14th ICCRTS
"C2 and Agility"

Paper ID 023

Support for agile planning & execution of coordinated actions

Topic 9: C2 Architectures and Technologies

Hans Peukert, David Lincourt, Birgit Zimmermann

Point of Contact
Hans Peukert

Chief Strategist
Global Defense & Aerospace Sector
SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
Telephone +49 6227 7 41296
E-mail Address: hans.peukert@sap.com

Abstract

Planning, training and execution of defense missions is a process with a high degree of task breakdown and collaboration. In many cases these plans exist only in textual form and have to be broken down into discrete actions with significant manual effort. This paper presents a tool called “Collaborative Task Manager” (CTM), which enables modeling, exchange, and reuse of user-defined task structures, derived from such plans in an easy-to-use manner.

Users’ working tasks can be created and managed ad-hoc in hierarchical action lists in the individual users’ workspaces. Users can further delegate tasks over email. Individual actions on tasks in the personal workspaces and task delegations are tracked on a central server, to integrate the personal task hierarchies of all process participants to end-to-end processes. This enables transparency into parties involved in a process, their tasks, and the status thereof.

Reuse of process experience is enabled through so called “task patterns”. These can be extracted from executed ad-hoc processes or created from scratch as best-practice recommendations. Task patterns, as alternative to textual mission and training plans, represent adaptable, reusable building blocks of collaborative processes. They can be used to construct task hierarchies for complex collaborative processes.

Keywords: Task management, ad-hoc workflow, end-user development, computer supported cooperative work

1 Introduction

In many areas there is an increasing amount of knowledge intensive processes. Many of these processes are performed in distributed team work. Training of defense missions for example consists of knowledge intensive, highly collaborative processes. These processes are in the focus of Computer Supported Cooperative Work (CSCW) [3]. CSCW is "computer-assisted coordinated activity carried out by groups of collaborating individuals." [3] Collaborative processes need a high degree of communication, which is supported by groupware, like email or chat. The flow of asynchronous processes can be supported via workflow management systems. But conventional workflow systems often do not provide sufficient flexibility to support ad-hoc, collaborative processes [2, 16]. In addition, in many times it is not possible to model those processes in such a way that it is possible to optimize them. This can be seen in many organizations [15].

In military mission training scenarios processes are often documented in textual from. It is a high manual effort to create text-documents that describe the processes, needed to execute a training scenario. In addition the documents describe one fixed proceeding. Deviations from the standard proceeding are often not taken into account. But mostly, in reality the execution of a process follows special rules. Persons, who have to perform certain processes, have to adapt the training documents in such a way that they fit to their current situation, which often differs from the standard proceeding described in the document. Thus the overall process is shaped through the individual actions and expertise. But up to now this individual view on processes has been largely ignored. This leads to the effect that the knowledge about how to perform a process in reality resides in the minds of those persons, who perform the processes.

To be able to deal with the complexity of these processes they have to be tailored according to the individual point of view and have then to be connected towards the achievement of common organizational goals, e.g. a complete mission training. In the area of business process management this novel view on processes is called "Process of Me" [9]. We suggest that the same need for agility and people-focused process support applies also for military processes where ad-hoc coordination and decision support is needed by people with different expertise and responsibility areas. The "Process of Me" means that end-users should be provided with adequate techniques that allow them to perform a process in the way that is needed in a specific situation by reconciling their individual perspective on the process with the organizational one. In addition end-users have to be enabled to express their knowledge about processes in order to make this knowledge accessible for other persons.

For empowering individual adaptations of process-related software artifacts this paper considers end-user development techniques. 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 artifact" [12]. This paper presents a tool called Collaborative Task Manager (CTM) that uses end-user development techniques to enable end-users to model ad-hoc, exchange, and reuse light-weight task structures and to compose organizational processes from individual, user-defined task hierarchies. Thus CTM supports the "Process of Me".

For illustrative purposes, CTM is discussed here in the context of a unit training scenario. Such scenarios are specified for example in US Army mission training plans [18, 21] These plans describe field training exercises (FTXs), e.g. the relocation of heavy maneuver forces on the battlefield, which are composed of several situational training exercises (STXs) such as the deployment of the unit. Each STX consists of a sequence of collective tasks (e.g. performing pre-deployment supply activities) that are further broken down into a hierarchy of subtasks. In particular, such tasks maybe of analytical character, retrieving decision relevant information (e.g. the identification of supply requirements), or of transactional character (e.g. the coordination of additional supplies). The efficiency of such operations can be enhanced significantly by CTM through the usage of task patterns that

- describe the break down of high-level tasks into lower level tasks,
- represent adaptable and reusable building blocks of collaborative processes,
- provide descriptive and contextual information of tasks at all levels which may include light-weight analytical and transactional applications such as widgets or interactive forms,
- can be retrieved from repositories as executable task hierarchies into a Microsoft Office application (MS Outlook).

2 Related Work

From Business Process Management perspective, one of the major possibilities to offer support for agile processes is through generation, adaptation and reuse of user-defined task hierarchies. Riss et al. [15] suggest using “task patterns” and “process patterns” as alternative to static workflows, as the latter are to rigid and do not provide adequate support for ad-hoc, knowledge-intensive processes. Ad-hoc task hierarchies are further used to bridge routine and ad-hoc work [6, 11]. Holz et al. [10] present an approach which enables proactive information delivery on tasks and instance-based task reuse. While the above approaches promise to realize adequate support for agile business processes, they miss some intrinsic aspects of user-driven process composition.

