2011 International Conference on Economics and Business Information
IPEDR vol.9 (2011) © (2011) IACSIT Press, Bangkok, Thailand

Context Model for Representation of Business Process Management Artifacts

Matthias Wieland[1+], Daniela Nicklas[2], Frank Leymann[1]

[1]Institute of Architecture of Application Systems, Universität Stuttgart, Germany
[2]Department for Computer Science, Carl von Ossietzky Universität Oldenburg, Germany

Abstract. Context-aware systems adapt their functionality and behavior to the user and his or her situation. To do so, they need context information about the user's environment, e.g., about different kinds of real world objects. To model and manage context information, many systems have been developed. An important part of context that is often neglected is the state and context of the applications that users are currently executing. The contribution of this paper is to present an extension for a standard context model that allows the representation of the context of workflow based applications. By that, business process management environments are enabled to annotate their context and provide it for other context-aware applications and users.

Keywords: business process management; workflows; context-awareness; human tasks; services

1. Introduction

Most context-aware systems are based on context models. Typically, these models contain data about real world objects and virtual information objects that are relevant at a certain location. The main purpose of a context model is to provide dynamic context data at runtime on request for different applications. Furthermore, based on the definition of context by Dey [1] the context of a context-aware application itself should be part of the context model as well. However, most existing context models do not allow the integrated capturing of the context data belonging to context-aware services, workflows and human tasks as well as their interrelations. The contribution of this paper is to present such an integrated context model for the most important artifacts of business process management (BPM). As a consequence, applications and users will be aware of the context and state of all workflows, services and tasks that are executed in an area, e.g., inside a production facility of a company. This additional context information allows new kinds of context-aware applications and enables a context-aware BPM.

The advantage of using workflow technology to develop context aware applications is that the application can be split up into different steps. Each step of the application (activity in a workflow) can be a service call or a human action. Single steps of the application can be monitored and represented in the context model. In standard (programmed instead of modeled) applications, the internal control flow is not visible for external users and applications. Hence, it is difficult to determine the context of a standard application. On the other hand, it is very important for context-aware workflows to be able to query dynamic up-to-date context information at runtime. In combination with a context model, such data can be easily made available for workflows. In this area, many works have been published and this is not the focus of the presented paper. Here we focus on the representation of the context of the BPM artifacts in a queryable model with the aim to provide context about all BPM systems. This brings all advantages of context-aware computing to the BPM area and enables a context-aware service registry where services may be selected based on their context, lookup of workflows responsible for a geographic area, and support of context-aware human tasks in workflows, to show the tasks on a map instead of an only role-based worklist. Furthermore, the provided context infor-

+ Corresponding author. Tel.: +49 711 685-88235; fax: +49 711 685-88472. *E-mail address: wieland@iaas.uni-stuttgart.de.*

mation allows displaying a map based overview of all available business processes, their execution state, their artifacts and their connection to the real world objects in the execution environment.

The reminder of the paper is structured as follows: Section 2 contains an overview on related work. Section 3 presents the main contribution of the paper, the context model for BPM. This model consists of three parts: workflow model, service model and task model. Section 4 explains shortly the implementation and the first prototype using the presented context model and Section 5concludes the paper.

2. Related Work

Many researchers addressed the need for explicit context management in a context model and of context usage in different applications (see [2] for a comprehensive survey). Early approaches adapted software design methods such as widgets or a network stack. However, they provided little support for building a general context model. Later, many comprehensive approaches for context model specification and management were presented like the Context Modeling Language (CML) and its associated software engineering framework [3]. CML provides a graphical notation to specify formally the context requirements of a context-aware application and allows automatic transform to a relational schema to manage the context information data in a database. The Nexus context model [4] with the Augmented World Language (AWML) supports the modeling and management of standardized and extensible context models. It already provides a schema for modeling context of services that became part of the BPM context model. Because of that, we used the Nexus context model in the paper and extended it for our needs to a full schema of a BPM context model. Furthermore, standards like the OpenGIS Web Feature Service (WFS) define an interface that allow request at context management systems for retrieving geographical features across the web using platform-independent calls. This leads to a further standardization of context management. The WFS, however, does not describe the schemas of the context models; this has to be done by each application on its own. All context models allow modeling any kind of data but in general do not provide a concrete context model schema for BPM.

There are languages available to capture models of workflows (e.g. BPEL, BPMN, YAWL, EPC, XPDL….). Extended workflow models for handling context information have been developed under the term of context-aware workflows [5, 6,7] and [8]. These models describe the control flow, dataflow and the service calls in the workflow as well as the usage of context-data in the workflow. The workflows themselves are not context providers, e.g., they do not tell where a workflow model is located or where the tasks the workflow initializes have to be executed. This has the disadvantage that no optimizations based on the context of the workflow parts can be done, e.g. proximity of a worker to a task or of a service to a workflow. This shows that a context model for BPM is needed. In [9] first concepts to model the context of workflows are presented, but there only the workflows and not the other required artifacts (services and human tasks) are considered in the model.

