Using Context-aware Workflows for Failure Management in a Smart Factory

Matthias Wieland, Frank Leymann, Michael Schäfer
Institute of Architecture of Application Systems
Universität Stuttgart, Germany
{firstname.lastname}@informatik.uni-stuttgart.de

Dominik Lucke, Carmen Constantinescu,
EngelbertWestkämper
Institute of Industrial Engineering and Management (IFF)
Universität Stuttgart, Germany
{firstname.lastname}@iff.uni-stuttgart.de

Abstract—**In factories many processes are executed in parallel. The manufacturing processes are managed by Manufacturing Execution Systems. In the case of machine failures these systems provide only rudimentary or no support to the workers or shop-floor managers. As a consequence the failures have to be fixed as fast as possible for being able to continue manufacturing processes. For such cases context-aware workflows can be used to support the workers and to coordinate the work that has to be done for repairing purposes. In the Nexus Project we introduced the concept of context-aware workflows and context integration processes to be able to implement all kinds of processes going on in a smart environment. As a case study we modeled a failure management process as a workflow and executed it in a factory. Furthermore, we show the concepts behind this kind of workflows: the context integration processes and the context-aware human tasks. Finally, end user applications for the interaction of the workers with the workflow are presented. For that we developed an application concept providing a mobile solution for workers and a web-based solution for an office environment. The main contribution of this paper is to show how to implement such a failure management process as a context-aware workflow.**

Keywords-context-awareness; context models; workflow systems; human tasks; production environments; humane system design; mobile applications; ubiquitous computing

I. INTRODUCTION

Workflow technology as automation method gained major influence in many enterprises and within the software industry. It provides methods and corresponding products to support modeling, execution, and management of business processes that have been carried out manually and through a diversity of non-integrated systems before. The integration of these non-integrated systems is possible by employing workflow management systems. With modeling their processes as workflows (instead of hard-coding them into their systems), the enterprises were enabled to change their processes more easily; they became more flexible and adaptive, to survive within a more and more dynamic market [1]. However, there are major application domains where processes are not supported by workflow technology yet like the domain of manufacturing environments. Here, the processes are planned and executed using Manufacturing Execution Systems (MES) or Enterprise Resource Planning systems (ERP). A difficulty of these technical processes is that they are crossing the boundary to the physical world.

A commonly used context definition by Dey [2] states that context is "any information that can be used to characterize the situation of entities". In this definition entities are real world objects. So, a fundamental difference between traditional business processes and manufacturing processes is that within manufacturing processes context data has to be used that is captured based on these real world objects. Since miniature sensor technology and wireless communication becomes ubiquitous, it is possible to capture and observe more and more of these real world events. This leads to the idea of a Smart Factory, which allows the (automatic) collection and distribution of information, knowledge, and tasks to all work places based on physical events [3].

The Smart Factory approach developed at the Universität Stuttgart, defines a Smart Factory as a factory that is context-aware and assists people and machines in execution of their tasks by using context [4]. Here, context-aware means that the system can take context information as the location and status of a factory object, like the position of a tool and its working state (in work, damaged, etc.) into consideration.

Failure management in manufacturing is a very important topic. Only a rudimentary support of failure management is available in the standard manufacturing management systems, e.g., a notification of a failure via SMS or email is possible. This is caused by the high complexity and the diversity of failures. To improve failure management we show in this paper our approach of supporting failure management by coordinating and supporting the repair process with context-aware workflows. Furthermore, we show how humans can be integrated into context-aware workflows since human work is needed in the repair process. Based on this real-world scenario of failure management we explain the concepts we used to implement a prototype with a set of context specific end user interfaces.

The paper is structured as follows. Section 2 describes related work and fundamentals. Section 3 describes the basis for the system which is the *extended context data model*. In Section 4 the main concepts of *context-aware workflows* the context integration processes and the *context-aware human tasks* are introduced. Section 5 presents the *failure management process*. Finally, in Section 6 the prototype implementation is presented by explaining the *architecture* for execution of the context-aware workflow and the *end user applications* for human interaction with the context-aware human tasks. As well, *measurements* of the prototype implementation are presented. Section 7 concludes the paper and points out future work.

