

From Personal Task Management to End-User Driven Business Process Modeling

Todor Stoitsev¹, Stefan Scheidl¹, Felix Flentge², and Max Mühlhäuser²

¹ SAP Research, SAP AG, Bleichstr. 8, 64283 Darmstadt, Germany

² Telecooperation Group, Darmstadt University of Technology, 64283 Darmstadt, Germany
{todor.stoitsev, stefan.scheidl}@sap.com,
felix@tk.informatik.tu-darmstadt.de,
max@informatik.tu-darmstadt.de

Abstract. The need to involve business users in process modeling is largely perceived in the context of Business Process Management systems. This can facilitate the elaboration of consistent process models which are better turned to users' needs and organizational changes. Despite the variety of tools and notations, process modeling remains hardly accessible for business users, who lack advanced technical skills. This paper presents an integrated approach for end-user driven business process modeling which uses web service based activity tracking to generate weakly-structured process models by capturing data on personal task management. These models can be adapted and reused for ad-hoc process support or exported to formal workflows by delivering the business knowledge to process designers and software developers. Interconnection of ad-hoc and formal workflows results in enhanced process flexibility and allows complementation of formal workflows through deviations at runtime. The approach is validated through the Collaborative Task Management (CTM) prototype.

Keywords: business process modeling, process-enhanced groupware, end-user development, agile workflow, computer supported cooperative work.

1 Introduction

Effective Business Process Management (BPM) can bring competitive advantages to enterprises in the fast evolving global market. Often, the only ones to understand the matter and complexity of business processes are the end users of enterprise software, who execute them on a daily basis. The need to use the detailed process knowledge of end users during the implementation of BPM solutions in enterprises is clearly perceived and emerges in analyst reports e.g. as the need for “increased business collaboration in process modeling” [9]. It calls for bridging the process understanding of all stakeholders involved in a Workflow (Wf) project - the business users and the business technology staff, i.e. process designers and developers. As a result, standardized graphical notations such as the Business Process Modeling Notation (BPMN) [18] have emerged. Visual process modeling is enabled in a variety of enhanced BPM

solutions. However, achieving process support that is better turned to users' needs and organizational changes by "letting end users do the tailoring" demands "both domain expertise and advanced skills in computer use" [17]. Therefore upfront process modeling remains inaccessible for business users, who have detailed domain knowledge but limited technical expertise.

This paper presents an integrated approach which overcomes the above limitation and enables end users to become informed participants in business process modeling. The approach is based on collaborative task management. It is implemented and validated through a process-enhanced groupware system - the Collaborative Task Manager (CTM) which provides enhanced End-User Development capabilities. End-User Development 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 artefact" [15]. In the presented paper a process model is considered as a software artifact, which can be adapted and enacted to support human-centric business processes. The major motivation behind the tool is to render appropriation of process models to the end users.

Section 2 provides an overview of related work on agile process support. In section 3 we present the approach for end-user driven process modeling. The basic components of the CTM prototype, implementing the approach, are presented in section 4. A validation of the approach based on a CTM case study is described in section 5. Section 6 provides conclusions and gives future research directions.

2 Related Work

The need to support knowledge-intensive business processes raises advanced flexibility expectations on Wf management systems [20]. Tailoring of task and process representations according to the individual point of view and interconnecting them towards the achievement of global enterprise goals emerges as a common strategy for realizing process agility. Riss et al. suggest the recognition and reuse of emerging "task patterns" and "process patterns" as alternative to static Wfs [19]. Holz et al. [11] present a further task-centric approach which enables proactive information delivery on tasks and instance-based task reuse. Ad-hoc task hierarchies are further used to bridge routine and ad-hoc work [5, 13]. The above approaches discuss agile process support but do not consider involving end users in formal process modeling and enabling process "tailoring as collaboration" [17] between business users, process designers and developers. Wf projects often suffer from inconsistencies, resulting e.g. from "projecting the sequence of an interview onto real work situations or by assuming logical dependencies which do not correspond with reality" [10]. We therefore suggest that enabling a seamless transition from underspecified to formal process definitions is important as it could enable the derivation of consistent, real-life compliant Wfs for rigidly recurring activities and shorten the Wf implementation lifecycle. This study presents an approach for involving business users in process modeling towards enhanced adaptability of BPM to users' needs and process changes.

We suggest that similarly to tailoring of software systems, process tailoring should be ensured through a "gentle slope of complexity" [16], where users with different business and Information Technology (IT) background are able to efficiently tailor

reusable process definitions. Process mining approaches are capable of generating workflows from logged data on ad-hoc collaboration or events in formal systems [1]. However they do not allow users to tailor the emergent workflows at use time. The need for user-centric approaches arises, which ensure unobtrusiveness and in the same time enable “informed participation” of end users in business process composition by fostering “social creativity” [8] and allowing domain experts to proactively drive process optimization in enterprises.

Related literature reveals 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 [4] or personal to-do lists [3]. Agostini et al. [2] cross the boundaries of the personal workspace and integrate to-do lists and email within email-based workflows. However, the authors do not discuss decoupling of Wfs from the system as explicit process models, and how such models can be exchanged, adapted and reused. As end users have different levels 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)” [8] of user-defined process models for their iterative refinement and complementation.