On the first place these approaches do not consider embedding the process support in the actual end users’ working environment. We suggest that this is important in order to involve seamlessly end users in process tailoring without confronting them with proprietary tools and software environments. This aspect is supported through studies on user-tailorable systems, which leverage the importance of ensuring a “gentle slope of complexity” [14] for tailoring activities. Thus we suggest providing a socio-technical environment, embedded in the current end-users’ working applications, where users with different technology background are enabled to collaboratively evolve a tailoring culture for enterprise processes. Embedded support for process composition is provided through process mining approaches, which derive process models based on data from event logs [1]. However, process mining techniques are bound to the respective systems and do not allow the users to adapt the emerging workflows during use time. The users are not able to establish a cognitive relationship between their actions in the formal system and the emerging process models.

Secondly, agile, knowledge-intensive processes are not strictly confined to procedures but are executed based on the individual knowledge and coordination of different stakeholders. Thus process knowledge can evolve over time and lead to establishing best-practices which are refined each time when a new, unexpected case occurs. To support refinement of captured process models, we suggest the “seeding, evolutionary growth, and reseeding” (SER) process model [8]. The approach proposed in this paper provides extensive support for process analysis in the context of SER through task instance-based ancestor/descendant relationships [16]. This aspect is not sufficiently addressed through other approaches for user-centric process support [5, 9, 10, 14] This presented approach enables composition of weakly-structured process models for supporting ad-hoc, human-centric processes.

3 Addressed Problem Areas

Mission training processes as described in STXs are mainly described in textual form and have to be turned into actions with significant manual effort. To reduce this effort a tool support would be desirable, that provides the possibility to create executable and reusable "task patterns" from such plans in an agile and easy-to-use and efficient manner. During execution, an end-to-end process view should bring transparency into parties involved, their tasks, and the status thereof. Task patterns should be created using a simple editor or directly out of ad-hoc process instances. This could help to perform the processes more efficient. But up to now no suited support for this scenario is available. This is caused by the fact that, according to [18], current work practices in ad-hoc business processes face several intrinsic problems, which we consider equally relevant also for agile military processes:

1. The first problem concerns **lacking transparency**: According to [5], in informal processes email is the main tool used for exchange of tasks and task-related information. The personal tasks are mainly organized in to-do lists [1]. In textual mission training plans often not even email or task list support is available. But neither textual plans nor email nor task lists provide a detailed overview of the whole process composed of several collaborative tasks of different stakeholders.
2. In addition **no structured storage and retrieval of process knowledge** is possible: The tasks related to processes are often exchanged via email. Therefore users often spent considerable effort to search for process-related data in their email folders [1]. Everyone has individual strategies for storing emails. It is therefore not possible to predict how the "sorting" practice of the users will scale over time. But it can be said, that increasing data amount increases search effort. This causes a decrease in user efficiency and makes it harder to retrieve needed information.
3. Current work practices show a **lack of exchange of process knowledge**: As described in the introduction the knowledge on how to perform a process often is implicit knowledge, remaining in the minds of those persons, who perform the processes or it is stuck in personal email and file folders. Thus the experts know what to do, but they cannot share it efficiently with their colleagues. This leads to problems when experts are not available. Then they and cannot provide support, which is problematic especially for time critical activities.
4. Often a **disjunction between best-practices and running processes** exists: Process guidelines like mission training plans are often written down as text documents (e.g. using Microsoft Word). But text documents do not provide the possibility to observe to what extent a described (best) practice is being followed, or why deviations have occurred. Thus a disjunction between best-practice documents and running processes emerges.
5. In addition an **inability to trace evolving best-practices** can be seen: Due to changing conditions best-practices often change over time, e.g. new suppliers for technical equipment emerge, new budget constraints etc. But the process information is stored in email and file folders as well as in text-based documents. This does not allow structured comparison. Thus it is not possible, to perform a reasonable evaluation, to what extent best-practices have to be adapted to new evolutions. It also cannot be evaluated, if different variations have to be managed for different application contexts. Therefore it is a high manual effort to evolve the process guidelines over time.

This paper shows a possibility to solve those problem areas within the military usage scenario of field training exercises and situational training exercises described before.

4 Collaborative Task Manager

The Collaborative Task Manager (CTM) is an email-integrated asynchronous task management tool. It follows a human-centric approach: It allows end-users to model and reuse unstructured, knowledge-intensive, collaborative processes. For that purpose CTM enables end-users with as well as without IT expertise to compose all tasks needed to perform a certain process and to monitor these tasks. As described before, many processes like mission training processes are described in the form of text based best-practices (e.g. Microsoft Word documents). Situational training exercises for example are often written down in textual form. But in reality it has to be taken into account that often ad-hoc deviations from these documents exist. How to deal with these exceptions often resides in people's minds.

CTM allows making this knowledge explicit, as it enables end-users to capture the steps of an ad-hoc process (e.g. a situational training exercise) in the form of so called *CTM tasks*. CTM tasks are MS Outlook tasks stored in a so called to-do list within Microsoft Outlook. The tasks can be created and managed ad-hoc. CTM tasks contain the full context needed to perform a certain task, as they can contain a description of how to proceed, attachments (e.g. documents or links), information about persons involved in the task etc. This is supported via a Microsoft Outlook plug-in.

More complex tasks (as described in situational training exercises) that are composed of several steps can be easily divided into subtasks. Moreover, if tasks belong to a collaborative process, they can be comfortably delegated via email. Individual tasks of all process participants can be integrated to the whole end-to-end process. This leads to a complete hierarchy of tasks and subtasks that can be processed by several persons in a collaborative way. The evolving task hierarchies reflect informal ad-hoc processes that meet the requirements of the "Process of Me" metaphor.

CTM provides extensions to Outlook mail and task functionalities that use web services to track user actions executed on CTM tasks, like task delegation or task completion. All task related email exchange is stored on a central server. The information on the central server as well as the information tracked by web services can be analyzed. Thus information on the whole process distributed over several persons can be derived from the analyses. The information is accessible to all persons involved in the process. Thus it is possible to monitor the execution of a whole process, composed of several ad-hoc tasks residing in different user workspaces. This proceeding allows defining all discrete actions that are needed to compose the tasks for STXs.

