ARCHITECTURE FOR END USER-DRIVEN COMPOSITION OF UNDERSPECIFIED, HUMAN-CENTRIC BUSINESS PROCESSES

Todor Stoitsev, Stefan Scheidl
SAP Research, SAP AG, Bleichstr. 8, Darmstadt, Germany
todor.stoitsev@sap.com, stefan.scheidl@sap.com

Felix Flentge, Max Mühlhäuser
Telecooperation Group, Darmstadt University of Technology, Darmstadt, Germany
felix@tk.informatik.tu-darmstadt.de, max@informatik.tu-darmstadt.de

Keywords: Software architectures, distributed applications, end user development, computer supported cooperative work, ad-hoc workflow.

Abstract: Enterprises are constantly struggling to optimize their business processes in order to gain competitive advantage and to survive in the fast evolving global market. Often, the only ones to understand the matter and complexity of these processes are the people, who actually execute them. This raises the need for novel business process management approaches, which can enable business users to proactively express process knowledge and to participate in the business process management and design according to their actual expertise and problem solving strategies. The presented paper describes an architecture, which supports a framework for end user-driven composition and management of underspecified, human-centric business processes. The solution builds up on email-integrated task management and enables dynamic generation of decentralized-emerging process structures through web service-based activity tracking. The captured process execution examples are shared in central enterprise repositories for further adaptation and reuse. This enables "seeding, evolutionary growth, and reseeding" of user-defined, weakly-structured process models towards global best-practice definitions.

1 INTRODUCTION

A significant shift in the view on enterprise processes has been recognized in the last years. While conventional workflow solutions are well suited for static, predefined processes, the importance of unstructured, knowledge-intensive work in enterprises raises new flexibility expectations (Aalst et al., 1999; Schwarz et al., 2001). This is accompanied with the increasing demand to enable business users to develop and exchange substantial business process knowledge, which could increase the overall enterprise efficiency (Wiig, 2004). Recent analyst reports clearly revealed that the traditional enterprise process modeling perspective is being replaced by tailoring of business processes according to the individual point of view and connecting them towards the achievement of common enterprise goals (Gartner, 2006). This novel view raises new challenges for the next generation Business Process Management (BPM) by stating the fundamental need to leverage individual expertise of business users towards the definition and management of agile business processes.

In the literature End User Development (EUD) is defined as "a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact." (Lieberman et al., 2006). Within the presented paper a process model is considered as a software artifact, which can be enacted and reused to support underspecified, human-centric processes. The presented paper is motivated through the possibility to "render" appropriation of process models to end users and to "exploit the potential of opportunity-based and emergent changes" from the introduction of groupware in enterprises (Wulf & Jarke, 2004). The paper describes a generic architecture for enabling EUD of decentralized-emerging, weakly-structured process models. The

architecture is implemented in the Collaborative Task Manager (CTM) prototype. The latter is not explicitly discussed in the paper whereas references to certain functionalities are mentioned as clarifications for the presented concepts.

The paper is organized as follows. Section 2 presents related work in the area of agile process support. Section 3 provides an overview of a framework for light-weight composition and management of ad-hoc business processes, supported through the presented architecture. The architecture is described in section 4. Conclusions and future research directions are given in section 5.

2 RELATED WORK

Riss et al. (2005) discuss the challenges for the next generation BPM by proposing the reuse of emerging "task patterns" and "process patterns" as alternative to static workflows. A further task-centric approach which enables Proactive Information Delivery (PID) on tasks and instance-based task reuse is presented by Holz et al. (2005). Task centric approaches bridging routine and ad-hoc work are also known (Bernstein, 2000; Jorgensen, 2004). The above studies present comprehensive strategies for supporting knowledge work. However, they do not consider methods or architectures for embedding ad-hoc process support in the existing working environment of end users. We suggest that this issue is important as similarly to tailoring of software systems, tailoring of ad-hoc processes through the definition, adaptation and reuse of user-defined task hierarchies should be ensured through "gentle slope of complexity" (MacLean et al., 1990), where users with different business and Information Technology (IT) background can efficiently shape and exchange reusable task definitions.