Similarly to email-based Wfs, we suggest involving end users in process composition by leveraging their experience with standard tools for task management (to-do lists) and collaboration (email). In this respect, a “gentle slope of complexity” [16] for process tailoring can be provided by closely integrating the process definition in the actual user working environment and unfolding emergent processes behind the scenes in an unobtrusive, implicit manner. For achieving this we propose enabling enterprise-wide, collaborative “programming by example” [14] by implicitly reconciling data on personal task management of multiple process participants to end-to-end process execution examples. In our previous work [21] we have described a framework for light-weight composition of ad-hoc business processes. It generally enables end users to create hierarchical to-do lists by breaking down tasks into sub tasks. Tasks can be delegated over email, whereby the recipients can further break down the received tasks and delegate resulting (sub)tasks to other end users. Changes of individual tasks in the personal end users’ to-do lists are tracked over web services on a central server instance where task data is replicated in a tracking repository. Tracking of email exchange for task delegation integrates the personal to-do lists of different process participants to overall Task Delegation Graphs (TDG) on the server. TDGs represent weakly-structured process models that are captured as actual process execution examples and contain all task data including artifacts (attachments) and stakeholders’ information. TDGs enable informed participation of end users in process composition by providing a workflow-like overview of evolving collaborative tasks beyond the capabilities of common email and to-do lists.

The introduced framework enables SER of weakly-structured process models through extraction, adaptation and reuse of Task Patterns (TP) [19, 21]. In the following a TP is considered as a reusable task structure, comprising one task with its sub task hierarchy and the complete context information of the contained tasks like e.g. description, used resources, involved persons etc. TPs can be enacted to create a new process instance and execute it along the provided example flow. This flow can be altered by changing suggested task delegations or reusing referenced TP hierarchies. TP adaptation and reuse can result in evolution and complementation of captured

process examples. This evolution is traced through task instance-based ancestor/descendant relationships [21]. TPs generally enable end users to establish best-practices and to trace best-practice deviations in different application cases.

In the presented paper we discuss an approach that involves end users in formal process modeling based on implicitly generated TDGs by bridging ad-hoc and formal Wf models towards increased “business collaboration in process modeling” [9].

3 Approach

The presented approach supports process formalization through transformation of user-defined TDGs to formal workflows based on the task change and evolution history. The resulting workflows are hence implicitly modeled by all process participants and can be extended by process designers or developers in a shared context, containing ad-hoc and formal process representations. This enables process “tailoring as collaboration” [17] between business users, process designers and developers. An overview of the process definition cycle is given in Figure 1.

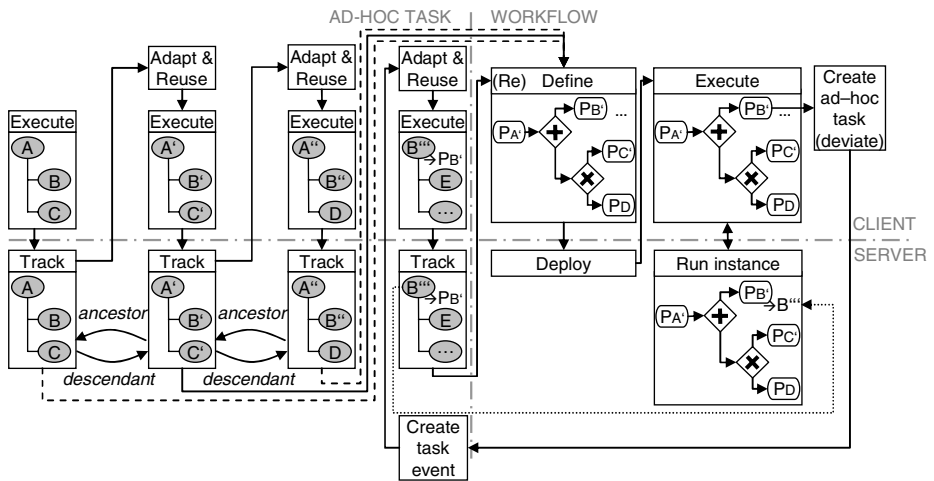


Fig. 1. Process definition cycle

A user is managing and executing tasks in a hierarchical to-do list in a task management client (root task A with sub tasks B and C in upper *CLIENT* layer). Task changes are tracked over web services to replicate task data in a tracking repository on a central server (lower hierarchy of task A in *SERVER* layer). Tracked tasks can be extracted, *adapted* and *reused* (root task A' with sub tasks B' and C'). Task instance-based *ancestor/descendant* relationships to the corresponding originating task are set iteratively for each task in the resulting hierarchy. Task reuse can result in different task variances, e.g. in task A'' the expected task C'' is replaced with task D .

When a process definition is triggered for given task (A'), the formal Wf is defined based on the complete evolution history, e.g. for task A' these are the *ancestor* and

descendant task hierarchies respectively of A and A'' , and task change history of related task instances. Task changes which alter task status, percent complete or task artifacts, are considered as task processing changes, denoting that the user is acting on a given task. Parallel flows in the resulting formal Wf are created for tasks, which have received task processing changes in parallel. For example if task B' has received a first task processing change in given time t_1 and a further task processing change at given time t_n , each task and each delegated task on the same tree level under the parent task of B' (such are C' and D) is considered parallel to B' if it has received a task processing change at a given time t_i such that $t_1 \leq t_i \leq t_n$. The period t_1 to t_n is referred to as the range of task B' .

Task ranges are a simplified way to suggest task sequencing. This is due to the fact that ad-hoc tasks can be executed without meeting any pre- or post-conditions. The resulting sequencing is hence based on suggestions and during model conversion, the user should be able to view the task change and evolution history and estimate whether the suggested flow is correct. SER can improve the accuracy of the generated workflows, i.e. if a given TP is reused multiple times and task ranges overlap in multiple executions, the tasks can be considered parallel with greater certainty. SER can enable also the modeling of alternative flows, i.e. based on substitution and cancellation of subsequent tasks in different TP application cases (in Figure 1 '+' denotes parallel and 'X' exclusive split).

