Benefits of Business Process Context for Human Task Management

Matthias Wieland, Daniela Nicklas, and Frank Leymann

***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. Many systems and data models are available for the management of context information. An often neglected but important part of context is the state and context of the applications that users are currently executing. This paper presents the benefits of using the context of workflow-based applications in the area of human task management. We show what kind of new task clients for mobile users are enabled by a context model for business process management and present an implementation of the system.**

***Index Terms*—business process management; workflows; context-awareness; human tasks; services.**

I. Introduction

Context models are very important because most context-aware systems use context models for retrieving their needed context. 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 Dye[6] the 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 relationships. This paper presents such an integrated context model for the most important artifacts of business process management (BPM) and it shows the advantages of using this context model. For example applications and users will beware 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 context-aware BPM.The main contribution of the paper is to show how human task management benefits of the introduction of a context model for BPM by presenting the newly enabled types of human task client applications.

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 already been published and this is not the focus of the presented paper. We focus on the representation of the context of the BPM artifacts themselves in a query able model with the aim to provide context about all BPM systems. This brings all the advantages of context-aware computing to the BPM area and enables context-aware service registry where services may be selected based on their context. It also enables look-up of workflows responsible for a geographic area, and context-aware human tasks in workflows to show the tasks on a map instead of an only role-basedworklist, which will be explained in, detail later. Furthermore, the provided context information 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 paper has four parts: Section II contains an overview on related work. Section 0explains the context model for BPM whichhasbriefly been presented in [21]. Based on this model Section IV explains the advantages of using such a context model for BPM in the area of human task management. Therefore, our implementation of a first prototype of a mobile task management application for android and a web based context explorer will be shown. Section V concludes the paper.

II. Related Work

Many researchers addressed the need for explicit context management in a context model and of context usage in different applications (see [3] 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,we present many comprehensive approaches for context model specification and management like the Context Modeling Language (CML) and its associated software engineering framework[8]. CML provides a graphical notation to formally specify 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.

Manuscript received July 14, 2011; revised July 31, 2011

M. Wieland is with the Institute of Architecture of Application Systems, Universität Stuttgart, Germany (wieland@iaas.uni-stuttgart.de).

D. Nicklas is with Department for Computer Science, Carl von Ossietzky Universität Oldenburg, Germany (dnicklas@acm.org).

F. Leymann is the head of the Institute of Architecture of Application Systems, Universität Stuttgart, Germany (leymann@ iaas.uni-stuttgart.de).

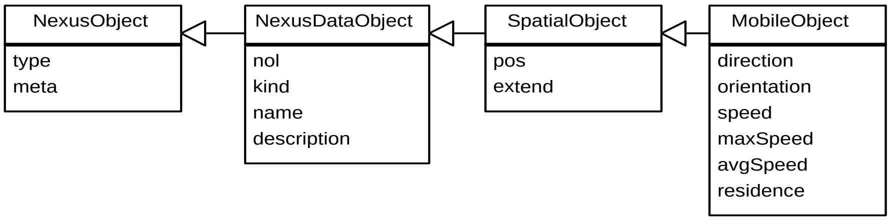

Fig. 1. Overview of the main classes of the Nexus context model

The Nexus context model [7] 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 Open IS Web Feature Service (WFS) define an interface that allows querying 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 [2, 18, 9] and [22]. 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 tube 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 [16] 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.

It is important to know that BPEL already supports integration of humans by the BPEL4People extension [1] or vendor specific solutions. For this purpose, a task management system distributes work tasks to human participants. However, there are no systems available that allow the modeling of context-aware tasks in a standard BPEL environment.

However, some research approaches exist in this area: xBPEL[4] is a BPEL language extension for modeling mobile participants in workflows. The Pergola system executes xBPEL and allows integration of people into BPEL workflows without constraining the users to their desktop PC. The WHAM System [10] supports mobile workforce and applications in workflow environments based on IBM products. In contrast to these solutions, our context-aware human task management builds on top of the already used WS-BPEL4People standard and can be used as addition on top of a running workflow system parallel to normal business-workflows and conventional human tasks.