Task hierarchies (e.g. for a complete STX) can be stored as task patterns [17, 15]. Task patterns can be further adapted and reused in recurring situations. This makes it possible for end-users to formalize tasks in the way they are performed in reality, and to recognize and exchange best practices. Thus CTM allows evolving personal as well as organizational (in the case of collaborative processes) best practices. These best practices provide a guideline of how the different steps in a collaborative process should be accomplished. From end-user development perspective, through tracking of ad-hoc processes, and extraction, adaptation and reuse of task patterns CTM enables "programming by example" [12, 7] of weakly-structured processes [18]. Programming by example is an end-user development technique, which is based on capturing and re-executing user actions in a software system. A common application of this technique on personal level is e.g. macro recording in MS Office applications. CTM utilizes programming by example on organizational level to support ad-hoc processes.

In addition with CTM it is possible to find out if deviations to a best practice described by STXs exist. The deviations are stored in the form of CTM tasks and can be analyzed. This allows to adapt the best practices if necessary. The best practices for STXs reflected by task patterns can be reused in other training situations.

In summary, CTM has the following main advantages:

1. CTM facilitates reuse as well of personal as of organizational knowledge about recurring tasks as described in STXs.
2. CTM allows exchange of task-related knowledge that can be stored in a personal folder structure or on a central server.
3. CTM makes unstructured, collaborative tasks transparent.

5 Tasks and Task Patterns

Processes are composed of steps that are represented via tasks in CTM. CTM tasks are provided as extensions to MS Outlook tasks and are distributed asynchronously via email. This is due to the reason that many users know how to work with this kind of tasks. But CTM offers additional functionalities that enable hierarchical task decomposition and enhanced collaboration support. CTM tasks can be created directly in the CTM to-do list, which is an extension of the MS Outlook task list.

All CTM tasks are stored in individual to-do lists for all users. CTM enables users to create hierarchical to-do lists (shown in Figure 1). Thus the to-do lists allow users to store tasks in a tree structure by decomposing them further into subtasks. CTM enables users to insert or remove tasks and subtasks into this tree structure in an easy to use manner: when users create or edit CTM tasks, they work with familiar MS Outlook task fields.

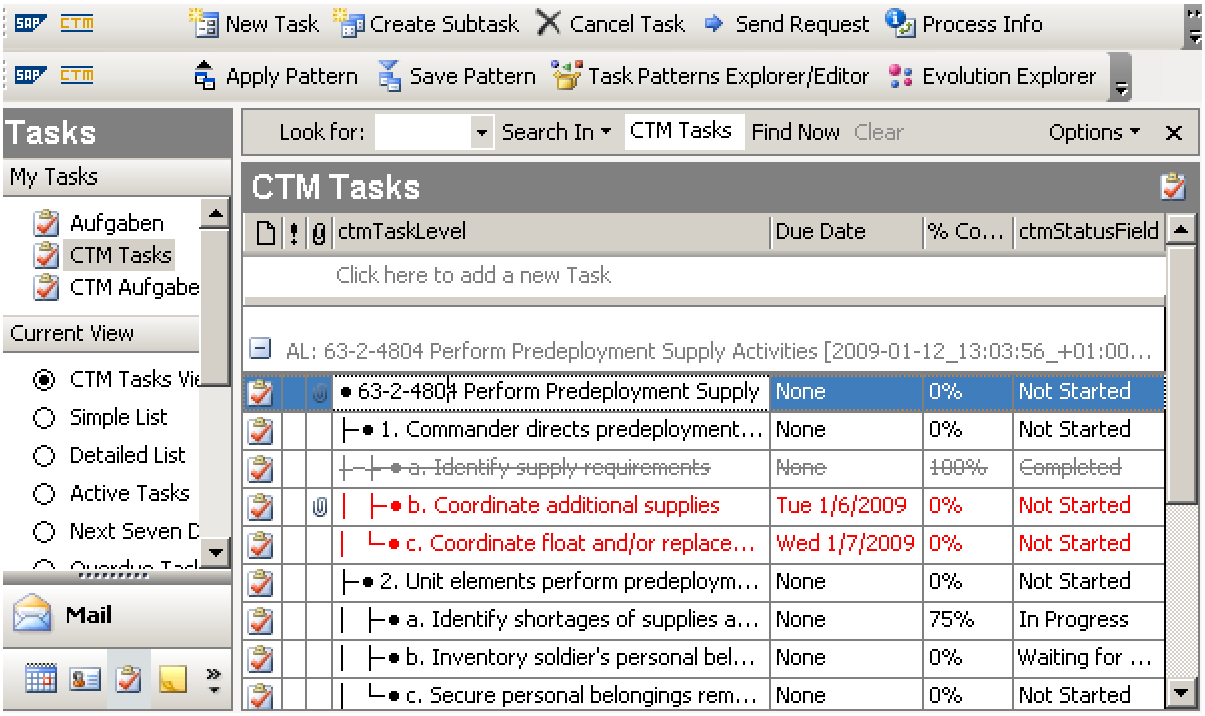

Figure 1: CTM To-Do List.

As stated before today's workflow support often does not offer the possibility to make knowledge about process execution explicit in a way that is usable for end-users. One of the main purposes of CTM is to enable these end-users to formulate and reuse their knowledge about specific tasks. This can be done by using task patterns [17, 15]. Task patterns are reusable task structures, comprising a task and its subtasks with the complete context information, associated artifacts (like documents related to the task) and human actor information (who participates in performing the task).