The hierarchical order of tasks in TDGs is considered during model transformation by enabling different export modes for a task with subtasks: (i) as sub process, containing the sub tasks – this mode is pre-selected if a parent task contains data like e.g. attachments, detailed description etc., which is transferred to one or more of the sub tasks; (ii) as atomic task before the sub tasks' sequence – this mode is pre-selected if the parent task data is not transferred to any of the child tasks; (iii) as group element (e.g. BPMN group artifact – cf. [18]), embracing the sub tasks as logical association – this mode is pre-selected if the parent task contains only a subject.

Delegations in a TDG are considered during the model transformation as follows: (i) if a delegated task has no sub tasks on requester side it can be omitted, or preserved along with the recipient tasks in the resulting model. Omission is pre-selected as it results in model simplification when the task was fully processed by the recipients. In case of delegation to multiple recipients sub tasks of recipient tasks are handled as children of the same parent and checked for overlapping ranges (parallel execution). (ii) if a task was delegated, but the requester has added subtasks to it in their to-do list, requester and recipient tasks can be preserved as independent process nodes, or they can be merged by selecting one of them as the preferred, resulting Wf task. In the latter case requester and recipient sub tasks are handled as children of the same parent and checked for overlapping ranges.

Generated Wf tasks receive a reference to the originating ad-hoc task (P_A , P_B etc.). A Wf is deployed on the server and executed through a Wf engine. During execution, users are enabled to deviate from a formal Wf by creating an ad-hoc task for a given Wf task. This issues an event over the server, creating an ad-hoc task in the to-do list of the respective delegate by additionally transferring the Wf task information to the resulting ad-hoc task, including a reference to the ad-hoc task, used for Wf task definition (for P_B , this is B'). The recipient of the deviating ad-hoc task can adapt and reuse the original task (B') (*ancestor/descendant* references for task B'') are not

shown for simplicity). The resulting task (B'') receives a reference to the deviated Wf task ($P_{B'}$), and the latter receives a reference to the deviating ad-hoc task (B''') when it is tracked. This allows interrelation of the Wf task to ad-hoc task and vice-versa and navigation from the to-do list and TDG to the suspended Wf and from the Wf to the TDG of the ad-hoc task. The execution of the deviated Wf task can continue, i.e. if the deviation is an extension to the suspended Wf rather than an exception that requires Wf termination. While the ad-hoc task management server tracks the changes of the deviating ad-hoc task hierarchy (of B'''), the Wf server tracks the state of the deviated Wf task ($P_{B'}$ – started, suspended, ended). This allows evaluation whether a deviating ad-hoc task and the respective Wf task continue in parallel or the ad-hoc task is completed before the Wf is processed further. After the Wf has ended, the Wf model can be *redefined* by considering the deviating ad-hoc flow in addition to the original ad-hoc task hierarchies, used for Wf definition.

4 Collaborative Task Manager (CTM)

The presented approach is implemented and validated through the CTM prototype. CTM is a task management tool with enhanced End-User Development capabilities and addresses two main issues: (i) light-weight composition of weakly-structured process models for ad-hoc process support; (ii) formalization of weakly-structured process models for automation of rigidly recurring processes.

4.1 Programming by Example of Weakly-Structured Process Models

In order to ensure integrated support in a common user working environment, the CTM front-end is delivered as a Microsoft Outlook (OL) add-in. CTM extends OL mail and task items and enables “programming by example” by capturing OL events and using web services to replicate task data in a tracking repository, residing in a database on the CTM server. The CTM to-do list is shown in Figure 2. Extensions to the standard OL tasks enable end users to create hierarchical to-do lists. When the end user is creating or editing a CTM task they work with the familiar OL task fields. Files can be added to CTM tasks as common OL attachments.

A CTM task is delegated through a “Request” email message, which recipients can “Accept”, “Decline” (similarly to meeting requests in OL) or “Negotiate”. The latter action allows iterative clarifications on tasks. When a request is accepted, and later on completed by a recipient, they issue a “Declare Complete” message, to which the requester can respond with “Approve Completion” or “Decline Completion”. The actual discourse takes place in the email text, independently from the given message type. This allows open-ended collaboration and prevents from submitting user behavior to strict speech-act rules, which is a known limitation in speech-acts adoption [7]. All task-related email exchange is associated to a task dialog and stored on the server. Dialogs can be inspected through a process tree web overview, where the nodes provide links to task and email information including text and attachments.

CTM tracks the task-related email exchange and integrates the to-do lists of different process participants to a TDG [21] as shown in Figure 3, where individual tasks

