals that documents of type “experience report” might be of interest. The inclusion of these context elements can be achieved automatically by means of the “owl:TransitiveProperty” directive when using an OWL reasoner. This inference is depicted by the arrows in Figure 14.

The definition of the “search for related documents” task is assumed to specify the search portlet as the corresponding target application. Hence, when the user clicks the “Activate” button, the system will guide him to this portlet in order to search for related documents. As a result from the context-based search, the procurement guidelines document from Figure 7 will be found as relevant as it is annotated with the procurement topic. Also in combination with an explicit user query, it can be expected that the inclusion of the context information in the query will be significantly improve the search performance.

While the working context (from the knowledge stance model) is mainly defined on a type-level, the inclusion of interaction context elements (from the user’s interaction with other portlets during previous tasks) will also reveal instance information. Assume a newly introduced product is the “Shower Companion” within the “audio” product category and this information has found its way into the interaction context by browsing the product database. As a consequence, the search engine will also find the product experience report from Figure 7 with a particularly high score as it has the type “experience report” and deals with a product of the same product category.

Now, consider the next task of the “lookup related product experiences” action, which might be “contact experts for their opinion”. The task might point to a messaging portlet. By using the
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals

Figure 14. Example of a working context

```xml
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:inwiss="http://www.inwiss.org/schema#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
    <inwiss:process rdf:resource="http://www.inwiss.org/processes/Procurement"/>
    <!-- Inferred working context -->
    <inwiss:role rdf:resource="http://www.inwiss.org/roles/Purchaser"/>
    <inwiss:role rdf:resource="http://www.inwiss.org/roles/ShopAssistant"/>
    <dc:subject rdf:resource="http://www.inwiss.org/topics/Procurement"/>
  </rdf:Description>
</rdf:RDF>
```

working context information, the system can look for persons of the “purchaser” and “salesperson” roles. Note that Michael Bates (from Figure 10) will be found even though he works in a different department on a different process. The link exists solely through the “develop knowledge about products” activity.

So far, the interaction context has so far been volatile and bound to a user session in INWISS. As an extension we propose to bind it to a workflow instance. This way the context will persist for the lifetime of the workflow instance and can even be transported from one user to the other if different responsibilities are defined. Also when using collaboration technology sending the context along with the user messages seems promising as the message can automatically be enriched with and carry the current context. The recipient can thus easily use the portal to find information related to the message received.

CONCLUSION AND FUTURE WORK

This paper discussed how the concept of knowledge stance can be applied to portals which are an important technology to support knowledge work in the context of business processes. We presented the experiences from developing a prototype that applies Semantic Web technologies, proposed extensions and discussed how they can be implemented. A semantic description of information resources and therefore Semantic Web standards and technologies are constitutional for the implementation. The next steps are to develop the portal further based on our proposals. The open source workflow engine jBpm and the workflow editor JaWE may serve as a basis, however they need to be enhanced to support the activity-oriented perspective of knowledge stances. A modeling notation to model knowledge stances also needs to be defined.
So far, we considered the working context as statically resulting from modeling. As future work the context arising from occasions should not be considered statically defined but rather evolving so that knowledge stances can continuously evolve by gathering knowledge related to specific instances of processes, activities, and user actions. In addition, the context could also flow between users by means of knowledge-related actions and workflows. By this way, we provide means to communicate practices linked to a knowledge stance between multiple persons (e.g., that fulfill similar functions within a process) and to generalize proven practices and knowledge connected to a knowledge stance. This idea of communicating the context among users of course raises security and privacy issues. Finally, ways for automatic detection of occasions need to be studied, which could be based on detection of identifying patterns in the history of the interaction context.

Altogether, knowledge stance-oriented portals can be seen as a step towards making knowledge work in business processes more efficient by supporting integrated and context-oriented access to heterogeneous systems.

REFERENCES


A Context-Based Approach for Supporting Knowledge Work with Semantic Portals


ENDNOTES

a http://www.inwiss.org, last accessed April 1, 2005
b http://www.ibm.com/software/genservers/portal/, last accessed April 1, 2005
c http://www.sap.com/solutions/netweaver/enterpriseportal/, last accessed April 1, 2005
d http://www.openrdf.org, last accessed April 1, 2005
e http://jena.sourceforge.net, last accessed April 1, 2005
f http://www.dublincore.org, last accessed April 1, 2005
g http://portals.apache.org/jetspeed-1/, last accessed April 1, 2005
h http://www.microstrategy.com, last accessed April 1, 2005
i http://www.openrdf.org, last accessed April 1, 2005
j http://jakarta.apache.org/lucene/, last accessed April 1, 2005
k The concepts “Person” and “name” are borrowed from the Friend of a Friend (FOAF) project; http://www.foaf-project.org, last accessed April 1, 2005
l http://www.jbpm.org
m http://jawe.objectweb.org, last accessed April 1, 2005

This work was previously published in Int. Journal on Semantic Web & Information Systems, Vol 1, Issue 3, edited by A. Sheth and M. Lytras, pp. 64-88, copyright 2005 by IGI Publishing (an imprint of IGI Global).