Evidences from related literature show that user strategies for organizing daily activities are far from any process or case-definition context and mostly rely on common office tools such as email (Bellotti et al., 2005) or personal to-do lists (Bellotti et al. 2004). Agostini et al. (1997) cross the boundaries of the personal workspace and integrate to-do lists and email within email-based workflows. However, the authors do not discuss mechanisms for decoupling email-based workflows from the system as explicit process models, and how such models can be exchanged, adapted or reused. As end users have different level of technical expertise and attitudes towards maintaining process data, we suggest that it is important to consider possibilities for "seeding, evolutionary growth, and reseeding (SER)" (Fisher et al., 2004) of user-defined task structures for their refinement and complementation.

This paper presents a distributed architecture for supporting enterprise-wide "programming by example" (Liebermann, 2001) of weakly-structured process models. This EUD technique supports unobtrusiveness by enabling generation of ad-hoc workflows from captured, executed activities, and yet allowing users to proactively tailor the emergent processes at use time. The system tracks user actions on tasks in users' personal to-do lists and replicates task data on central server instance. There, personal to-do lists of different process participants are integrated to end-to-end processes based on tracked email exchange for task delegation. Captured data is disseminated in different server repositories which enables multiple perspectives on processes: process, resource or user -centric.

3 FRAMEWORK OVERVIEW

The architecture described in this paper, supports a framework for light-weight composition and management of ad-hoc business processes (Stoitsev et al., 2008). This section provides a high-level overview of the framework to clarify the architectural entities described further in the paper.

Aalst et al. (1999) discuss business process flexibility by suggesting three basic dimensions of a workflow – "case", "process" and "resource" dimension. Human activities thereby comprise cases, which are handled through corresponding processes, executed through a sequence of tasks by using appropriate resources. Unpredictability of ad-hoc activities implies dynamic adaptations of tasks and resources and deviations in case handling. The used framework considers these dimensions through its entities: tasks, artifacts, human actors and Task Patterns (TP) (see also Riss et al., 2005).

A framework overview is shown in Figure 1. A case repository with reusable task structures, generated through user activities, is shown on the left hand side – case information is available in individual workspaces of various users and is additionally replicated on central enterprise repositories as described further in this section. Tasks A and B are part of a concrete collaborative process. Tasks C and D represent process fragments, elaborated by domain experts and may be part of running processes or explicit case descriptions. On the right hand side, the provision of resources by external artifact managers is shown. Different type

of artifacts and their association to tasks is discussed further in this section.

While user U_1 interacts with a hierarchical task list (tasks A … $A_{2.1}$ depicted through gray ellipses) in the local workspace, the system uses web services to track and replicate all task structure and context information on three central enterprise repositories: (i) task tracking repository – holding task structure and context information associated to task objects (e.g. subject, description status etc.); (ii) artifact repository – providing central storage for task attachments (artifacts); (iii) user repository – storing information about the involved persons (human actors) that can be provided as expertise recommendation if a task is reused later on.

Task delegation over email (A_1 → B) is tracked to bind the individual task hierarchies (to-do lists) to a Task Delegation Graph (TDG) (Stoitsev et al., 2008), which in Figure 1 includes only two users and is hence highly simplified. Changes on task instances in running processes are propagated with notifications iteratively throughout the TDG. This allows stakeholders to adapt their corresponding tasks according to the current situation by changing content or status information.

Tasks may contain references to Externally Managed Tasks (EMT) (B_1 → C). An EMT represents a possible execution (breakdown) of a given task. An EMT reference can be used to fetch the complete EMT task hierarchy including context information of the contained task instances from the server. An EMT can contain further EMT references. The reference chain ends with a task without further references (task D). Changes on EMT trigger notifications to the reusing task instances. Stakeholders can update to the new structure or preserve the locally used EMT copy and release the reference, which corresponds to an apply TP operation as discussed further in this section.