II. RELATED WORK AND FUNDAMENTALS

The Workflow Management Coalition defines workflows as "the computerized facilitation or automation of a business process, in whole or part" [5]. That means the execution of workflows is fully computerized. In contrast, business processes describe the progress of work done in an enterprise in a form that is not directly executable by workflow systems. However the idea of ubiquitous and pervasive computing and the idea of context-aware and mobile computing open up new possibilities for using business processes [6]. The benefit of integrating context-aware computing into workflows is to extend the application area of workflows from business processes to manufacturing processes. This is a big advantage because now a company can support the production with the same technology already used in the back office or administration [7]. In [8], we have already discussed all the main technical issues about context-aware workflows. In this paper however, we want to enhance the concepts for allowing the integration of humans into context-aware workflows. This allows building ubiquitous work environments using the concept of context-aware human tasks.

Business processes are normally modeled using graphical notations such as the Business Process Modeling Notation (BPMN) [9] or the event-driven process chains (EPCs) [10]. None of them contain explicit elements for modeling context-aware processes. For the automated execution with a computer, the processes must be expressed as workflows using a workflow execution language like YAWL [11] or WS-BPEL [12]. Since WS-BPEL (or BPEL for short) became the standard in this area, it is used as basis for the development and execution of our context-aware tasks.

Note that BPEL already supports integration of humans by the BPEL4People extension [13] or vendor specific solutions. For this purpose, a task management system distributes work tasks to human participants. 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 [14] is a BPEL extension for modeling mobile participants in workflows. The PerCollab system executes xBPEL and allows integration of people into BPEL workflows without constraining the users to their desktop PC. The WHAM System [15] supports mobile workforce and applications in workflow environments based on IBM products. In contrast to these solutions our system builds on top of the already used WS-BPEL4People standard and can be used as addition on top of a running workflow system in parallel to normal business-workflows and conventional human tasks.

III. EXTENDED CONTEXT DATA MODEL

The Nexus approach [16] federates various spatial context models, which are stored in the so-called context servers. For the integration of context information from different context models a standardized context schema is needed. In the Nexus project we decided to build an object-oriented data model for that purpose. The basic objects are represented in the *Nexus Standard Classes*. Based on this standard schema extensions can be defined for different application domains [16]. 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 [17], [18] of which an excerpt is shown in Figure 1.

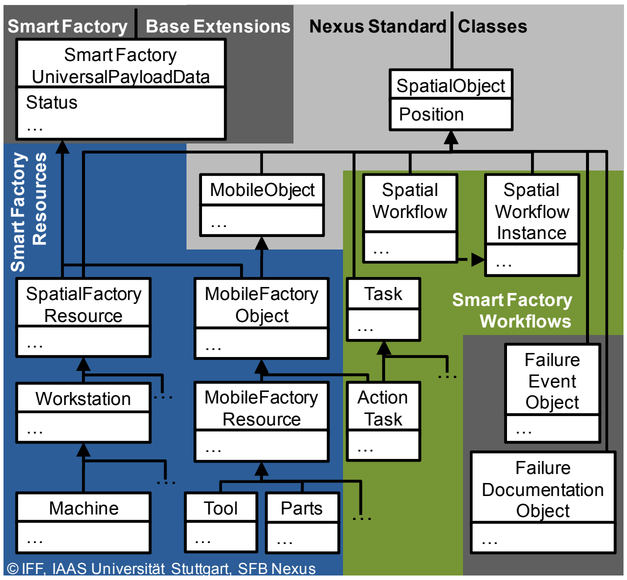

Figure 1 Excerpt of the context data model for integration of workflow and production context based on the Nexus standard classes

The Smart Factory Data Model consists of several packages. Following packages are important in the failure management scenario: *SmartFactory Resources* for representing production resources such as machines and tools. *Smart Factory Workflows* for the representation of processes and tasks context in the Smart Factory. The package *SmartFactory 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 actors. 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 to rarely needed data such as manuals.