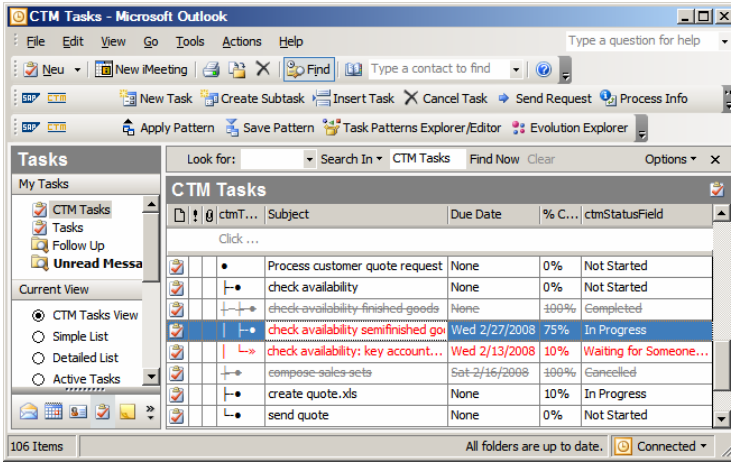


Fig. 2. CTM to-do list

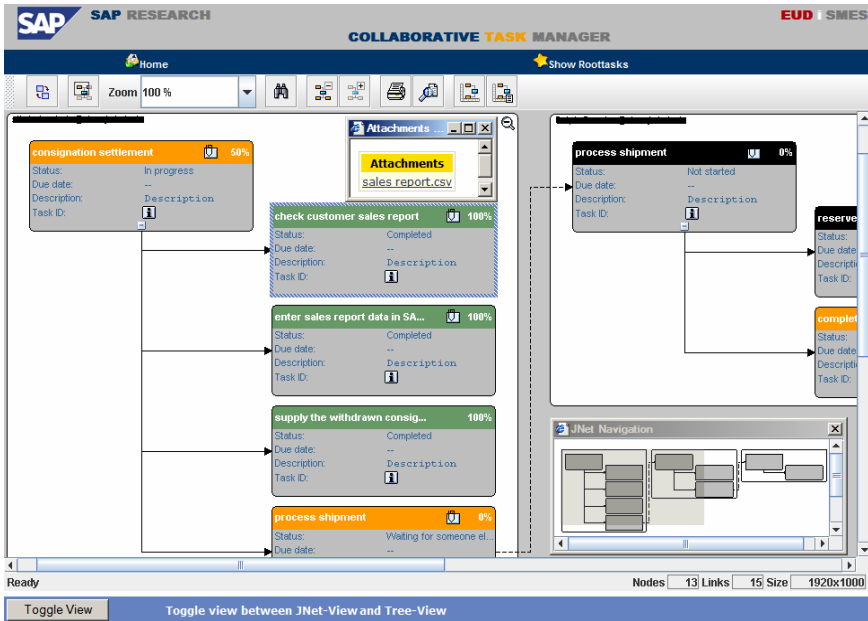


Fig. 3. Task Delegation Graph (TDG)

reside in different user containers (user data is blacked-out for privacy reasons in all figures in the paper). TDGs provide a workflow-like overview of collaborative activities where users can view status of related tasks, identify potential bottlenecks and evaluate work distribution. Currently, due date, task processing status and percent complete indications are provided. Attachments, added in OL tasks, are replicated in a central artifacts repository in a database on the CTM server, and are accessible in the

task nodes. We focus on the composition and adaptation of process models by considering business users who can share information without extensive privacy requirements. Therefore no fine-grained authorization framework is currently provided. Such needs to be considered for CTM usage in a larger enterprise context.

4.2 SER of Weakly-Structured Process Models

CTM enables export of a local task from the personal to-do list to a single TP, and export of a complete TDG from the server to multiple TPs which represent the personal task hierarchies of different users and are interlinked through suggestions according to the delegation flow. TPs can be saved in local or remote TP repositories. A local TP repository is a XML document [21]. Remote TP repositories reside in a database on the CTM server. TPs are managed in the TP Explorer (see Figure 4), which provides rich editing and search functionality on task trees and on data in context fields on the right hand side, and allows also task search and extraction of TPs from the tracking repository. When editing the process execution examples (interlinked TPs) in this component “the user is not required to interact in the interface domain of computational abstraction, but works directly with the data that interests him or her” [15]. In that sense CTM enables editing through direct manipulation of the task fields. The “Name”, “Description” and “Suggested Execution Time” fields hold simple task information in text format and are self-explanatory. The “Owner” field recommends expertise, i.e. when a task is extracted from an executed process the owner is the person, in whose to-do list the task was residing. The field “Suggested Delegates” contains information about the persons, who have the expertise to execute a given task, i.e. upon task extraction from collaborative process the task recipients are set in this field. The “Suggested Pattern” field holds a reference to a TP which should be used for the further processing of a task. In case of TDG extraction, such references in requester tasks point at recipient tasks, used for the further task processing. The recipient tasks are themselves extracted as separate TPs. Task attachments are represented as “Artifacts”. Adding of custom artifacts in the TP Explorer replicates these to the artifacts repository.

Tps can be reused through an “Apply Pattern” operation in the to-do list. It opens the TP Explorer, where the user can search for TPs in TP repositories and in the tracking repository. Applying a TP reactivates the process example by generating the task hierarchy and filling the pre-modeled content information in the to-do list. Available delegates are suggested when delegation is initiated. Suggested TP references are also included in the resulting tasks and can be used by the person, activating the TP, to accomplish the task themselves without delegations. If a delegation is issued, the recipient task receives a reference to the suggested TP so that the recipient(s) can adapt and reuse it.