The Artifacts provide an excellent mean to operationalize the back office. The artifacts provide the means for the end user on how to perform the task by accessing relevant documents or systems (based on his authorization). Artifacts can be for example Adobe Interactive Forms, widgets, system access, links, electronic documents, etc. (In Figure 4 on the lower right side an artifact is shown: In this case the artifact is an Adobe Interactive Form of the US DoD DA Form 581, giving the end user the means to perform the ordering of ammunition by filling out of the interactive form and operationalizing the back office by performing the ammunition order process. When the Interactive Form is saved, it automatically syncs with the respective back office Application, e.g. SAP ERP.)

A task pattern can also contain human actor information. This means that the pattern makes a proposal who should work on the task. The actor can be a person or an organization with an associated outlook account. All tasks are sent to the recipient of the task who has to work on this task via email.

Task patterns can be saved to local and global task pattern repositories [19]. The first repository type allows local storage and reuse of process information for personal purposes. The latter repository type considers exchange of global (organizational) best practices. If a recurring task has to be done, users can search for existing task patterns defining the structure of the task. Task patterns can be created in two different ways:

1. A task pattern can be created manually from scratch in order to formulate best practices. This can be done by using the “Task Pattern Explorer / Editor” (TPEE) (see Figure 2). The TPEE offers simple yet powerful functionalities for setting up task pattern structures and organizing them in shared repositories. These task patterns are an explicit representation of personal knowledge about how to solve a specific problem. They are not derived from the to-do list.
2. A task pattern can be extracted from executed ad-hoc processes. This can be done by saving executed tasks from the to-do list in the form of a task pattern. These task patterns reflect knowledge collected during the execution of the tasks. Task patterns that have been created in this way can also be edited via the TPEE.

With TPEE various task details can be specified:

- task name,
- textual task description,
- task ownership,
- suggested task execution time,
- suggested delegates who are experts for the execution of this tasks type,
- suggested task patterns for a further decomposition of the inspected task,
- artifacts as attachments to the task which my be various files providing context information, analytical or transactional applications.

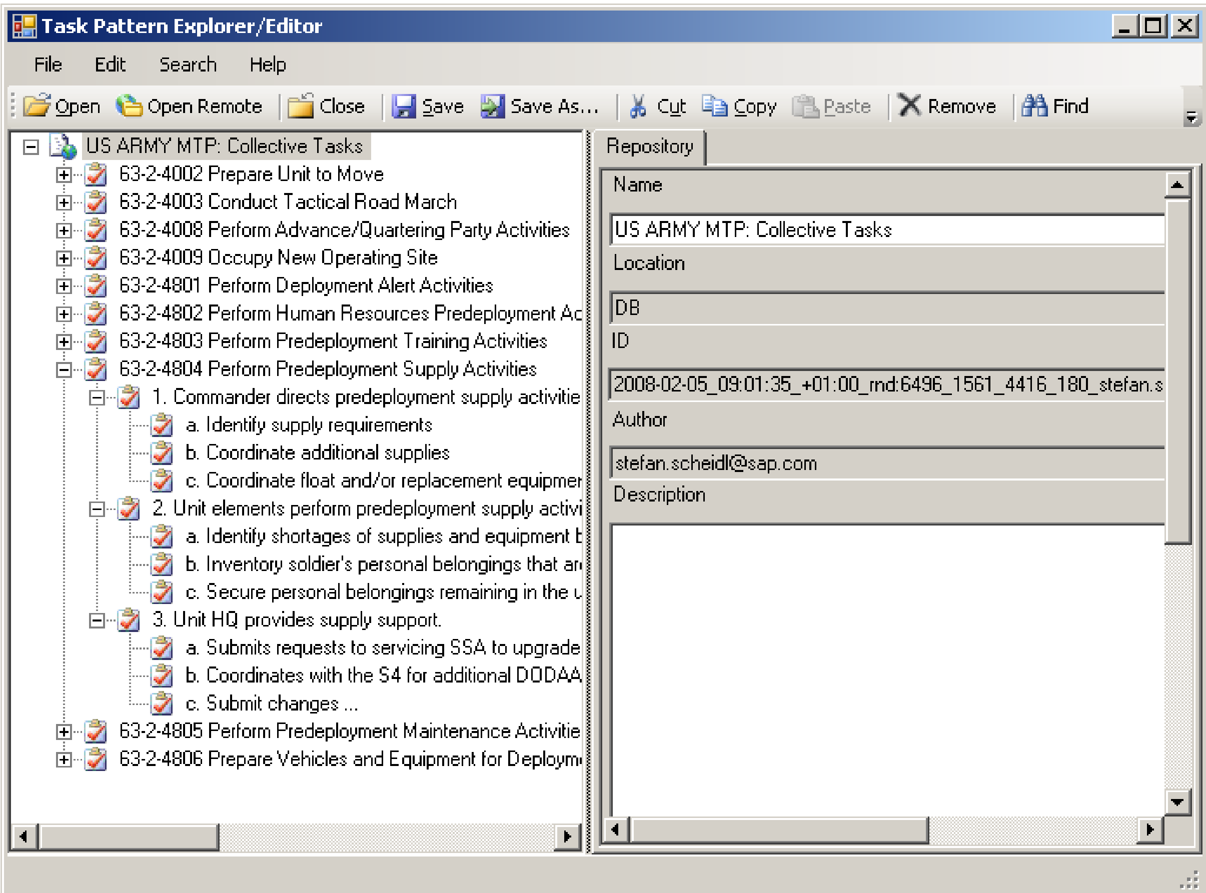

Figure 2: The Task Pattern Explorer / Editor.