Like Figure 1 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 a workflow model (Spatial Workflow), 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

 ISBN: 978-1-61208-100-7

failure. The concepts and functionality behind context-aware workflows is described in the following section.

IV. CONCEPTS OF CONTEXT-AWARE WORKFLOWS

In this section the needed concepts for realization of context-aware workflows with human interaction are shortly presented. For further details of the basic concepts other papers are already published that present them in more detail [8], [6], and [19]. Hence, in this section only the most important concepts are explained for understanding the realization of failure management workflows.

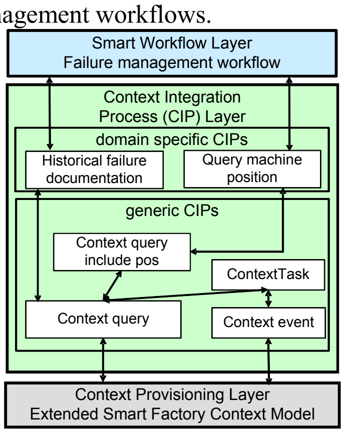

Figure 2 Context Integration Process realizing context-aware workflows

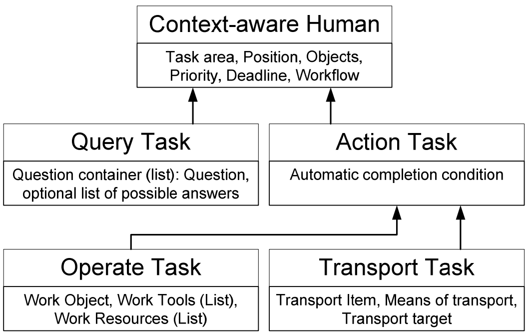

Figure 3 Modeled types of context-aware tasks

A. *Workflow Concepts*

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 (see Figure 2) and represent the application logic. 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 that the applications can be modeled by domain experts without technical knowledge about 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 and providing the task context in the context model.

B. *Context-aware Human Tasks*

The tasks are the way to integrate human work into the automated workflows. This is needed in order to complete manual work with workflows. We have defined different types of tasks for different kinds of work. The most important ones are shown in Figure 3. The tasks are classes in the context model and have different attributes, which are inherited from the super classes. As a consequence 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 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., in a radius of 100m near 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. Furthermore, the context-aware tasks have another advantage. They are connected to real world objects, which are the reason for the task. This information is important for the user, so he does not have to search for the object but can locate it in a map. Furthermore, this has also advantages for automation: An action task can be automatically completed by monitoring the real word 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 a user does not have to go to a terminal 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 input data into the computer for answering a question about the real world object he has examined. For example a question could be: “Please conduct a sight check of tool 132”. The answer can 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.

For implementing all the task types each is modeled as a CIP that has application logic for creating the task in the workflow system and in parallel insert it into the context model. After the task is completed it is deleted out of the context model. Most important, while the task is executed the contexts of its connected real world objects are monitored and on occurrence of the completion condition the CIP completes the task using the privileges of the Context management system.

V. FAILURE MANAGEMENT PROCESS

In a factory many processes are executed in parallel, e.g., different manufacturing and maintenance processes. The processes are executed normally with Manufacturing Execution Systems (MES).