A task instance can contain different types of artifacts. Artifacts are files, e.g. Microsoft Word/Excel documents, archive (zip) files or even executable files, which are used or generated during task execution and are associated to tasks e.g. as attachments. Artifacts can be: (i) locally managed, non-externalized i.e. not replicated in a remote repository (black circle in $A_{2.1}$ in the client workspace) – used to store confidential resources; (iii) Externalized Artifact (EA) (white circle in $A_{2.1}$ represents local EA reference to an EA in the artifact repository – gray circle with a black outline). EAs are implicitly replicated and matched based on binary checksums on the server and allow detection of common tasks based on the usage of similar resources (e.g. C_2 and D use the same EA), which assists towards document-based PID (Holz et al., 2007); (iii) Externally Managed Artifact (EMA) (white circle with a black dotted outline in $D_{2.2}$ represents an EMA reference to an EMA in the artifact repository - gray circle with a black dotted outline). EMAs are notifications-enabled and allow

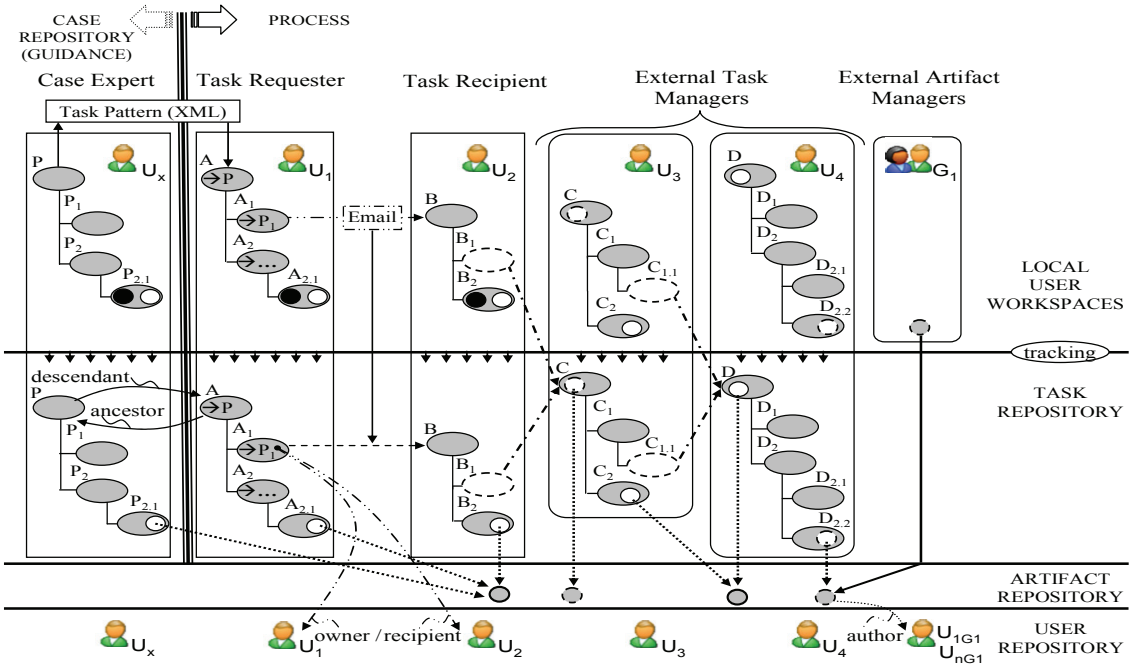

Figure 1: Framework overview.

dynamic synchronization of artifacts used in personal tasks with changes of domain experts. As an extension to the original framework we suggest storing data for EMA author in the user repository.

Task-related expertise is stored through basic user information - user name and id (email address). Two roles are supported: (i) owner – pointing at the person, whose individual to-do list contains the task (e.g. U_1); (ii) recipient(s) – pointing at the person(s), who have received a delegated task in a collaborative process (e.g. U_2).

The reuse of emerging task hierarchies is enabled through TPs. A TP contains a task with its complete sub task hierarchy and context information of all contained task instances, including artifacts and human actors. A TP can be exported from an arbitrary task in the to-do list in the local workspace or a complete TDG under a given task can be exported from the task tracking repository. TP can be stored to local or remote Task Pattern Repositories (TPR). Local TPR are XML files (Stotisev et al., 2008). Remote TPR store global best-practices (TPs) and are extensions to the original framework, enabling activation of end-to-end process examples throughout the enterprise without exchanging TP XML files but through referring to centrally stored TPs. One implicit central TPR is the task tracking repository.

When a TDG is extracted, personal task hierarchies of different process participants are saved as separate TPs and requester tasks receive EMT references to the corresponding recipient TPs. The recipient tasks are generally considered as suggested TPs for further handling of the requester tasks. Weakly-structured process models hence emerge as TPs (process fragments) which are interlinked based on suggestions. The collaborative flow is defined through suggested recipients in tasks.