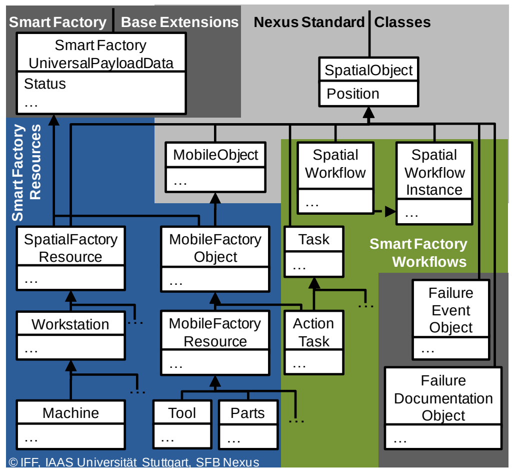

Fig. 2. Excerpt of the context data model for integration of workflow and production context based on the Nexus standard classes

III. Context Model for BPM

In this section, we present the context model for BPM: the underlying Standard Class Schema, the extensions for workflows, services, and human tasks, and finally a short overview on the management of the context model within the Nexus platform. Since this paper is an extension of [21], the first parts of this section are similar.

A. *Standard Class Schema*

Fig 1 presents an excerpt of the main classes of the used standard Nexus context model [7]. It consists of generic standardized objects (*Spatial Object*, *Mobile Object*) inheriting from basic objects (*Nexus Object*, *NexusDataObject*). This allows a shared usage of the context model for different applications: whereas specialized objects these objects can always can express application-specific characteristics be mapped to generic objects. They 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*. 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 we do it 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. It is an important detail that the main data is still managed in the standard system, e.g. UDDI for services, workflow engine for workflows, and task management system for human tasks.

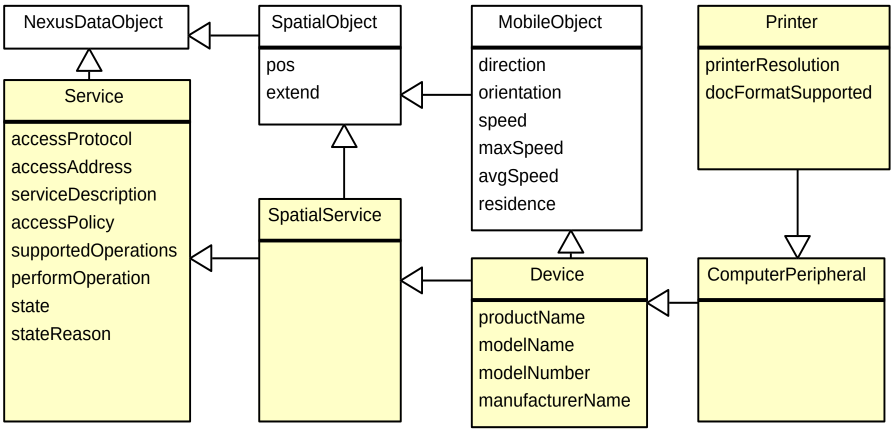

Fig. 3. Context model extension for services

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 it still can be used as if it was context-aware by using the context model instead.

B. *Extended Context Data Model*

For the integration of context information from different context models the, standardized context schema is needed. The *Nexus Standard Classes* represents the basic objects. Based on this standard schema, extensions can be defined for different application domains [7]. This has the advantage that the context data can be shared between different applications even from different domains. For the failure management in the Smart Factory we have defined such an extended context schema [12], [13] an excerpts shown in Fig 2. Similarly, the extension for the BPM context model is defined and both extensions can be used together at the same time by the applications.

The Smart Factory Data Model consists of several packages. The following packages are important in the failure management scenario: *Smart Factory Resources* for representing production resources such as machines and tools, and *Smart Factory Workflows* for the representation of processes and tasks context in the Smart Factory. This package is the context model for BPM extension that will be explained in detail in the following sections. The package *Smart Factory Base Extensions* contains a set of virtual objects representing documentation or detected failures. Further packages not in the focus of this paper refer to production orders, products, sensors and actuators. In order to ensure that objects of the Smart Factory have attributes describing location (position attribute) and condition (status attribute), they inherit these attributes from the Nexus Standard and Smart Factory Base Extension classes. The classes from *Smart Factory Base Extension* package contain additionally attributes for target locations (location specification), execution dates of tasks or document references, andto data such as manuals that is only rarely needed.

As Fig 2 shows in detail, the package *Smart Factory Resources* represents the real-world objects of a factory, such as buildings, production segments, machinery and equipment (tools, fixtures and testing equipment), storage resources and transportation means. The package *Smart Factory Workflows* contains an object for holding the context of BPM e.g., the failure management workflow with an area where it can be applied. In case a failure occurs an instance of a workflow (Spatial Workflow Instance) is created at the location of the handled failure event. The instance also creates human tasks (Task, Action Task) for the manual work that has to be done to repair the failure. The concepts and functionality behind context-awareservices, workflowsand tasks are described in the following.