SER of TP through their iterative adaptation and reuse can result in refinement of captured process examples. CTM enables tracing of evolving TPs through task instance-based ancestor/descendant relationships [21]. Such are set iteratively between the tasks in the originating hierarchy and the corresponding tasks in the resulting hierarchy always, when a task hierarchy is reused, e.g. through copy/paste in the TP Explorer or save/apply pattern. Through navigating in evolution hierarchies, the user

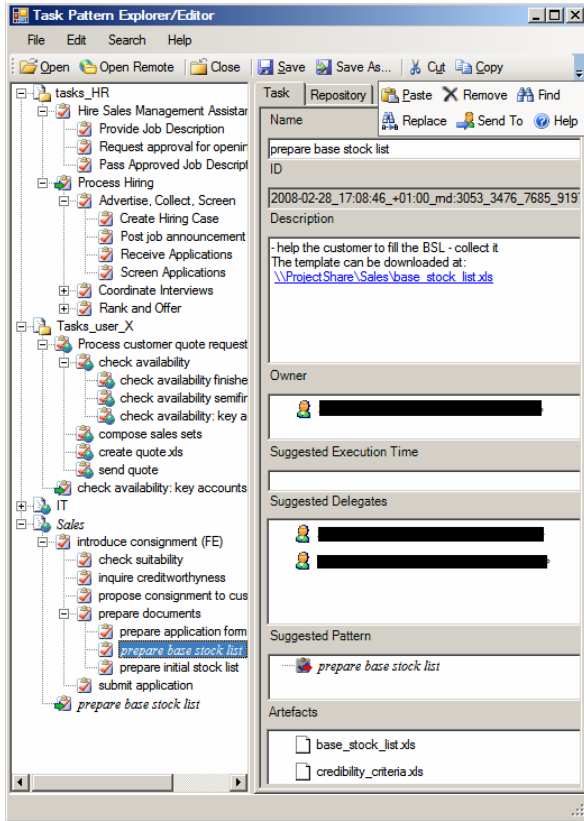


Fig. 4. Task Pattern Explorer/Editor

can view the TDGs and dialog flows of tracked ancestors/descendants. Task evolution can be viewed in an Evolution Explorer in the CTM OL add-in.

4.3 From Email and To-Do to Formal Workflows

In CTM, rigidly recurring process fragments can be detected based on the captured TP evolution resulting from SER. For process formalization CTM uses the JBoss Business Process Management (jBPM) solution [12]. jBPM Wfs are modeled in a graph-oriented, visual language – the jBPM Process Definition Language (JPDL). The Wfs can be deployed and executed on a JBoss server, where they are accessed over a web front-end. jBPM process modeling is originally performed in a JPDL designer, provided as an Eclipse plug-in. However, CTM enables transformation of user-defined TDG to formal JPDL Wfs in the CTM OL add-in, by bridging ad-hoc and formal process representations. We should stress here, that TDGs result from ad-hoc user behavior which is not constrained through formal business rules. Therefore, the process expert performing the transformation has to ensure that inconsistencies in the TDGs will not impact on the quality of the formal models. The added value from

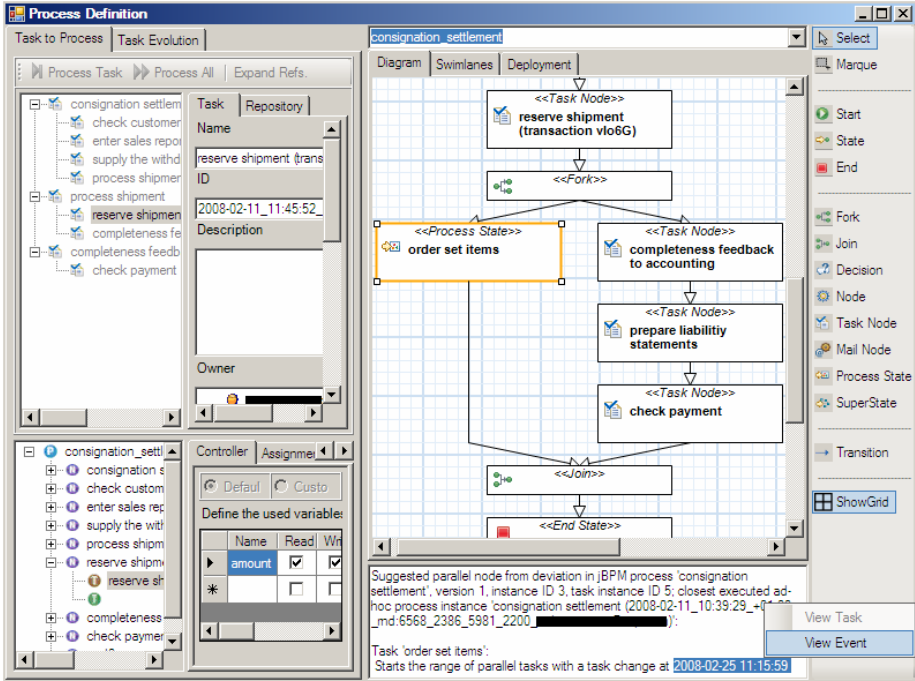


Fig. 5. CTM process definition environment

the introduced approach is that the expert is able to work with data, which was implicitly defined by the business users during their daily activities. The degree, to which the generated formal Wf models will need to be corrected or complemented, depends on how the end users are dealing with ad-hoc CTM tasks.

CTM Process Definition Environment. The CTM process definition environment is shown in Figure 5. The upper left corner contains a view, displaying the task hierarchy in the same manner as the TP Explorer. Processed tasks receive the jBPM task icon and a gray foreground. Tasks can be processed along the hierarchy through the 'Process Task' (stepwise) and 'Process All' (iteration) buttons. During task processing the appropriate export modes (cf. 2) for tasks with sub tasks and delegated tasks are provided in additional dialogs. A jBPM super state is used as a group element. The generated JPDL graph is displayed in the upper, central view in Figure 5. A toolbox on the right hand side allows advanced users to select appropriate tools and edit the model. If multiple (sub)processes are exported, the user can switch between them in the drop-down list in the upper central part of Figure 5. The tree in the lower left part contains the generated jBPM process entities (nodes and transitions). A tab control for setting their properties is provided on the right. The 'Controller' tab enables users to set parameters for task nodes. An 'Assignment' tab allows setting of jBPM task assignments such as e.g. swimlanes. The latter are automatically generated based on task owner information where each swimlane is defined through an expression 'user(email_address)' (swimlanes can be edited in a dedicated 'Swimlanes' tab - see

upper central part of Figure 5). The task properties tab control further contains a ‘Form’ tab, where advanced users can edit the xhtml code of a jBPM task’s web form. CTM automatically generates this code by additionally embedding links to the original TDG and used artifacts (available in the artifacts repository) of ad-hoc tasks, and controls for creating an ad-hoc task for deviation from a jBPM workflow and for accessing the to-do list of such a task.