ISBN: 978-1-61208-100-7

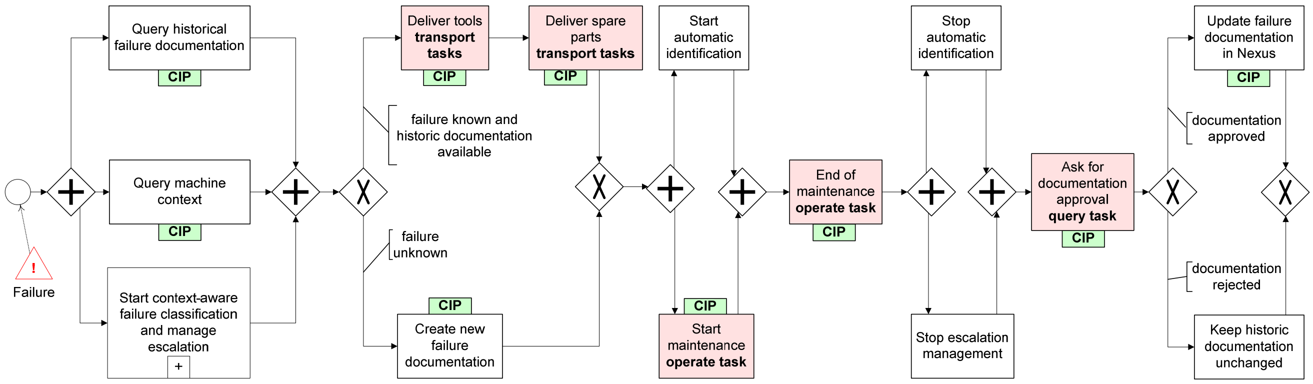

Figure 4 Context-aware workflow for handling a failure in the Smart Factory

In case of machine failures only rudimentary or no support is provided to the workers or shop-floor managers by these systems. In situation the workers are usually on their own to fix the failure as fast as possible for being able to continue manufacturing processes. For such cases context-aware workflows can be used to support the workers and coordinate the work that has to be done for repairing the failure efficiently. In the Nexus Project we introduced the concept of context-aware workflows and context integration processes to be able to implement all kinds of processes going on in a smart environment. As a case study we modeled this failure management process as a workflow and executed it in a real factory. The main contribution of this paper is to explain how to implement such a process as context-aware workflow.

Figure 4 shows a simplified version of the process for handling a failure. The complete BPEL code is accessible on our web-site [20]. First the workflow calls Context-Integration Processes (CIP) as sub-workflows to gather all needed context information about the failure: the context of the machine (location) and historical documentation about similar failures to analyze how and with what tools and parts the failure was repaired the last lime. In parallel the failure is classified in its seriousness and effects on the complete production. Then a branch is made whether a solution to the failure is already known or not. If a solution is available in historic documentations the needed tools and spare parts for the maintenance have to be transported to the machine with the failure. This is done using *transport tasks*. When the resources reach the target location the transport tasks are completed automatically by a context event. After that the maintenance is started by creating an *operate task*. In parallel to the task execution a documentation system monitors automatically, which tools and spare parts are actually used by the worker. Technical, the identification of tools and spare parts is implemented by using RFID tags. Due to harsh environmental conditions (e.g. metal influence, dirt and water) a object specific selection and mounting of the RFID tags for a reliable identification is required. In the prototype we did this for important tools (e.g. different types of screwdrivers, ladders, transport boxes, …). The *operate task* monitors the machine switching from failure state to OK state and completes then automatically. At the end of the workflow the automatic gathered documentation is presented in a *query task* to the worker for approval. Now the worker can decide whether he wants to edit and afterwards publish the automatically recorded documentation in the context model for improvement of future failure handling or if he wants the documentation to be deleted for privacy reasons.

VI. Implementation

We have created a prototypical implementation of the described concepts and modeled the failure management workflow. For that we first designed the extended Smart Factory context data model. Then we developed the task models and deployed all to the context servers. After this we modeled the context-aware workflows with BPEL (CIPs and failure management workflow) for coordinating the work in the Smart Factory. We also developed an application concept with different applications (mobile and web-based) for accessing the work tasks in distinct environments (on the move and in office). With this tools available context data stored in any context server can be visualized in order to get an overview of the factory.