Applying a TP creates the complete task structure in the local user workspace by feeding all context information and setting all references to suggested TPs (EMTs) and used artifacts. EAs and EMAs are fetched from the artifact repository and attached to the local tasks in the workspace. All tasks resulting form TP application are accordingly replicated on the task tracking repository. Users can change the prescribed collaborative flow of an enacted process example by entering different than the suggested recipients. References to suggested TPs (EMTs) can be forwarded with delegations or used by the task owner to fetch the corresponding EMT and execute the tasks themselves. This allows opportunity-based and emergent adaptations of enacted process examples. If a user applies a TP with suggested TP references (EMTs) from a local TPR, the locally referenced structures can be exported iteratively in a default, user-specific remote TPR so that they can be reused by other users.

To allow tracing of deviating cases, ancestor/descendant relationships are maintained. These are set iteratively in task hierarchies, resulting from TP application ($\rightarrow$P in A, $\rightarrow P_1$ in A_1 etc.). Copy/paste operations of task fragments executed during TP editing in a visual environment also produce such references. These references allow comparison of resulting with originating entities (ancestor – descendant) and comparison of resulting entities with same origin (descendant – descendant). As each task instance preserves its ancestor reference, fine-grained, instance-based traceability of task reuse is enabled.

4 ARCHITECTURE

This section presents the architecture supporting the above framework. The term “office applications” used in the following is a conceptual term and does not explicitly refer to Microsoft Office or imply the features offered by this environment. The presented architecture is implemented in the CTM prototype, which is integrated in Microsoft Outlook (OL) by exploiting the fact that tasks and email are offered in the same office application.

The system has a three-tier architecture consisting of client, server and persistence layers as shown in Figure 2. The client layer contains the logic for the personal task management. The server layer comprises the components, providing the tracking functionality, the overall repository access and data retrieval, and the business logic (e.g. notifications handling). The persistence layer encompasses the runtime data storage for task tracking and the user, artifact and TP repositories.

4.1 Client Layer

The client-side components are integrated in the office applications environment of the end user. The focus is set on the email client, which may contain also certain task management and calendar functionality.

4.1.1 Office Applications Integration Layer

The task management system is coupled to the office applications over an Office Applications Integration Layer (OAIL). The latter enables the usage of email

for the purposes of the task management system, by serving as a proxy and enabling email pre-formatting of outgoing emails and appropriate handling of incoming emails with task-related content. If task management functionality is included to some extent in the office applications environment, the integration layer should preferably make use of it by enabling tracking of task-related operations (e.g. edit, create, delete) through web services. The integration layer should also provide functionality to embed the user interface of the task management system in the office applications environment. Concretely the CTM front-end is delivered as an OL Add-In, where the OAIL comprises classes with proprietary extensions of OL mail and task items, adding custom properties and a set of event handlers.

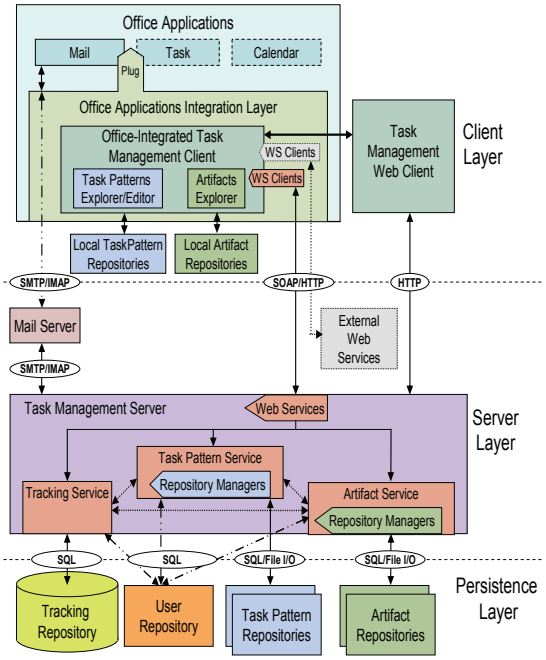

Figure 2: System architecture.