A textual explanation of the relevant transformations for each task is given in the lower central part of Figure 5. It describes the overlapping ranges and refers to the appropriate change events. Task change and evolution history is provided in the ‘Task Evolution’ tab, shown in Figure 6. The task evolution tree in the upper left part contains on root level the task ancestors and their references resulting from delegations, followed by the currently processed task and task descendants if available. The TDG of tracked ancestors/descendants can be viewed through the “View in Repository” button. Task change history is displayed in the lower tree. Changes are given with their time of occurrence and changed properties on the right.

Generated jBPM Wfs can be saved as process files or deployed as fully-functional Wfs on the jBPM server. Both functionalities are provided in the ‘Deployment’ tab in the upper central part of Figure 5. Process files can be copied in the JPDL designer, where the Wfs can be extended by developers.

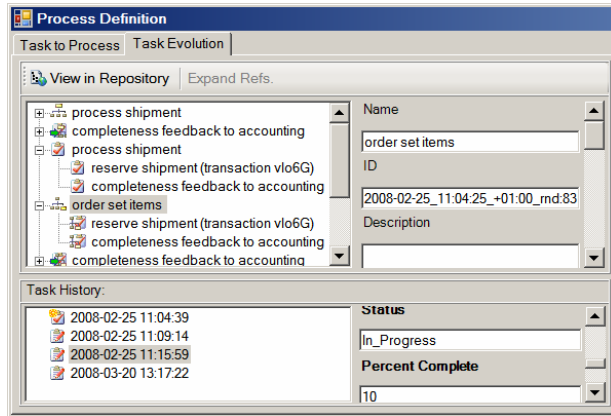


Fig. 6. Task evolution and change history

Workflow Execution and Deviations. Deployed jBPM Wfs can be executed through the jBPM engine. Process instances are started and monitored through the jBPM web front-end. Wf tasks, generated in CTM, contain in their web forms additional buttons for creating and accessing ad-hoc tasks. Creating an ad-hoc task opens a web form, where the user can provide recipient information (email address), task subject and description. When the form is submitted, this information is sent to the CTM server along with the process instance and task IDs of the deviated jBPM task. The server issues a “create task” event to the CTM client of the specified recipient, which creates a new CTM task in their to-do list. CTM uses web service events, for which each OL client subscribes on the server on OL startup. Identification for sending events to a

particular client is based on the user email address provided upon subscription. When the created ad-hoc task is tracked, the jBPM process instance and task ID are used to map the resulting ad-hoc task to the deviated Wf task on the server. The TDG of the created ad-hoc task can be opened from the Wf task's form in the jBPM front-end and vice-versa. After a process is completed, the Wf can be redefined by considering the deviating ad-hoc task hierarchy along with the original hierarchies for Wf definition.

5 Case Study

Setting and Extent of Use. The CTM case study was conducted at a manufacturing company (150 employees) and involved 6 users: COA - Chief Officer Assistant; CSO - Chief Sales Officer; SL1 & SL2 - Sales Employees; ITL - IT Department Lead; ITE - IT Employee. ITL and ITE were dealing with computers at an advanced level but did not have any programming skills and hence matched the type of end-user tailors. The other participants were typical business users. All users used OL as email client. CSO, SL1 and ITL also used OL tasks before the CTM installation. The trial was initiated with a workshop in which we gave a 1 hour presentation on the tool, followed by 30 minutes individual training of each user on the basic functionalities. Detailed user guides were provided to all participants. The jBPM export functionality was not included in the installations and manuals to preserve the focus on informal process support, addressing equally IT and business users. The trial lasted 8 weeks. Daily backups of the CTM database were scheduled and collected for evaluation each week. The evaluation concluded with a short video recording and transcription of the tool use, followed by a structured debriefing interview, in which we asked each participant to assess the basic features and rate to what extent CTM improved their ability to manage work using Likert scales and freeform explanations.

In a second iteration with SL1, SL2 and CSO we additionally performed an exercise for execution and refinement of a recurring process. The process was for settlement of consignment sales and occurred twice in the database backups. As consignment sales reports were sent in the end of each week and consignations were settled each Monday, the process seemed very appropriate for automation. We generated a jBPM Wf from a captured TDG and organized a workshop with the involved users. The workshop started with a 40 minutes tutorial on the jBPM web front-end where we explained to the users how deviations can be handled through creation of ad-hoc CTM tasks. Then we asked the users to process a weekly consignment settlement for a customer by maintaining the tasks in the jBPM Wf and deviating where needed. We used think-aloud and contextual inquiry [6] methods to track their strategies and intents. The exercises were videotaped for analysis.