A. Architecture

The needed architecture for executing the context-aware workflows consists of the following important parts (shown in Figure 5): The context management system implemented by the Nexus Federation. It manages the context model in a distributed scalable system. The second part is the workflow system that consists of the Workflow Management System (in our implementation it is an Oracle BPEL engine), the human task management service and an application server for hosting web-applications. The applications are used as user interface for the employees to conduct the human tasks and also for managing the executed workflows. For the management the standard Oracle BPEL dashboard can be used. But the human task application has to be extended to take the context of the tasks into account. Therefore, we developed a new web-client that shows all tasks and connected real world objects on a map (Figure 8). By that the context of the tasks is visible. Furthermore, a mobile application for accessing the tasks on a mobile device is required (Figures 6, 7). This results from the environment of the smart workflows. Normal workflows are used in business environments where all

 ISBN: 978-1-61208-100-7

users have a desktop computer and are not walking around all the time. In a smart environment, such as a smart factory, the users are mobile at all times. As a consequence they have to carry their task list with them. Furthermore, the mobile device knows the current location of the user and can filter the tasks by proximity. We implemented both applications and tested them in a Smart Factory on their usefulness and performance (see Section D).

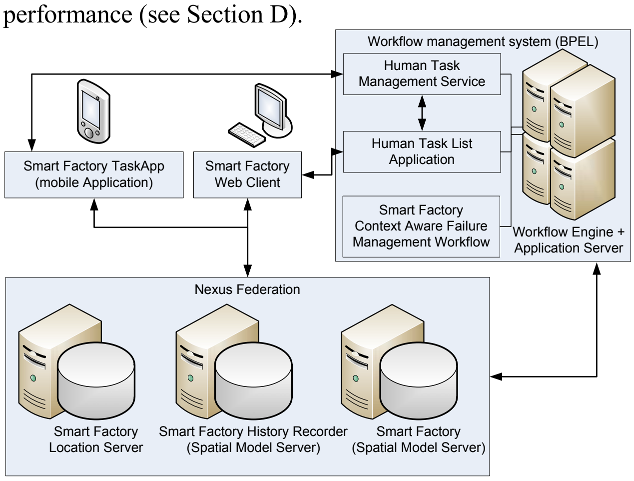

Figure 5 Architecture for execution of context-aware workflows

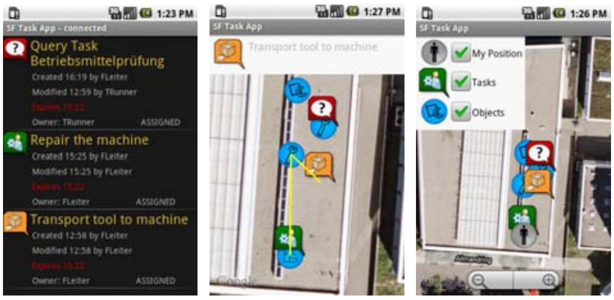

Figure 6 Mobile task application showing a work list, work map with selected transport task and the map filter

B. *Mobile Client TaskApp*

Figures 6 and 7 show screenshots of the mobile application on a Google 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 as usual in a task list (Fig. 6 left side) and secondly what is new (on a mobile device based on BPEL4People) as a task map (Fig. 6 right side). This means the context of the tasks is used to display them in a map in addition with the objects that are related to the task. Furthermore, the position of the mobile phone is shown as user location for a better orientation of the user. For example the transport task has a dependency to the tool that has to be transported (see Fig. 6 middle) and a connection to the target machines where the target of the transport is located.

Figure 7 shows the details view of the different task types. Here all the connected real world objects are listed and can be selected to be shown on a map. Furthermore, the task can be claimed and completed here by one fingertip. This calls the workflow system and changes the state of the task. It is also shown if the task is already executed and who is working on it. Also 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).