C. *Service Model*

The Service Model allows modeling any kind of service. The Class *Service* shown in Fig 3 represents any digital available service like, e.g., a Web Service or agnostic 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 *Spatial*

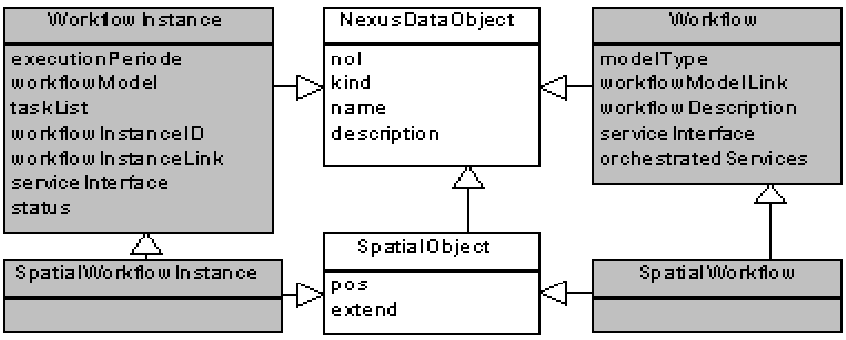

Fig. 4. Context model extension for workflows

Service adds location context to a service. Hence, 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[23].

D. Workflow Model

We called the special kind of Workflows that are able to control smart environments “Smart Workflows” [8]. They form the top most layer of the context-aware workflow concept and represent the application logic, which is shown in Fig 5. As most workflows they are orchestrating the control flow between automated programs as workflow activities. The smart workflows use Context Integration Processes (CIP) to access the context models in a domain specific application optimized way. This allows the modeling of the application by domain experts without technical knowledge of the context management system. Inside the Context Integration Process Layer the CIPs form a hierarchy of workflows calling each other as sub processes. The generic CIPs are used by every smart workflow and are the gateway to the context management system. Furthermore, the context task CIPs provide additional functionality to create and manage human tasks in a workflow system. They are providing the task context in the context model.

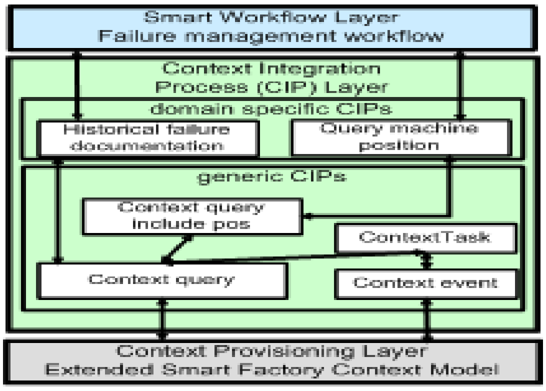

Fig. 5.Context Integration Process realizing context-aware workflows

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 the execution of the application. An executed *Workflows* called*WorkflowInstance*. For both cases, different classes are available in the context model. Fig 4 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, e.g., is 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 *Spatial Workflow*. Any object in the context model provides links to the original system where the original artifacts are stored. For a workflow, e.g., there isa link to the workflow management system with the workflow model. In the context model, only the additional context information for the workflow is stored. Thus, it is avoided to produce redundant data.

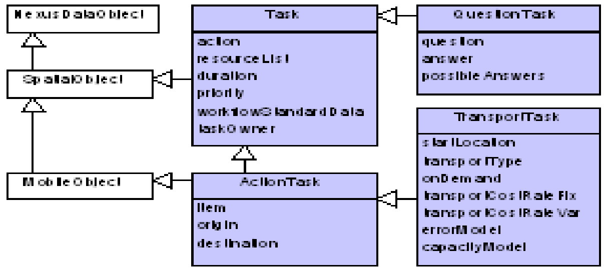

Fig. 6. Context model extension for human tasks

E. Human Task Model

Normally, computers can execute services without manual interaction and that is why workflows use orchestrated services for automation. Sometimes, however, manual work is also required for executing workflows. Human tasks can represent such manual work steps. These human tasks are managed by a task management system that provides a service interface for being accessed by the workflow. We have defined different types of context-aware tasks for different kinds of work. Fig 7 shows the most important ones. The tasks are classes in the context model and have different attributes. Consequently, the management of the tasks is split up between the workflow system and the context management system. The workflow system manages the state and execution of the tasks using a task list and a human task service. The context management system manages the context as for example the location of a task or the area where it is visible. By intersecting both the workflow data such as roles of people that are able to execute the task and the context data like the location of people, the amount of potential tasks per user can be reduced. Using that concept only tasks are shown on the work list that are e.g., within a radius of 100m of a user of the role worker. This helps users to select appropriate tasks faster than normally because the list contains fewer tasks and is context-aware. The context-aware tasks also have another advantage. They are connected to real world objects that cause the task. This information is important for the user because he does not have to search for the object but can locate it in a map. This also has important advantages for automation:

An *action task*can automatically be completed by monitoring the real world object. A *transport task*is completed when the object reaches its target location. An *operate task* is completed automatically when the object of work switches to the state OK again. This automatic completion saves time because the user does not have to go to the terminal in order to enter that he has finished the task. He only has to interact with the real world objects. Only for the *question task* a user has to put data into the computer for answering a question on the real world object he has examined. For example, the question could be “Please conduct a sight check of tool 132.” The answer can then be the remaining usage time of the tool. By that, context can be acquired or verified where no sensor is available or the sensor is not precise enough. Furthermore, tasks are shown in different kinds of user interfaces for the interaction with workers.

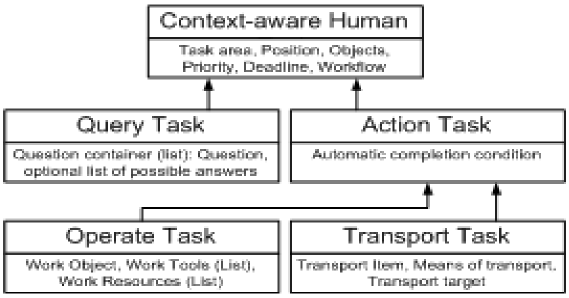

Fig. 7. Modeled types of context-aware tasks

Hence, the tasks have to be represented in the context model shown in Fig 6. The context model allows a *task* to be aspatial object with location (pos) and to have an extent. Thus,taskscan 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 onlydesignedthree subtypes in the context model of tasks for important basic work tasks: *Question Tasks* for answering a question about a resource, *Transport Tasks* for transporting a resource from location A to location B and*ActionTasks* for working on a resource in order to change its state. The operate task was not represented by a own class in the context model because it is very similar to the action task and can be expressed with the attributes of an action task. 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, and this needs additional time and lowers the acceptance of the system. Further details on the implementation of the task model can be found in [19]. In this paper, it is also described how the task system can be implemented without tracking the worker because this is problematic from a privacy point of view.

F. *Context Model Management*

The Nexus approach [7] federates various spatial context models which are stored in the so-called context servers (see Fig 8).

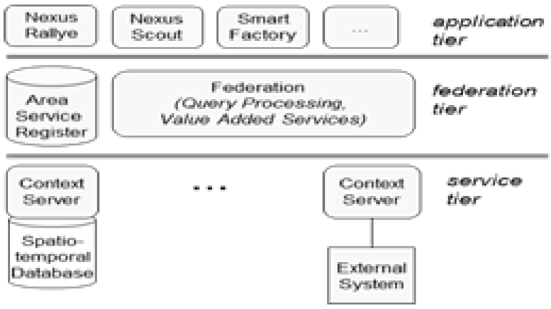

Fig. 8. Nexus Platform Architecture

The platform integrates context from various sources and provides a useful data integration abstraction for different context-aware applications. It contains context data of different kinds, like geographical data (map data), dynamic sensor data, infrastructural context, or links to related information (e.g., HTML documents).

The Nexus platform is organized in three tiers: service tier, federation tier, and application tier.

The *service tier* contains different context servers that hold the context data, gathered by sensors or modeled by data domain experts of the different context providers. Different application domains may contribute context data by maintaining own context servers. There are various realizations of context servers [7]: e.g., for map data, a DBMS with spatial extensions could be used [17], depicted as the left-most context server. Other context data from external systems can be integrated using wrappers [11] that could even run on separated systems, as can be seen on the right side. Only context servers hold the right to create modify and delete data objects. They are responsible for the integrity of their data. The integrated data model for applications is realized as a virtual integration of this local context model sat the federation tier.

The *federation tier* integrates context information. It holds numerous functional components that process the raw context data to provide more expressive data, and a registry for different context providers, the area service register. The federation service allows inserting, querying and modifying context data over different context sources. The event service provides an interface to register events and delivers notifications. Value added services offer additional functionality for convenience with proprietary interfaces (e.g., a map service to generate maps, or a navigation service for routing).