Findings - Ad-Hoc Process Support. An excerpt from the case study metrics is given in Table 1. All participants reported that creating CTM tasks did not impede their work. We observed that users generally manage percent complete and status information, however not as precise estimation of work completion, but moreover *“to indicate that I'm working on it [a task] and avoid getting calls and emails from the others [sales], asking about status”* (ITE). We further encountered that users maintained attachments in CTM tasks, which was considered *“faster than email, as I only*

needed to attach the updated document and the others can pull the latest version [from the TDG]” (SL1). As CTM was used only by a small group of people, privacy issues were not raised during the trial. However ITL stated that authorization has to be considered for extended CTM use in the enterprise by providing the possibility to hide certain process fragments in black-box containers in the TDG overview. The users further considered that having *“a kind of checklist [TP] with all things I need to do and the documents I need is very useful ... especially if she [CSO] is not in the office [vacation]”* (SL2). The overall attitude was that global TP should be delivered by a (senior) domain expert, who can handle also the responsibility for providing them. Due to the restricted CTM usage, it was not possible to distribute TPs throughout the company, which prevented from developing a global strategy for TP management e.g. as alternative to text-based documents. Eventually, 2 remote TP were finally available (from ITL & CSO) whereas SL2 and ITE had developed local TPs.

Table 1. Excerpt of case study metrics

Metric	N
Created root tasks (ad-hoc processes)	8
Created tasks (overall)	46
Delegations	14
Unique attachments added	25
Attachment changes (diff. checksum, same name)	12
Percent complete changes	45
Task changes overall (only edit, no create/delete)	68
Created remote TP	2
Created local TP (files on user PCs)	4
Reused remote TP	1
Reused local TP	2

Findings - Formal Wf Definition and Refinement. A captured TDG of a process for settlement of consignment sales is shown in Figure 3 (task names are freely translated by the authors from German, customer name is removed for privacy reasons). SL1 receives a consignment sales report from a customer per email. The report is a CSV (Comma Separated Values) file, describing customer data, such as e.g. International Location Number (ILN), address etc., and consignment sales balance. SL1 “checks the report for consistency” as wrong input data like ILN can cause errors in the further processing. After that she “enters the sales report data in SAP R/3” system by copying the report in a special folder, from where the file is automatically read into the system. SL1 then describes the “supply for the withdrawn consignment items” in R/3 by specifying e.g. type and number of items. Then she asks SL2 to “process the shipment”. SL2 “reserves the amount for shipment” in another transaction in R/3 and sends a “feedback about the completeness” of the settlement to the CSO for accounting purposes. CSO receives the feedback and later on “checks the payment” for the re-supplied goods. We generated a jBPM Wf from the captured TDG, which contained the above tasks in a strictly sequential order. We then asked SL1, SL2 and CSO to process a weekly consignment settlement for a customer by maintaining the corresponding tasks in the jBPM Wf and deviating where necessary.

After SL1 transferred the data from the customer sales report to the R/3 system, she cross-checked the resulting invoiced amount in the system with the amount in the sales report. There was a slight difference in both sums: *“Yes, sometimes the reported customer prices differ from our company prices ... this is mostly due to the different calculation of taxes as customer calculates per delivery and we per item”* (SL1). The differences were minimal and were considered insignificant: *“Well, as in this case it is usually a matter of cents ... we continue the settlement with the customer prices and ask Mrs. ... [COA] to contact the customer and request them to correct the prices for the next settlement.”* (SL1). As a result SL1 deviated from the currently started jBPM task “enter sales report data in SAP R/3”, and created a CTM task in her to-do list with the same name. She then created a sub task “cross-check invoiced amount” and to this subtask she added another subtask “ask customer for correction”, which she delegated to COA. As the process could in this case continue (with customer prices), SL1 returned back to the deviated jBPM task and completed it. She then completed the “supply for the withdrawn consignment items” task without deviations.

When SL2 started the “reserve amount for shipment” task he inspected the data about previous deliveries in R/3 and the reported amount of sold items in the customer sales report. For one of the consignment items he noticed that the reported sales exceeded the previously delivered amount: *“We ship this item per store and I assume that the customer has transferred items between their stores, without notifying us. ... I'll need to inform Mrs. ... [CSO] so that she can issue liability statements for the excess”* (SL2). SL2 considered that such inconsistencies will be propagated with the “completeness feedback” to CSO, so he entered a comment in the jBPM Wf, explaining the inconsistency. A further consignment item needed to be shipped as a set of multiple, smaller items. In the concrete case, items from the set were not delivered to the customer in the required amount and had to be re-supplied additionally: *“Sets are often requested with different content from different customers ... we have to adapt and deliver the set items on demand.”* (SL2). SL2 hence deviated from the started “reserve amount for shipment” task in the Wf and created an ad-hoc task “order set items” in his to-do list: *“This is actually the same shipment procedure as for the other items ... We just process such set item deliveries independently as a special case.”* (SL2). He then reserved the shipment of the currently handled consignment items, leaving the set items for later, and returned to the deviated jBPM task to complete it, so that CSO can handle further the consignment settlement. SL2 then started processing the order of the set items.

When handling the “completeness feedback” task in the jBPM Wf, CSO read the comment of SL2 about the inconsistency in delivered and sold consignment items: *“I need to create liability statements for that [inconsistency] so that the customer can correct the problem on their side”* (CSO). For that CSO created an ad-hoc task “prepare liability statement” in the to-do list and started preparing the document. When she was ready later on, she returned to the jBPM Wf and completed the active “completeness feedback” task. For the missing set items, she later on received a delegated CTM task “completeness feedback” from SL2, who had reserved the shipment for these items. We were not able to follow the processing of the “check of payments”

task of CSO as this required customer actions. But CSO agreed that this would end the consignment settlement process and completed the task in the Wf.