Figure 2 displays a collection of mission training plans. The breakdowns of FTXs into STXs are collected in a first repository. Each of these STX references a task pattern in a second repository that describes how this STX is broken down into collective tasks. These tasks further reference patterns in a third repository that describe their break down to the lowest task levels. The patterns in the third repository are used in several STXs. In this way, the suggestion of delegates and patterns for further task refinement, as defined in a task pattern, prescribes a complete collaborative process. Task pattern referencing provides an easy way of composing a process out of several existing task patterns as reusable building blocks. The next section explains how such task patterns can be reused for training purposes.

5.1 Pattern Retrieval

The purpose of task patterns is not only to store knowledge about how to solve a specific problem (respectively task), but also to reuse this knowledge. CTM allows this by offering the possibility to take over existing task patterns during task creation. This can be done in the CTM to-do list. Taking over a task patterns during task creation can be done as follows:

The initiator of a mission training process creates a CTM task in Microsoft Outlook as a seed for a new training process. In this task, the concerned unit and specific circumstances for the process can be specified. The user may then retrieve a top level pattern from the repository, describing the FTX as a whole and how it is broken down into STXs.

To search for a suitable task pattern, the user opens the Task Pattern Explorer / Editor TPEE. This tool allows the inspection of available task patterns. A search functionality supports the discovery of suitable task patterns. Once a suitable task pattern is identified, it can be applied to the initial task. The add-in automatically creates the task substructure of the task pattern in the Outlook application as well as all its content, which is specified in the task pattern (Figure 3). This procedure allows comfortable reuse of existing task patterns for the definition of new tasks.

If a task is defined, which contains a reference to another task pattern, this task pattern is proposed as a further task refinement.

All CTM functionalities needed to define and to reuse task patterns are accessible in the Outlook application through context sensitive buttons in two command bars.

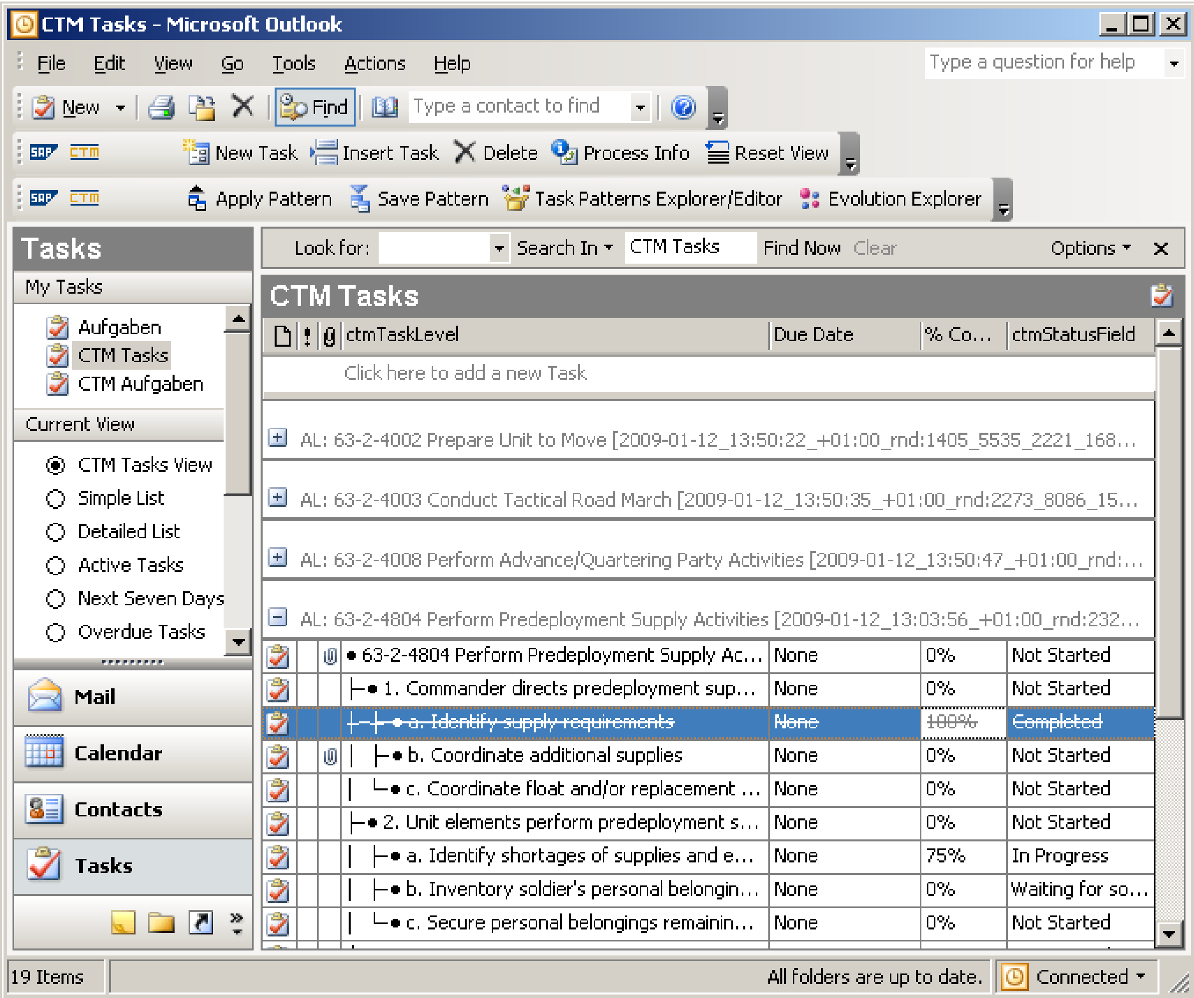

Figure 3: A task pattern instantiated in the office application.