Context-aware applications are located at the *application tier* and use the underlying provided services. To evaluate the Nexus platform, several context-aware applications were developed, like the NexusRallye (a location-based game)[15], the Nexus Scout (a tourist information service)[14], or the detail in this paper.

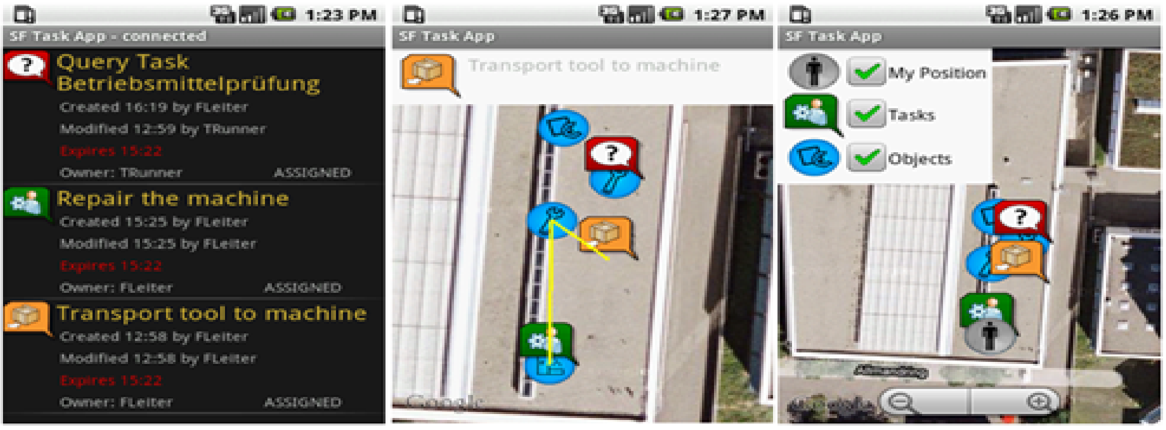

Fig. 9. Mobile task application showing a work list, a work map with selected transport task and the map filter

IV. Benefits for Human Task Management

We developed prototypes for innovative human task client applications for the interaction of the users with the human tasks. First, an Android application implements a mobile work map and seconds a web application implements at ask management dashboard. To test the application a context-aware workflow has been implemented that controls a failure management in production environments [20]. 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) [18] has been used for the insertion, updating and deletion of the context of all BPM artifacts. This means, if, e.g., a spatial workflow is started, the context of the workflow instance is inserted in the context model by calling the corresponding CIP. If the instance is changing its location, a CIP for updating the context is automatically executed and when the instance ends, the context is deleted by the delete context CIP. The CIPs is inserted in the normal workflows as sub process calls to avoid a pollution of the workflows with the context management activities.

A. Context Model Implementation

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 (For more information visit web site: http://nexus.informatik.uni-stuttgart.de/). 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) [5]. All XML schemas can be found on the project web site. The server architecture is based on Web Service interfaces and forms a federation of servers [7]. For different kinds of context, data (e.g., high update rate or high amount of data) different servers have been developed (see Section III.F).

B. Mobile Client Tsarap

Fig 9 and 10show screenshots of the mobile task client on a Google G1 android phone. This mobile application can be used anywhere on a mobile phone supporting android 1.6 or higher. The tasks are presented in two ways, firstly in Fig 9, left sides usual in a task list and secondly, what is new in the BPEL area as a task map in Fig 9,right side. This means the context of the tasks is used to display them on a map in addition with the objects that are related to the task. For example, the transport task has a dependency to the tool that has to be transported (see Fig 9 middle) and a connection to the machine what the target of the transportation is. Furthermore, the position of the mobile phone is shown as user location (icon with a person on a gray round background) for a better orientation of the user. However, the location of the user is never send to other systems by the mobile client for privacy reasons. Therefore, the location of the user always stays on the mobile client and the worker cannot be tracked.