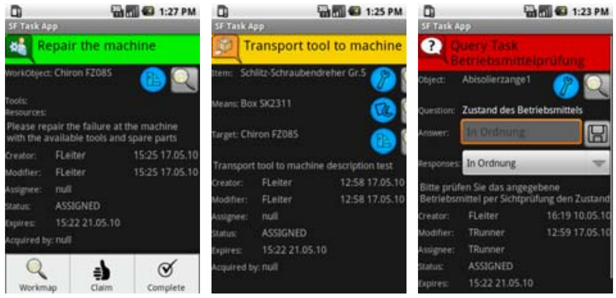

Figure 7 Mobile task application showing task details

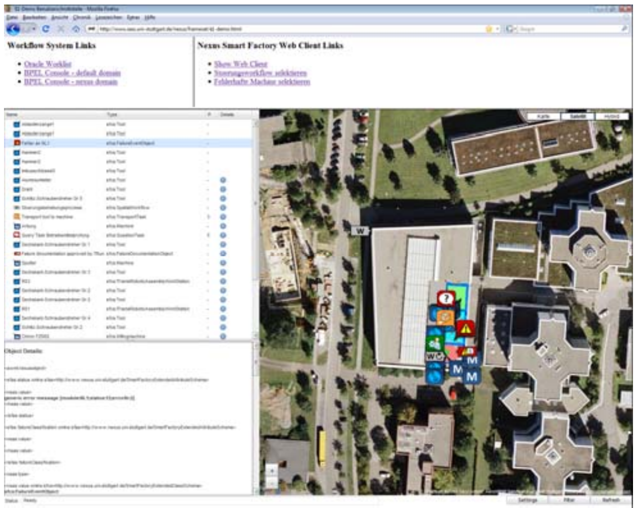

Figure 8 Web client showing context objects in the Smart Factory

C. *Web Client integration Portal*

The Web Portal Application is needed in parallel to the mobile client for 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, nothing has to be installed on the client computers. Figure 8 shows a screenshot of the web client showing all the objects in the Smart Factory on a map. An operate task is shown in orange. Again, the color represents the priority of a task. It is important to note that no workers are tracked for privacy reasons and hence are not visualized in the map. The symbol for a person represents a human task, which can be executed by different workers. Also 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 the workflow system for presenting workflow models for starting the workflow and workflow instances for observing their execution state. Furthermore, links to the workflow task management system for working on the tasks.

 ISBN: 978-1-61208-100-7

And last but not least 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. Or only tasks can be shown if the web-client is used as task map.

D. Runtime Measurements

We conducted measurements of both the web-client and the mobile client to find out the limitations of our solution. Table 1 shows the time (in seconds) for processing the data 1, 10, 50 and 100 tasks on the mobile client running on two different mobile Android phones and on the web client. Processing means to query the data from all sources and integrate it. The visualization is done in parallel and is not measured here. The result is that the system is able to handle at least up to 100 tasks in parallel. The refresh time on the mobile clients however is very high because of the limited processing power on the mobile devices. But for the user of the applications it is not noticeable that the updates take long, because the update is executed in a background thread. The fast execution of the web-client results from the fact, that only the data from the context model is used in the web client and no workflow data has to be integrated.

TABLE I. RESULTS OF THE RUNTIME MEASUREMENTS

Type of Client and Device Brand	Amount of Tasks			
	1	*10*	*50*	*100*
Web-Client - Firefox	0,08s	0,11s	0,36s	0,45s
Mobile-Client-HTC G1	1,63s	8,35s	37,82s	82,27s
Mobile-Client-Motorola Milestone	1,50s	5,10s	28,58s	53,49s

VII. CONCLUSION AND FUTURE WORK

In this paper we showed how to support the failure management in production environments by implementing this process as a context-aware workflow. The workflow uses context-aware human tasks for integration of the human workers executing the process.

Future work is to find a method how to allow the worker to work with the mobile applications using speech recognition to keep his hands free. In addition, performance improvements will be done on the mobile client for achieving a faster update of the task list for being able to handle more tasks in parallel.

ACKNOWLEDGMENT

This interdisciplinary research is funded by the Collaborative Research Center *Nexus: Spatial World Models for Mobile Context-Aware Applications* (grant SFB 627).