4.1.2 Office-Integrated Task Management Client (OITMC)

The Office-Integrated Task Management Client (OITMC) holds the complete presentation logic for the task management system within the selected office application (i.e. the email client). The OITMC also contains the composition environment for the personal to-do lists. Additional extended functionality for the inspection and adaptation of existing task structures is provided through a Task Pattern Explorer/Editor (TPE). It enables editing of local TPs (exported to XML files) as well as the complete functionality for creating, retrieving and editing remote TPRs and TP instances. Artifact handling is provided in an Artifact Explorer (AE) component. It enables users to add EMAs and explore their version and history and to further explore EAs and their referencing tasks through querying data from the server.

Important parts of the OITMC are the task management Web Service (WS) clients. These are responsible for tracking of task related actions on the task management server and executing updates and queries on the remote repositories. Additional WS clients can be plugged in the OITMC to execute task-related operations on external systems e.g. to trigger transactions in a workflow or Enterprise Resource Planning (ERP) system. A simple approach for binding such external WS in tasks is e.g. through executable attachments (artifacts).

4.1.3 Local Task Pattern Repositories

Although reuse of task structures should be utilized on enterprise basis, some task structures can still be stored locally. On the one hand local TPs can hold personal best-practices in a private, non-distributed manner. On the other hand, a local TPR may be used by caching mechanisms, included in the OITMC, to store local copies of once retrieved remote TPs in order to avoid communication overhead for repeated TP retrieval. Currently local TPRs are XML files, containing one or more TP definitions as described by Stoitsev et al. (2008) in an additional repository element.

4.1.4 Local Artifact Repositories

Artifacts are generally managed in globally accessible, remote Artifact Repositories (ARs). At the same time, artifacts can be stored also within local AR to ensure that sensitive documents will not be implicitly externalized throughout the enterprise - such are the locally managed, non-externalized artifacts. Local ARs can be organized as file system folders. Additionally, the OITMC can include caching mechanisms for storing once retrieved remote artifacts in a local AR, from where these will be retrieved on subsequent requests.

4.1.5 Task Management Web Client (TMWC)

The TMWC provides overview of the evolving task structures beyond the boundaries of the personal workspace. It allows inspection of the complete TDGs, where all attributes of the task nodes can be viewed – status, percent complete, due date etc. as

well as artifact lists (attachments). Detailed overview of the dialog flow for task delegation is enabled in the TMWC. The overall purpose of this component is to provide system independent environment for process navigation, where advanced users with higher process expertise can inspect work distribution and identify potential bottlenecks and optimization possibilities. The TMWC in CTM can be shown through "Process Info" links in tasks and emails in the OITMC. Although in CTM the TMWC currently provides only an overview of the emerging processes, in advanced implementations this component may include functionality for dynamic task changes in the displayed, shared-accessible task hierarchies (cf. Bernstein, 2000). These should be accordingly reflected in the OITMC and supplied with notifications.

4.2 Server Layer

The server layer encompasses two major components – the mail server and the task management server. External WS e.g. from ERP systems (cf. 4.1.2) can also be considered in this layer. The different components from the server layer do not necessarily reside on the same host.

4.2.1 Mail Server

The mail server is implicitly included in the architecture as the exchange of tasks is realized over email, i.e. this component is used by the office applications email client and could be e.g. a Microsoft Exchange server. Furthermore, email can be used for the propagation of notifications from the back-end to the clients. Therefore a bidirectional relation from the task management server to the mail server is depicted in Figure 2. However the presented architecture does not rely exclusively on "computational email" as known email-based workflows (Agostini et al., 1997) but utilizes WS to increase performance and extensibility.

4.2.2 Task Management Server

The task management server application provides all basic services for the system and realizes the connection of the clients to the remote repositories.

The Tracking Service updates the persistent state of tasks on the server when these are created or updated on the client. The service handles additionally the collaborative flow. All task-related email exchange is stored in dialog instances, mapped to the appropriate requester and recipient tasks. All messages are available in the tracking repository with text and attachments. The attachments are replicated to a remote AR. The tracking service feeds also owner and recipient data in the user repository to dynamically fill the expertise database. Notifications on task instance changes (cancel, complete, delete request) are also propagated though the tracking service to allow stakeholder adaptations. The support of offline system usage (personal task adaptations in a disconnected client) is a complex issue and is not discussed explicitly in the presented paper. This mode is generally supported through tracking buffers on client and server side. On client side such is a local tracking record file. On server side, unconsumed notification events are kept in a DB table. Switching in online mode triggers evaluations of the client and server records, aiming to avoid collisions on task actualization e.g. increase of percent complete for a delegated task cancelled by the requester. The compensation mechanisms and collision prevention are not final and still under research. However, synchronous tracking support seems currently sufficient as CTM test use takes place in offices, where users are online.