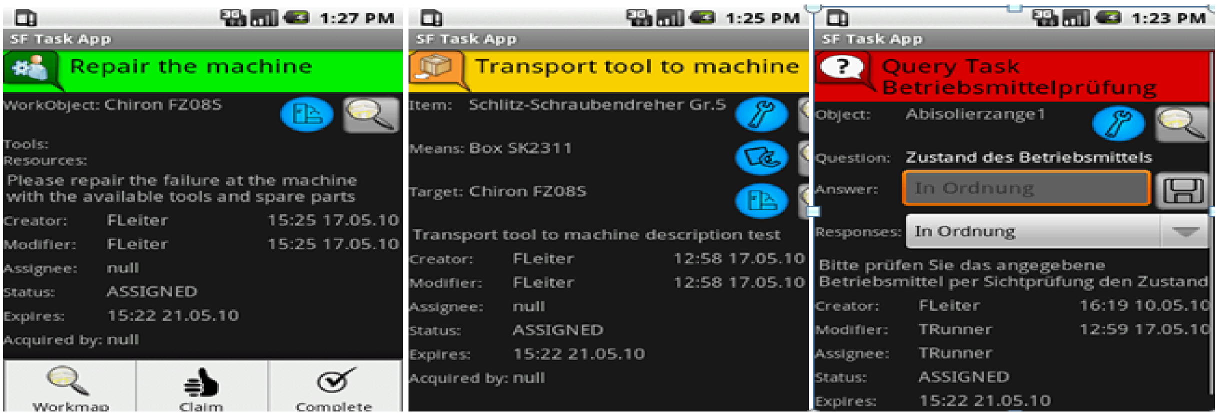

Fig.10. Mosbile task application showing task details

Fig 10shows a detailed view of the different task types. Here, all information of the task is shown, e.g. the connected real world objects are listed and can be selected to be displayed on the map. Furthermore, interactions with the task are possible, e.g. the task can be claimed and completed here by one fingertip. This is implemented by invoking the task management system from the mobile client and changes the state of the tasks there. One can also see whether the task is already been executed and who is working on it. In addition, the task can be shown on the work map for finding the shortest way. The colors represent the priority of the tasks (1-5: green over orange to red). The icons represent the type of task. The question mark represents a question task, the box with an arrow represents the transport task and the man with a rack-wheel stands for the operate task. The blue round icons represent work resources such as tools or spare parts.

C. *Web Client integration Portal*

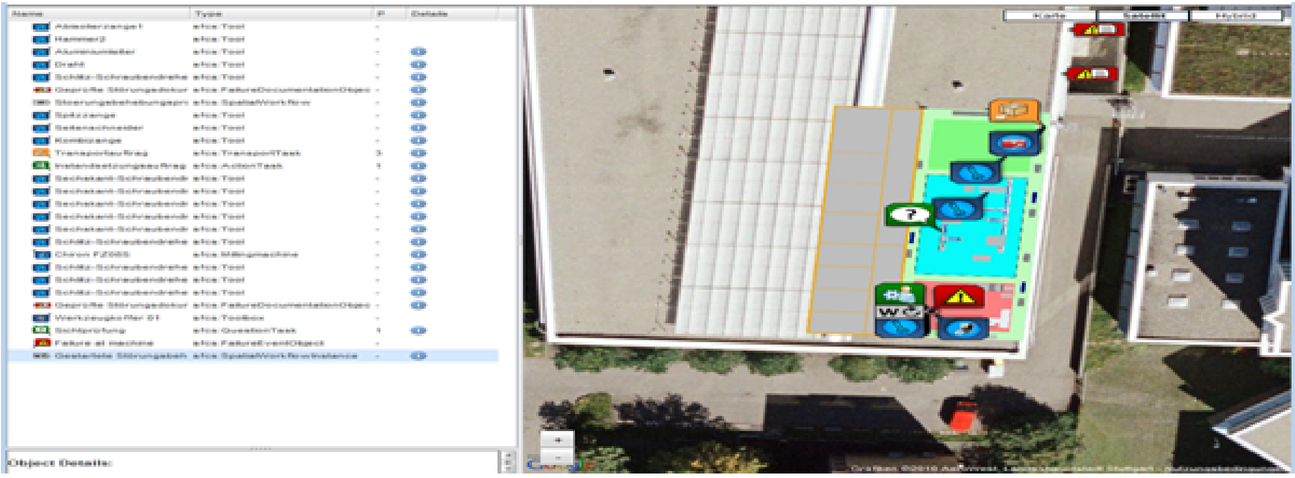

Fig.11. Web client showing combined objects of the Smart Factory and BPM context model