As described before performing a process in reality often leads to deviations from best practices as described in the task patterns. CTM allows for this by offering the possibility to adapt the tasks to the current situation. If for example a person, who normally executes a certain task, is on vacation, it is possible to name another person as responsible. Or if an additional step is needed in a specific situation, a task can be added to the task hierarchy that describes this step. Thus task patterns provide guidelines based on recommendation rather than on strict prescription. Task hierarchies can be created based on existing task patterns and adapted to the current needs.

As an additional feature it is possible to analyze deviations from task patterns stored in CTM. Thus it is possible to find out if a certain task structure changes over time. If necessary the task patterns can be adapted easily to the new structure. This is described in the next section.

5.2 Pattern Evolution

Task patterns often evolve over time. This means: A task pattern is created that covers the best practice for a certain task. But after a certain time new inventions or new guidelines make changes necessary. A live-saving activity, for example, is improved over time: new methods are found, new medical equipment is available etc. In those cases the patterns have to be adapted to the new situation.

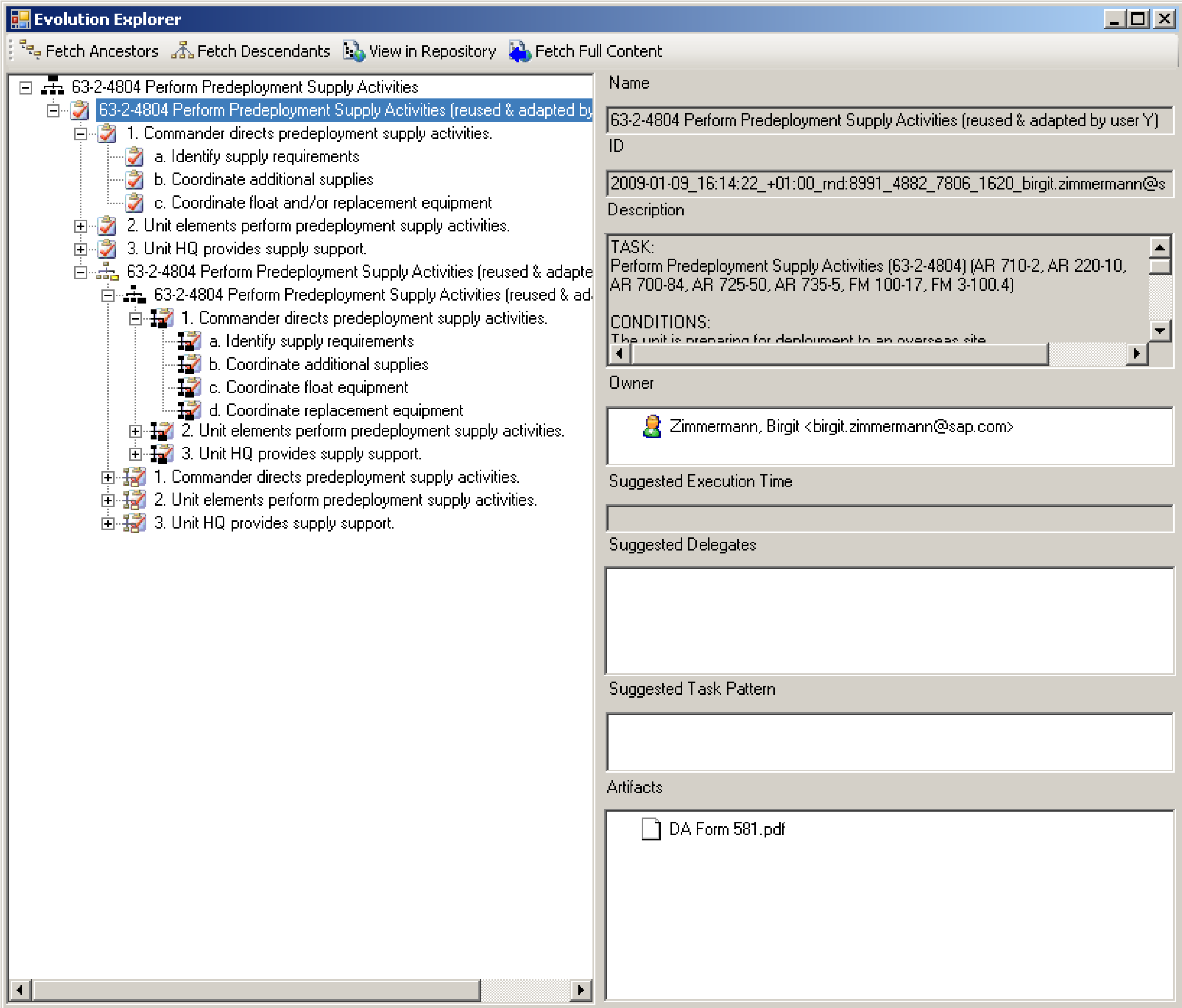

Figure 4: CTM Evolution Explorer.

CTM offers the possibility to analyze the evolution of task patterns through so called ancestor/descendant relationships [17]. If a pattern is reused in a different situation then it has to be adapted to the new situation. Let us assume that a pattern P1 is created based on a task T1. Pattern P1 is used to create a new, similar task T2. But when creating T2, the user finds that P1 does not fit all the needs given for T2. Therefore the user changes the recommendations given for T2 and adapts T2 to the given needs. The user eliminates some unnecessary steps; he changes some additional information, etc. As it seems likely that T2 is a recurring task the user creates a new task pattern P2 that is based on T2. To be able to track this evolution, relationships between task patterns are used. They can be examined via the CTM Evolution Explorer (see Figure 4).

Best-practices often have to be adapted to changing business conditions and different problem solving strategies of end users. The CTM Evolution Explorer provides functionalities to trace and compare deviating solutions for similar cases in order to enable detailed case analysis. This is done through ancestor/descendant relationships, which can be explored in the CTM Evolution Explorer (see Figure 4).