The Task Pattern Service updates remote TPs, enables search in the remote TPRs and delivers remote TP structures to the clients. It further provides functionality for delivering notifications on reused, suggested TPs (EMTs). Multiple Repository Managers can realize the connection to multiple TPRs of the same or different kind (e.g. database or file system based). User data is fed to the user repository analogously to the task tracking.

The Artifact Service provides functionality for the replication of EAs and for the management of EMAs on remote ARs. This includes all update, search and retrieval functionalities. The service can handle different repository types (e.g. database or file system based) through appropriate artifact Repository Managers. Author information of EMAs is stored in the user repository.

The Tracking Service and the TP Service communicate with each other to enable setting and retrieval of ancestor/descendant relationships between tracked tasks and explicit TP instances. Both services communicate with the Artifact Service, to maintain associations of artifacts within active (tracked) tasks and TPs. All services have access to the user repository to feed expertise (task owner/recipient) and (EMA) author information. In CTM all services are based on the Simple Object Access Protocol (SOAP) and integrated into a java application server.

4.3 Persistence Layer

The persistence layer comprises the task tracking repository and the user, artifact and TP repositories. Components used to store different types of data can reside on physically different hosts. This would require additional repository mapping functionality in the respective services for relating the different entities – tasks, users, artifacts and TP. Currently CTM implements the different repositories as tables in a single DB for simplicity reasons. Thereby entities from the different repositories are associated through foreign-key relations.

4.3.1 Tracking Repository

This repository stores the data, generated through tracking of client-side user operations, on the server. The latest state of all user tasks from the personal workspaces is replicated in this repository. The tracking repository provides the input for the web-based overview of a TDG (generated enterprise process flow) in the TMWC and stores also all task-related dialogs, generated through task delegation.

4.3.2 User Repository

The user repository stores human actor information. Such can be added in the following ways: (i) through tracking of evolving tasks in running processes i.e. task creation feeds owner data, task delegation delivers recipient information; (ii) saving of a remote TP, containing human actor information (owner, suggested recipients); (iii) adding and editing of EMA adds author information. This approach enables user-based navigation throughout document and task repositories and tracing what process fragments and resources are accessed or submitted by a given user. No domain-specific roles are introduced currently as this could harm the generic character of the approach and its ability to support various user activities in different enterprises from different business domains.

4.3.3 Remote Task Pattern Repositories

A remote TPR holds reusable task structures. More than one repository can be maintained within the system. One implicit TPR is the tracking repository. However, while task structures in the tracking repository may be changed frequently when a user updates their personal to-do list in the OITMC, the idea of a remote, central TPR is to keep reusable process fragments in a more consistent manner and to provide global best-practices. As a TP can represent an EMT, TP changes must be supported with appropriate notifications handling to allow stakeholder adaptations. Remote TPRs can be implemented using DB tables. Access and management of different repository types should be provided in a unified manner by the Task Pattern Service. Currently CTM supports only one remote, DB-based TPR in parallel to the tracking repository.

4.3.4 Remote Artifact Repositories

A remote AR holds globally reusable artifacts. One or more ARs of the same or different kind, e.g. DB or file system based, should be managed in a unified manner through the Artifact Service. Currently CTM uses a single combined AR – a DB table with artifact context information as name, checksum, version etc. and links to actual artifact content (files) on the server file system. All remote artifacts – EAs and EMAs are currently kept in the same repository (DB table) where EMAs have additional version attributes. This can enable conversion of EA to EMA through extending the EA attributes' set.