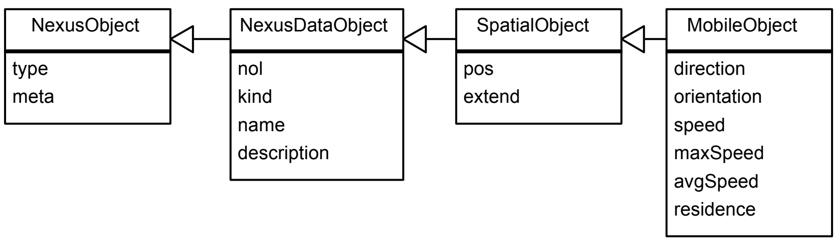

Fig. 1: Overview of the main classes of the Nexus context model

3. Context Model for BPM

Figure 1 presents an excerpt of the main classes of the used standard Nexus context model [4]. It consists of generic standardized objects (*SpatialObject*, *MobileObject*) inheriting from basic objects (*NexusObject*, *NexusDataObject*). This allows different application a shared usage of the context model: whereas application-specific characteristics can be expressed by specialized objects, these objects can always be mapped to generic objects, which then can be exploited by other applications. Extended schemas have to inherit from standard objects. Most objects in the Nexus context model are spatial world objects, e.g., *buildings* or *streets*.

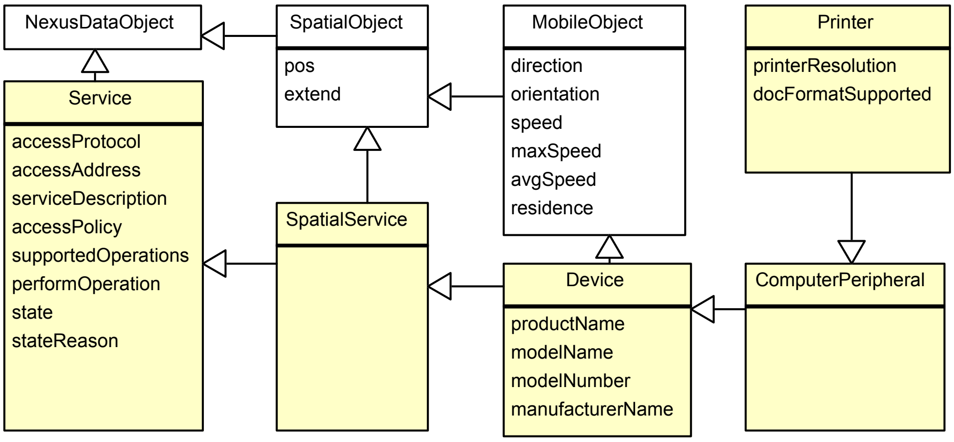

Fig.2: Context model extension for services

Every class represents a type of object and has different attributes that describe the state and context of the object. The classes are inheriting the attributes from their super classes as it is done in object-oriented programming. The BPM extension classes, presented in the following, contain the commonly needed attributes, but we do not claim that these are all attributes needed for any kind of application. Hence, it is easily possible to add new attributes if needed. The complete XSDs describing the extension formally are available at our web site[1]. An important detail is that the main data is still managed in the standard system, e.g. UDDI for services, workflow engine for workflows, task management system for human tasks. Only the added context data of each of these new objects is stored in the new context model for BPM. In the context model, a link is stored for each object to find the main data in the standard system. The standard system does not have to be changed and it does not have to know anything about context-awareness but still it can be used as if it was context-aware by using the context model instead.

3.1. Service Model

The Service Model allows modeling any kind of service. The Class *Service* shown in Figure 2 represents any digital available service like, e.g., a Web Service or an OSGi Service. The services and their operations are defined as attributes of the *service* class. Thus, the context model can be seen as a context-aware service registry. Furthermore, a *SpatialService* adds location context to a service. By this, context-aware service discovery can be supported and services get a location and an area where they can be used or are related to. Further subclasses are *devices* (e.g., a *printer*) which are at the same time spatial objects, mobile objects and services. Zhang presents further information in [10].