Finally, we re-generated the jBPM process model with all available data from the initial TDG and from the execution of the jBPM Wf with deviating tasks, i.e. under the supervision of SL1, SL2 and CSO, with who we discussed the export modes of ad-hoc tasks (cf. 2). The “order set items” was exported as parallel sub process whereas the other deviations were exported as sequential Wf tasks. A screenshot, showing a part of the final model is given in Figure 5. Users appreciated having the complete Wf with all possible deviations in it: *“If the reported balance is ok, I’ll just complete this task [liability statement] straight away ... but I certainly want to have it there to make sure I won’t forget it”* (CSO). Users highly appreciated the provided jBPM Wf functionality as the automated task assignment would save them the effort to distribute tasks per email as usual. They further reported that they consider the final Wf real-life compliant and will try to use it on regular basis and possibly to develop several variations for different customers.

6 Conclusions

The paper presents an integrated approach enabling informed participation of end users in business process composition by using collaborative “programming by example” based on personal task management. The approach is implemented and validated through the CTM prototype. Through a CTM case study we have shown that the presented approach is adequate and efficiently reduces the cognitive distance between work tasks and Wf modeling (End-User Development) tasks. The approach introduces several gentle slopes of complexity and provides added value on personal task management as motivation to overcome each one of them. Usage of CTM ad-hoc tasks is motivated through transparency in collaborative processes, exceeding common email and to-do list capabilities. The proactive extraction and adaptation of TPs is motivated through the ability to exchange and reuse previous experience.

The presented approach further enables transformation of implicitly generated TDGs to formal process models. The formalization benefits from multiple representations and fosters tailoring as collaboration between business users, process designers and developers by allowing the latter to work in a shared context between user-defined and formal process representations. Deviations from formal Wfs during execution are enabled with on-demand, ad-hoc task hierarchies. In the case study we have shown how such deviations enable end-user driven process model refinement.

We will continue to investigate further scenarios of CTM usage in order to enhance the ad-hoc to formal conversion capabilities, considering also possibilities for Wf extensions with automated, computational tasks.

Acknowledgments. The reported work was supported financially by the German “Federal Ministry of Education and Research” (BMBF, project EUDISMES, number 01 IS E03 C). We thank to all participants in our user studies for their cooperation.

References

1. van der Aalst, W., Weijters, A.: Process mining: a research agenda. *Computers in Industry*, vol. 53. Elsevier B.V., Amsterdam (2003)
2. Agostini, A., De Michelis, G.: Rethinking CSCW systems: the architecture of Milano. In: *ECSCW 1997*, pp. 33–48. Springer, Heidelberg (1997)
3. Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D.G., Ducheneaut, N.: What a To-Do: Studies of Task Management towards the Design of a Personal Task List Manager. In: *CHI 2004*, pp. 735–742. ACM Press, New York (2004)
4. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., Grinter, R.: Quality Versus Quantity: E-Mail-Centric Task Management and Its Relation With Overload. *Human-Computer Interaction*, vol. 20, pp. 89–138. Lawrence Erlbaum Associates, Mahwah (2005)
5. Bernstein, A.: How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier. In: *CSCW 2000*, pp. 279–288. ACM Press, New York (2000)
6. Beyer, H., Holtzblatt, K.: *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco (1998)
7. Button, G.: What's Wrong With Speech-Act Theory. *Computer Supported Cooperative Work* 3(1), 39–42 (1994)
8. Fischer, G., Giacardi, E., Ye, Y., Sutcliffe, A., Mehanjiev, N.: Meta-Design: A Manifesto for End-User Development. *Communication of the ACM* 47(9) (September 2004)
9. Forrester Research. Increase Business Agility with BPM Suites. Forrester Research Inc. (2006)
10. Herrmann, T.: Evolving Workflows by User-driven Coordination. In: Reichwald, R., Schlichter, J. (eds.) *Tagungsband D-CSCW 2000*, pp. 103–114. Teubner (2000)
11. Holz, H., Maus, H., Bernardi, A., Rostanin, O.: From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management. *Journal of Universal Knowledge Management* (2), 101–127 (2005)
12. JBoss Business Process Management, jBPM, <http://docs.jboss.org/jbpm/v3/userguide/index.html>
13. Jorgensen, H.D.: *Interactive Process Models*. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway (2004)
14. Lieberman, H.: *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, San Francisco (2001)
15. Lieberman, H., Paterno, F., Wulf, V.: *End-User Development*. Springer, Heidelberg (2006)
16. MacLean, A., Carter, K., Löfstrand, L., Moran, T.: User-tailorable systems: pressing the issues with buttons. In: *Proc. CHI 1990*, pp. 175–182. ACM Press, New York (1990)
17. Mørch, A., Mehandjiev, N.: Tailoring as Collaboration: The Mediating Role of Multiple Representations and Application Units. *Computer Supported Cooperative Work* 9(1), 75–100 (2000)
18. Object Management Group, BPMN, <http://www.bpmn.org/>
19. Riss, U., Rickayzen, A., Maus, H., van der Aalst, W.: Challenges for Business Process and Task Management. *Journal of Universal Knowledge Management* (2), 77–100 (2005)
20. Schwarz, S., Abecker, A., Maus, H., Sintek, M.: Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse. In: *WM 2001, 1st Conference for Professional Knowledge Management*, Baden-Baden, Germany (2001)
21. Stoitsev, T., Scheidl, S., Spahn, M.: A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes. In: Winckler, M., Johnson, H., Palanque, P. (eds.) *TAMODIA 2007*. LNCS, vol. 4849, pp. 213–226. Springer, Heidelberg (2007)