The Web Portal Application is needed together with the mobile client for the presentation of the whole factory context in an office scenario for analysis. Here, a much bigger screen size is available and the overview of the whole context can be achieved. Furthermore, because it is a web-based application nothing has to be installed on the client computers. Fig11shows a screenshot of the web client showing all the objects in the Smart Factory on a map. A failure is shown in red with a yellow exclamation mark. For this failure already a workflow instance is started to handle the a failure (gray W with round arrow). This workflow has started all needed task for repairing the machine: An transport task is shown in orange. An operate and query task in green. Again, the color represents the priority of a task. Furthermore, all needed spare parts and tools are shown as blue icons. It is important to note that no workers are tracked for privacy reasons and hence are not visualized in the map. The symbol of a person represents a human task, which can be executed by different workers. In addition, the re-localization of the task is only done indirectly e.g., on location change of the real world object connected to the task. A further advantage of the web client is that it provides links to all kind of IT systems in the production. E.g., it provides for workflow model and instance objects a link to the workflow system for presenting the workflow models, for starting workflow instances, and for observing workflow execution. Furthermore, it provides links to the workflow task management system for working on the tasks. Finally yet importantly, it links to factory information systems with additional information about the resource, e.g., manuals or 3D representations. For focusing on one aspect, a filter for object types is provided. For instance, only failure events and machines can be shown as an overview of the current state of the machines in the factory. Only tasks can be shown if the web-client is used as task map.

V. Conclusions

This paper describes a context model for representing the context of workflow-based applications. Later, the benefits of using this context model are explained based on the area of human task management. The context 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 implementation of new innovative context-aware applications. Two of these new applications were presented.

The first application is a mobile application that replaces a standard work list with a work map. This application allows a mobile worker to easily find tasks that are near the location of the worker. Furthermore, the worker can interact with the tasks (claim, complete …) and the needed resources for the task (tools, parts …). Without such a mobile application, the worker would have to walk to a terminal computer for any interaction with the system. Therefore, this new mobile task client saves working time and makes workers more efficient.

The second application is the web client. It gives an overview of all contexts available in the factory. This for example allows the user to find the right workflow to repair a machine (context query for a failure workflow in the machine area), or show tasks on a mobile task map that are near the 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 brows able directory of the enterprise systems. This provides a good overview of all systems that take part in the business process.

REFERENCES

[1] BPEL4People, OASIS WS-BPEL Extension for People. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bpel4people.

[2] Liliana Ardissono, Roberto Furnari, Anna Goy, Giovanna Petrone, and Marino Segnan. Context-Aware Workflow Management. In *ICWE*, volume 4607 of *LNCS*, pages 47–52. Springer, 2007.

[3] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[4] Dipanjan Chakraborty and Hui Lei. Pervasive Enablement of Business Processes. In *Proc. of the Second IEEE Intl. Conf. on Pervasive Computing and Communications (PerCom 2004), 14-17 March 2004, Orlando, FL, USA*, 2004.

[5] N. Cipriani, M. Wieland, M. Grossmann, and D. Nicklas. Tool support for the design and management of context models. *Inf. Syst.*, 36:99–114, 2011.

[6] Anind K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1), 2001.

[7] Matthias Großmann, Martin Bauer, Nicola Hönle, Uwe-Philipp Käppeler, Daniela Nicklas, and Thomas Schwarz. Efficiently Managing Context Information for Large-Scale Scenarios. In *Proc. of the Third IEEE Intl. Conf. on Pervasive Computing and Communications*, 2005.

[8] Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2:37–64, 2006.

[9] Mitra Heravizadeh and David Edmond. Making Workflows Context-aware: A Way to Support Knowledge-intensive Tasks. In *Fifth Asia-Pacific Conference on Conceptual Modelling*, volume 79 of *CRPIT*, pages 79–88. ACS, 2008.

[10] J. Jing, K. E. Huff, B. Hurwitz, H. Sinha, B. Robinson, and M. Feblowitz. WHAM: Supporting Mobile Workforce and Applications in Workflow Environments. In *RIDE*, pages 31–38, 2000.

[11] Othmar Lehmann, Martin Bauer, Christian Becker, and Daniela Nicklas. From Home to World - Supporting Context-aware Applications through World Models. In *Proc. of the Second IEEE Intl. Conf. on Pervasive Computing and Communications*, 2004.

[12] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Context Data Model, the Backbone of a Smart Factory. In *Sustainable Development of Manufacturing Systems: Proceedings of the 42nd CIRP Conference on Manufacturing Systems*, Grenoble, France, June 3-5 2009.

[13] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Fabrikdatenmodell für kontextbezogene Anwendungen: Ein Datenmodell für kontextbezogene Fabrikanwendungen in der Smart Factory. *wt Werkstattstechnik online*, 99(3):106–110, 2009.

[14] Daniela Nicklas, Matthias Grossmann, and Thomas Schwarz. NexusScout: An Advanced Location-Based Application On A Distributed, Open Mediation Platform. In *Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003*, pages 0–1. Morgan Kaufmann Publishers, September 2003.