3.2. Workflow Model

Workflows are the center of the context model extension and represent workflow-based applications. That means the applications obtain a location or area where they are relevant. A workflow models the control flow of the application. It can be started for execution of the application. An executed *Workflow* is called *WorkflowInstance*. For both cases, different classes are available in the context model. Figure 3 shows the context model extension for representing the context of workflows. For a *workflow*, the corresponding workflow model installed in a workflow engine has to be linked and the service that can be used for starting the workflow has to be specified. For a *workflow instance*, it is important to know the state of execution, the processed data and the generated tasks. Workflows can be instantiated in parallel to execute the application logic at different areas. A failure management workflow is e.g. started any time a machine has an error at the location of the machine. The location of the *SpatialWorkflowInstance* is the location where the failure occurred, while the failure management workflow stays at the specified area, where it is responsible for repairing failures (e.g., a production facility). This is the area of responsibility of a *SpatialWorkflow*. Any object in the context model provides links to the original system where the original artifacts are stored. For a workflow, e.g. there is link to the workflow management system with the workflow model. In the context

[1] http://www.iaas.uni-stuttgart.de/nexus/2.0.2/SmartFactoryExtendedClassSchema.xsd

model, only the additional context information for the workflow is stored. Thus, it is avoided to produce redundant data.

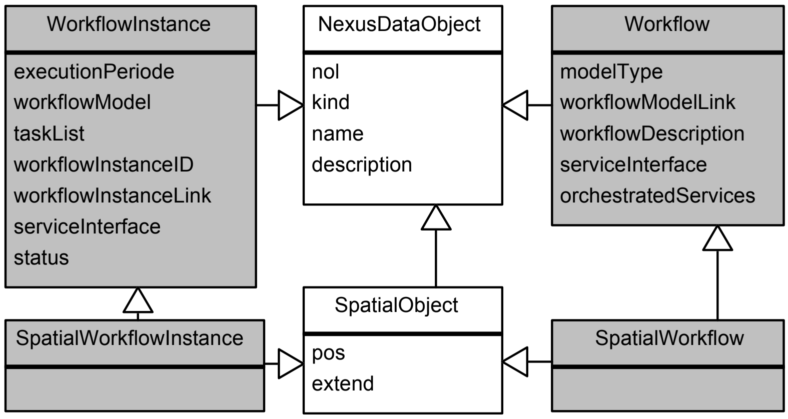

Fig.3: Context model extension for workflows

3.3. Human Task Model

Services are executed by computers and are used in workflows for automation. However, manual work is also required for executing workflows. So-called Human Tasks represent the manual work. These tasks are managed by a task management system that also provides a service interface for being accessed by the workflow. Furthermore, tasks are shown in user interfaces for the interaction with workers. Hence, the tasks have to be represented in the context model, which is shown in Figure 4. The context model allows a *task* to be a spatial object with location (pos) and to have an extent. By that, tasks can be presented in new kinds of user interfaces as a work map. This is very handy for workers because they can easily find work around their own current location or on their way.

We designed three subtypes of tasks for important basic work tasks: *QuestionTasks* for answering a question about a resource, *TransportTasks* for transporting a resource from location A to location B and *ActionTasks* for working on a resource in order to change its state. In any type of task, a resource plays an important role, thus, tasks are always connected to their real world objects. This allows optimizing the task behavior. For example, when the resource of the transport tasks arrives at the destination location B, the task can be completed automatically. Normally, tasks have to be completed by the users manually, which needs additional time and lowers the acceptance of the system. Further details on the implementation of the task model can be found in [11]. In this paper, it is also described how the task system can be implemented without tracking the workers, which is problematic from a privacy point of view.

4. Implementation of the Context Model

The context model, the BPM extension of the model and the server system for managing the context data are all implemented within the Nexus project[2]. The models are defined with XML schemas and a tool for easing the modeling of the schemas and the context data has been developed (Nexus Editor) [12]. The server architecture is based on Web Service interfaces and forms a federation of servers [4]. For different kinds of context data (e.g., high update rate or high amount of data) different servers have been developed. New client applications have been developed for the interaction of the users with the human tasks. An Android application provides a mobile work map and a web client provides an overview dashboard. A first prototype application has been built. It implements a failure management process for production environments [13]. This application shows the usefulness of the extended context model for BPM and allows us to evaluate the advantages of its practical usage. The concept of Context Integration Processes (CIP) [6] has been used for the insertion, updating and deletion of the context of all BPM artifacts. This means, e.g. if a spatial workflow is started the context of the workflow instance is inserted in the context model by calling the corresponding CIP.

[2] http://nexus.informatik.uni-stuttgart.de/

If the instance is changing its location, automatically a CIP for updating the context is executed and if the instance ends, the context is deleted by the delete context CIP. The CIPs are inserted in the normal workflows as subprocess calls to avoid a pollution of the workflows with the context management activities.