Ancestor/descendant relationships are set e.g. if a task hierarchy is transferred by copy/paste in the Task Pattern Editor. Iteratively, each task in the resulting hierarchy receives an ancestor reference to the task in the original hierarchy. In addition, if a Task Pattern from a remote

Task pattern repository is applied, the resulting task hierarchy receives all ancestor references to the applied remote task pattern. (In Figure 4 this is the node with black ancestor icon above the selected node). When a task Pattern is exported from a personal to-do list and saved to a remote task pattern repository, all tasks in the resulting task pattern hierarchy receive all related ancestor references. These references point to the corresponding tracked tasks under the extracted root (root node in Figure 4), from which the task pattern originates and which resides in the global task tracking repository. If the task, from which a task pattern TP originates, is (still) available in the tracking repository, and when this task pattern TP from a local task pattern repository is applied, references to tasks in the tracking repository are also set.

Repeated extraction and application of task patterns can produce complex ancestor/descendant hierarchies. The CTM Evolution Explorer allows inspecting all ancestors and descendants for a given task by fetching the full content (sub hierarchy and context information) for the selected nodes. As tasks on the tracking repository represent actual runtime process executions, the dialog flow of related ancestor/descendant tasks can be inspected in the original execution context.

By analyzing the information given in the evolution explorer it is possible to find out, if it is necessary to adapt certain task patterns, as they do not reflect the current best practice of performing the described tasks any longer. The evolution explorer is easy to use even for end-users without IT knowledge. Thus CTM makes it possible for end-users to define and evolve task patterns in such a way that they really fit their needs.

5.3 Benefits of Using Task Patterns

If a task is created based on a specific task pattern, the corresponding task structure of the pattern is applied to the new task in the to-do list. Thereby the complete context information (task name, description, attached information etc.) of the task and all its subtasks is also transferred. Information about suggested delegates for a given task (from previous executions) is included and automatically proposed if the task is delegated.

Additionally, suggested task pattern information (cf. Section 5) is embedded in the tasks resulting from task pattern reuse to offer recommendations for the further task decomposition. If a task has a specified suggested task pattern, a special indication of the “Apply pattern” button shows that there is a recommendation for further task decomposition (cf. Figure 5). The user can follow this recommendation to further structure the respective task. If the user on the other hand decides to delegate the task, the recipient will also receive the recommendation and can refer to it to decompose the accepted task further.

The use of task patterns allows end-users to store their knowledge about processes, like mission training processes, in an easy to use way. In addition they can comfortably access this knowledge by reusing existing patterns. If deviations from the best practice described in a pattern are needed, they can be also easily created. Thus CTM offers a comfortable possibility to store, reuse and adapt process knowledge.

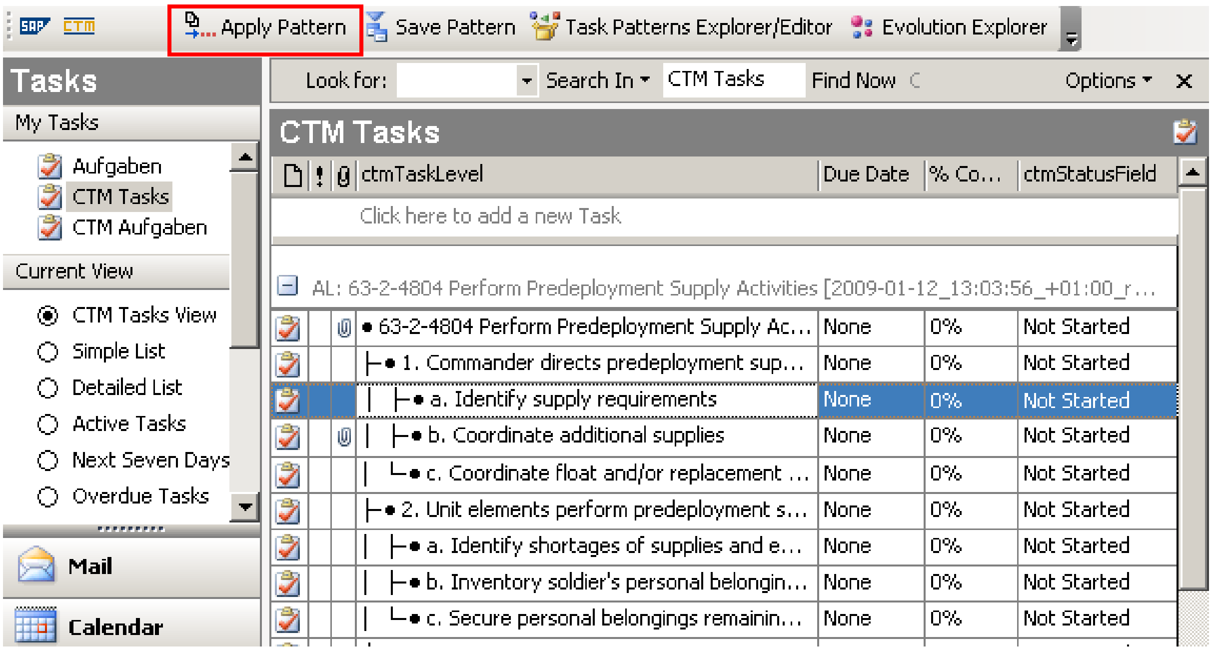

Figure 5: Recommendation to Apply a Task Pattern.

6 Collaboration

Tasks often are performed by several persons, working together in a collaborative way. CTM supports collaborative task processing and modeling of collaborative task hierarchies: It has already been mentioned that tasks at all hierarchy levels can be delegated to other people. If the user chooses to delegate a task from the to-do list which results from a task pattern, available suggested delegates defined in the task pattern are proposed as recipients of the request. The addressed delegates receive a request that can be accepted, declined or negotiated (see Figure 6).