5 CONCLUSIONS & FUTURE WORK

The paper presents an architecture which implements an integrated approach, leveraging user experience with standard office tools for collaboration and task management towards the definition of weakly-structured process models by end users. The underlying approach aims at ensuring a "gentle slope of complexity" for users engaging in process tailoring activities. To achieve that, the architecture enables enterprise-wide "programming by example" of distributed-emergent, ad-hoc processes through web service-based activity tracking. An important aspect is the provided possibility for "seeding, evolutionary growth, and reseeding (SER)" of weakly-structured process models in shared enterprise repositories. This enables refinement of generic processes towards global best-practice definitions in organizations. The presented architecture enables SER by distributing and interconnecting data in various repositories – tracking, user, artifact and TP repositories. This enables different perspectives on processes: (i) process perspective – describing the end-to-end process flow where tasks contain all relevant artifact and human actors' information and additionally allow tracing of evolutionary task reuse (ancestor/descendant hierarchies); (ii) resource

perspective - enabling detection of similar tasks based on similar resources, i.e. exploration of EA and EMA references in tasks; (iii) human actor perspective – enabling tracing of user expertise and contributions based on associated tasks and resources, i.e. task owner/recipient, EMA author relationships. These extensive data mining capabilities and the enabling of end users to model and enact end-to-end process execution examples by combining existent and new TP, user and artifact data go beyond known email-based and evolutionary workflows.

A long term CTM evaluation is planned which will allow scalability assessments for the entities, managed through this architecture. This evaluation should further reveal possibilities for automatic detection of rigidly recurring process facets. We further plan to investigate the mapping of user-defined process descriptions to formal process modeling notations towards automation of rigidly recurring process fragments.

ACKNOWLEDGEMENTS

The work, this paper is based on, was supported financially by the German Federal Ministry of Education and Research (project EUDISMES, number 01 IS E03 C).

REFERENCES

Aalst, W.M.P. v. d., Basten, T., Verbeek, H. M. W., Verkoulen, P. A. C., Verhoeve, M., 1999. Adaptive Workflow: On the interplay between flexibility and support. In *ICEIS'99*, Setubal, Portugal, pp. 353-360.

Agostini, A., De Michelis, G., 1997. Rethinking CSCW systems: the architecture of Milano. In *ECSCW'97*, Springer, pp. 33-48

Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D. G., Ducheneaut, N., 2004. What a To-Do: Studies of Task Management towards the Design of a Personal Task List Manager. In *CHI'04*, ACM Press, pp. 735-742.

Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., Grinter, R., 2005. Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload. *Human-Computer Interaction*, Lawrence Erlbaum Associates, Vol. 20 (2005), pp. 89-138.

Bernstein, A., 2000. How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier. In *CSCW'00*, ACM Press, pp. 279-288.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A., Mehanjiev, N., 2004. Meta-Design: A Manifesto for End-User Development. *Communication of the ACM*, Vol. 47, No. 9 (Sep. 2004).

Gartner Research, 2006. *Person-to-Process Interaction Emerges as the 'Process of Me'*, Gartner Inc.

Holz, H., Maus, H., Bernardi, A., Rostanin, O., 2005. From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management. *Journal of Universal Knowledge Management*, Vol. 0, No. 2 (Nov. 2005), pp. 101-127.

Jorgensen, H. D., 2004. *Interactive Process Models*. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Lieberman, H., 2001. *Your Wish is My Command: Programming by Example*, Morgan Kaufmann.

Lieberman, H., Paterno, F., Wulf, V., 2006. *End-User Development,* Springer.

MacLean, A., Carter, K., Lövstrand, L., Moran, T, 1990. User-tailorable systems: pressing the issues with buttons. In *CHI'90*, ACM Press, pp. 175-182.

Riss, U., Rickayzen, A., Maus, H., van der Aalst, W.M.P., 2005. Challenges for Business Process and Task Management. *Journal of Universal Knowledge Management*, Vol. 0, No. 2 (Nov. 2005), pp. 77-100.

Schwarz, S., Abecker, A., Maus, H., Sintek, M., 2001. Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse. In *WM'01, 1st Conference for Professional Knowledge Management*, Baden-Baden, Germany.

Stoitsev, T., Scheidl, S., Spahn, M, 2008. A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes, *LNCS 4849*, Springer, pp. 213–226.

Wiig, K. M., 2004. *People-focused knowledge management: how effective decision making leads to corporate success*, Elsevier Butterworth–Heinemann.

Wulf, V., Jarke, M, 2004. The Economics of End-User Development. *Communications of the ACM*, Vol. 47, N. 9 (Sep. 2004).