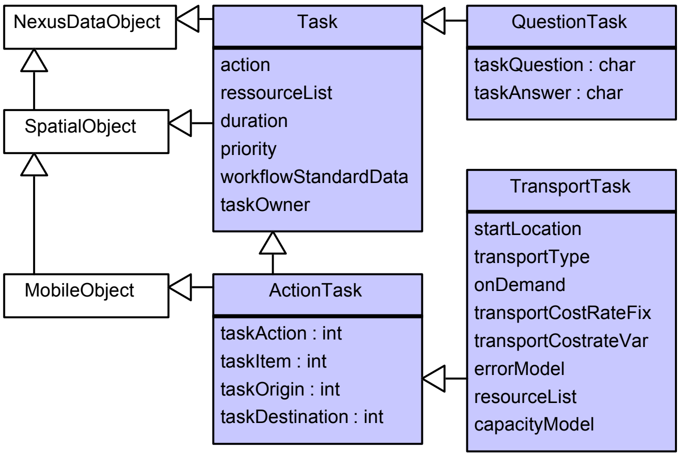

Figure 1: Context model extension for human tasks

5. Conclusions

In this paper, a context model for representing the context of workflow-based applications is introduced. The model consists of three parts, one for each main artifact in the BPM area: workflows (models and instances), human tasks and services. Using the presented context model, all these artifacts provide their own context, e.g. location of a human task. Thus, they get context annotated and by that allow other context-aware applications and users to sense their state and react on changes. For example, this allows a user to find the right workflow to repair a machine (context query for a repair workflow in the machine area), or show tasks on a map that are near a user. A further advantage of the new context model extension is that for each object, it may contain links to many other enterprise systems (e.g. the ERP, MES, Workflow System, SharePoint, Machines, Manuals, task management system, Service partners …) where the original data of the BPM artifacts is stored. This helps workers and managers to lookup the data conveniently using our implemented web client as a browsable directory of the enterprise systems. This provides a good overview of all systems that take part in the business process.

Finally, it is possible to simulate business processes before their rollout based on a context simulation. Here, the complete environment is modeled in a context simulation and agents simulate the actors (like e.g. workers or machines). The actors can then observe and interact with the workflows and tasks via the context model and can change the environment (e.g. transporting resources). Thus, the processes can be executed, visualized and optimized before rollout.

6. References

[1] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, 2001.

[2] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.

[3] K. Henricksen and J. Indulska, "Developing context-aware pervasive computing applications: Models and approach," *Pervasive and Mobile Computing*, vol. 2, pp. 37–64, 2006.

[4] M. Großmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas, and T. Schwarz, "Efficiently Managing Context Information for Large-Scale Scenarios," in *Proc. of the Third IEEE Intl. Conf. on Pervasive Computing and Communications*, 2005.

[5] L. Ardissono, R. Furnari, A. Goy, G. Petrone, and M. Segnan, "Context-Aware Workflow Management," in *ICWE*, ser. LNCS, vol. 4607. Springer, 2007, pp. 47–52.

[6] M. Wieland, P. Kaczmarczyk, and D. Nicklas, “Context Integration for Smart Workflows,” in *Proc. of the Sixth IEEE Intl. Conf. on Pervasive Computing and Communications*. IEEE, 2008, pp. 239–242.

[7] M. Heravizadeh and D. Edmond, “Making Workflows Context-aware: A Way to Support Knowledge-intensive Tasks,” in *Fifth Asia-Pacific Conference on Conceptual Modelling*, ser. CRPIT, vol. 79. ACS, 2008, pp. 79–88.

[8] H. Wolf, K. Herrmann, and K. Rothermel, “Modeling Dynamic Context Awareness for Situated Workflows,” in *OTM Workshops*, ser. LNCS, vol. 5872. Springer, 2009, pp. 98–107.

[9] O. Saidani and S. Nurcan, “Towards Context Aware Business Process Modelling,” in *Proc. of the 8th Workshop on Business Process Modelling, Development and Support*. Springer, 2007.

[10] Y. Zhang, “Dienstmodellierung in Nexus,” Diploma Thesis, University of Stuttgart, 2005, (in german).

[11] M. Wieland, C. Längerer, F. Leymann, O. Siemoneit, and C. Hubig, “Methods for Conserving Privacy in Workflow Controlled Smart Environments - A Technical and Philosophical Enquiry into Human-Oriented System Design of Ubiquitous Work Environments,” in *Proc. of the Third Intl. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Sliema, Malta: IEEE, 2009.

[12] N. Cipriani, M. Wieland, M. Grossmann, and D. Nicklas, “Tool support for the design and management of context models,” Inf. Syst., vol. 36, pp. 99–114, 2011.

[13] M. Wieland, F. Leymann, M. Schäfer, D. Lucke, C. Constantinescu, and E. Westkämper, “Using Context-aware Workflows for Failure Management in a Smart Factory,” in *Proc. of the Fourth Intl. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Florence, Italy: IARIA, 2010, pp. 379–384.