[15] Daniela Nicklas, Nicola Hönle, Michael Moltenbrey, and Bernhard Mitschang. Design and Implementation Issues for Explorative Location-based Applications: the NexusRallye. In Gilberto Camara Cirano Iochpe, editor, *Proceedings for the VI Brazilian Symposium on GeoInformatics: GeoInfo 2004; November 22-24, 2004*, pages 167–181, Sao Jose dos Campos, November 2004. INPE.

[16] Oumaima Saidani and Selmin Nurcan. Towards Context Aware Business Process Modelling. In *Proc. of the 8th Workshop on Business Process Modelling, Development and Support*. Springer, 2007.

[17] Thomas Schwarz, Nicola Hönle, Matthias Grossmann, Daniela Nicklas, and Bernhard Mitschang. Efficient Domain-Specific Information Integration in Nexus. In Hasan Davulcu and Nick Kushmerick, editors, *Proceedings of the 2004 VLDB Workshop on Information Integration on the Web : IIWeb-2004 ; Toronto, Canada, August 30, 2004*, pages 122–127. online, August 2004.

[18] Matthias Wieland, Peter Kaczmarczyk, and Daniela Nicklas. Context Integration for Smart Workflows. In *Proc. of the Sixth IEEE Intl. Conf. on Pervasive Computing and Communications*, pages 239–242. IEEE, 2008.

[19] Matthias Wieland, Carsten Längerer, Frank Leymann, Oliver Siemoneit, and Christoph 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, 2009. IEEE.

[20] Matthias Wieland, Frank Leymann, Michael Schäfer, Dominik Lucke, Carmen Constantinescu, and Engelbert 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*, pages 379–384, Florence, Italy, 2010. IARIA.

[21] Matthias Wieland, Daniela Nicklas, and Frank Leymann. Context Model for Representation of Business Process Management Artifacts. In Chun Hua Lin and Ming Zhang, editors, *International Proceedings of Economics Development and Research: IPEDR*, volume 9 of *Economics and Business Information*, pages 46–51. IACSIT PRESS, May 2011.

[22] Hannes Wolf, Klaus Herrmann, and Kurt Rothermel. Modeling Dynamic Context Awareness for Situated Workflows. In *OTM Workshops*, volume 5872 of *LNCS*, pages 98–107. Springer, 2009.

[23] Yijie Zhang. Dienstmodellierung in Nexus. Diploma thesis, University of Stuttgart, 2005. (in german).

Dipl.-Inf. Matthias Wieland is research assistant at the Institute of Architecture of Application Systems managed by Prof. Dr. Frank Leymann. He works in the DFG Collaborative Research Center "Nexus: Spatial World Models for Mobile Context-Aware Applications". His current research interests are context-aware workflows and how such workflows can support processes in production environments.

Daniela Nicklas is Junior Professor for database- and internet technology at the Department for Computer Science, University of Oldenburg. She received her Dr. rer. nat. (PhD) in Computer Science from the Universität Stuttgart in 2005, working on spatial context models and architectures for location-based applications, and her Diplom (MSc) in 2000, also in Computer Science, with an emphasis on Software Engineering. Her current research interests are mobile applications, context-aware computing and data stream technologies.

Frank Leymann is a full professor of computer science and director of the Institute of Architecture of Application Systems at the University of Stuttgart, Germany. His researches interests include service oriented computing and middleware, workflow- and business process management, Cloud Computing, transaction processing, integration technology, and architecture patterns.

Before accepting his professor position he worked for two decades for IBM Software Group building database and middleware products: He built tools supporting conceptual and physical database design for DB2; built performance prediction and monitoring tools for an object database system; was co-architect of a repository system; built both, a universal relation system as well as a complex object database system on top of DB2; and was co-architect of the Miseries family. In parallel to that, Frank worked continuously since the late eighties on workflow technology and became the father of IBM's workflow product set.

As an IBM Distinguished Engineer and elected member of the IBM Academy of Technology, he contributed to the architecture and strategy of IBM's entire middleware stack as well as IBM's On Demand Computing strategy. From 2000 on, Frank worked as co-architect of the Web Service stack. He is co-author of many Web Service specifications, including WSFL, WS-Addressing, WS-Metadata Exchange, WS-Business Agreement, the WS-Resource Framework set of specifications, WS-Human Task and BPEL4People; together with SatishThatte, he was the driving force behind BPEL4WS. In addition, he is co-author of BPMN 2.0.