REFERENCES

[1] F. Leymann and D. Roller, *EnglischProduction Workflow - Concepts and Techniques*. PTR Prentice Hall, Januar 2000.

[2] A. K. Dey, “Understanding and Using Context,” *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001.

[3] E. Westkämper, L. Jendoubi, M. Eissele, and T. Ertl, “Smart Factory - Bridging the Gap Between Digital Planning and Reality,” *Manufacturing Systems*, vol. 35, no. 4, pp. 307–314, 2006.

[4] D. Lucke, C. Constaninescu, and E. Westkämper, “Smart Factory. A Step towards the Next Generation of Manufacturing,” *CIRP Manufacturing Systems*, vol. 42, 2008.

[5] WfMC, “Workflow Management Coalition.” [Online]. Available: http://www.wfmc.org/

[6] M. Wieland, O. Kopp, D. Nicklas, and F. Leymann, “Towards Context-Aware Workflows,” in *CAiSE´07 Proceedings of the Workshops and Doctoral Consortium Vol.2, Trondheim, Norway, June 11-15th, 2007*, B. Pernici and J. A. Gulla, Eds. Tapir Acasemic Press, June 2007, Workshop Paper, pp. 577–591.

[7] M. Wieland, F. Leymann, L. Jendoubi, D. Nicklas, and F. Dürr, “Task-orientierte Anwendungen in einer Smart Factory,” in *Mobile Informationssysteme - Potentiale, Hindernisse, Einsatz. Proceedings MMS´06*, ser. Lecture Notes in Informatics (LNI), T. Kirste, B. König-Ries, K. Pousttchi, and K. Turowski, Eds., vol. P-76. Bonn: Gesellschaft für Informatik, February 2006, Conference Paper, pp. 139–143.

[8] M. Wieland, P. Kaczmarczyk, and D. Nicklas, “Context Integration for Smart Workflows,” in *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications*. Hong Kong: IEEE computer society, March 2008, Conference Paper, pp. 239–242.

[9] OMG, “Business Process Modeling Notation, V1.1,” http://www.omg.org/spec/BPMN/1.1/PDF, 2008.

[10] A.-W. Scheer, O. Thomas, and O. Adam, “Process Modeling using Event-Driven Process Chains,” in *Process-Aware Information Systems*, M. Dumas, W. Aalst, and A. Hofstede, Eds. Wiley & Sons, 2005, pp. 119–146.

[11] W. M. P. van der Aalst and A. H. M. ter Hofstede, “YAWL: Yet Another Workflow Language,” *Information Systems*, vol. 30, no. 4, pp. 245–275, 2005.

[12] *Web Services Business Process Execution Language Version 2.0*, http://docs.oasis-open.org/wsbpel/2.0/, OASIS Std.

[13] *WS-BPEL Extension for People*, http://www.oasis-open.org/committees/bpel4people/charter.php, OASIS Std.

[14] D. Chakraborty and H. 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, pp. 87–100.

[15] 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*, 2000, pp. 31–38.

[16] 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, pp. 331 – 340.

[17] D. Lucke, C. Constantinescu, and E. 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.

[18] ——, “Fabrikdatenmodell für kontextbezogene Anwendungen: Ein Datenmodell für kontextbezogene Fabrikanwendungen in der Smart Factory,” *wt Werkstattstechnik online*, vol. 99, no. 3, pp. 106–110, 2009.

[19] M. Wieland, D. Nicklas, and F. Leymann, “Managing Technical Processes Using Smart Workflows,” in *Towards a Service-Based Internet, First European Conference, ServiceWave 2008, Madrid, Spain, December 10-13, 2008. Proceedings*, ser. Lecture Notes in Computer Science, P. Maehoenen, K. Pohl, and T. Priol, Eds., vol. 5377. Springer Verlag, December 2008, Conference Paper, pp. 287–298.

[20] “Failure management workflow,” http://www.iaas.uni-stuttgart.de/nexus/demo/FailureManagement.bpel.

(all links have been followed on July 6th, 2010)