If another recipient then the one proposed by the pattern has to work on the task, it is possible to add this recipient to the task instead of the proposed one. To add more flexibility to task patterns it is also possible to add a role concept for task delegation. Then task patterns can contain the role a person should have that has to work on a task. At the moment this has to be realized via a work-around as MS Outlook only supports personal to-do lists but no role-associated to-do list that can be shared by several users. Therefore an additional Outlook account has to be created for this role that is accessible for all persons belonging to the role. The persons then can access CTM tasks in this role-account to-do list and can delegate a task to their own account if they have to work on it.

This is useful in situations as the following on: If, for example, the process of providing transportation services has to be trained, the following tasks have to be performed: the deployment has to be supervised, the transportation support has to be planned, the relocation of subordinate elements' bases and sustainment resources has to be supervised, etc. [22] These tasks contain subtasks like railway transport, transport by train, transport with airplanes etc. Many tasks have to be performed by different persons. In the task pattern it is not useful to provide the name of each driver, pilot etc. But it is useful to provide a role like "driver of truck", "pilot for plain of type XY", etc. The task pattern then does not describe how single persons collaborate, but how persons with specific role collaborate.

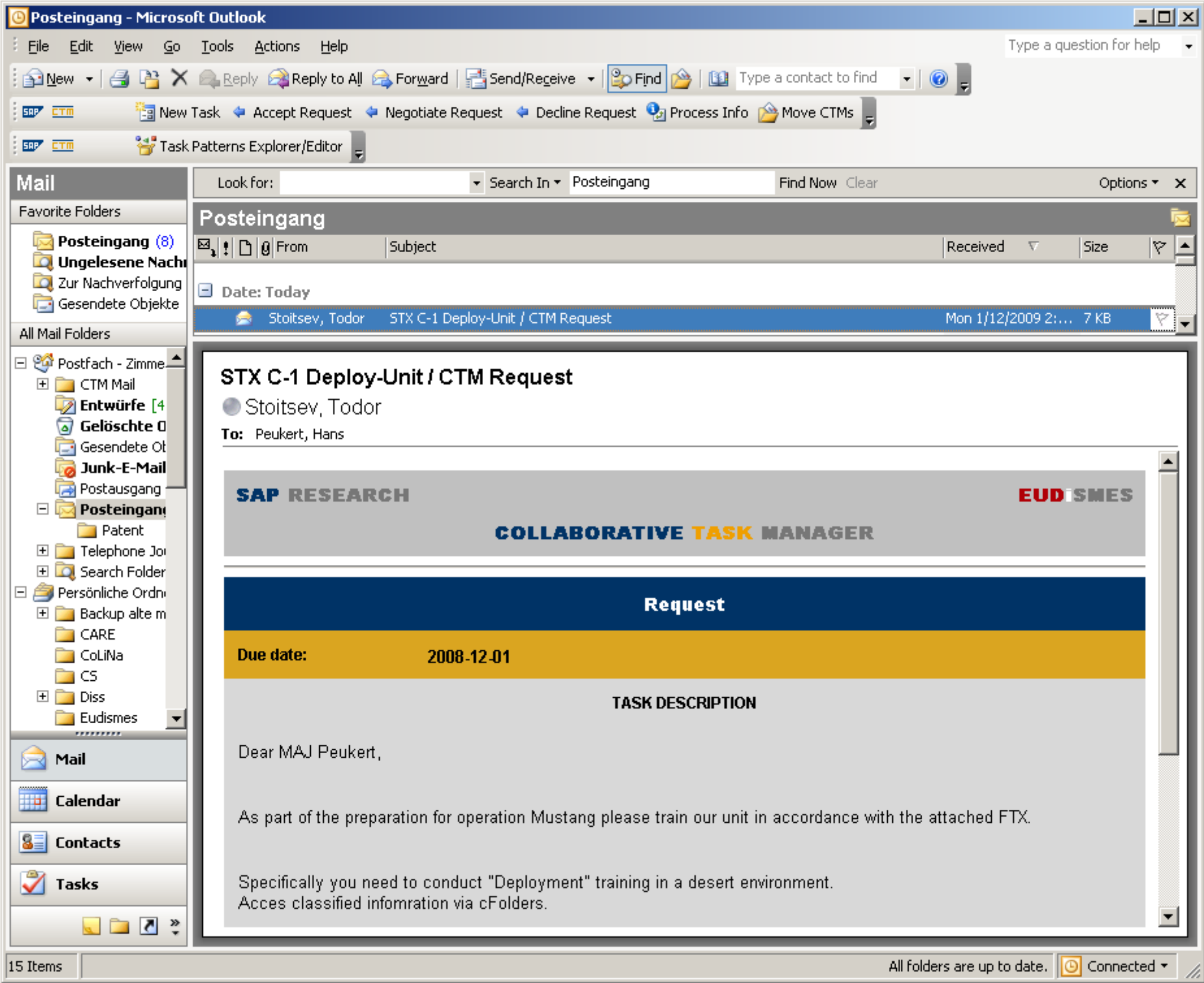

Figure 6: The recipient of a request is offered situated actions for the request.

7 Transparency

All activities on task instances in the to-do lists of different process participants are tracked to replicate the respective task structures and context information on a central server instance. Further, task delegations over email are tracked and bind the task hierarchy of the requester with the resulting task hierarchy on recipient site. Based on this tracking, task delegation graphs are constructed [17] through which all parties involved in a collaborative process can retrieve an end-to-end process overview at any time, covering information about

- the complete set of actors and their task assignments, and
- the complete set of tasks, their interrelations, content, and execution status (subject to authorization constraints).

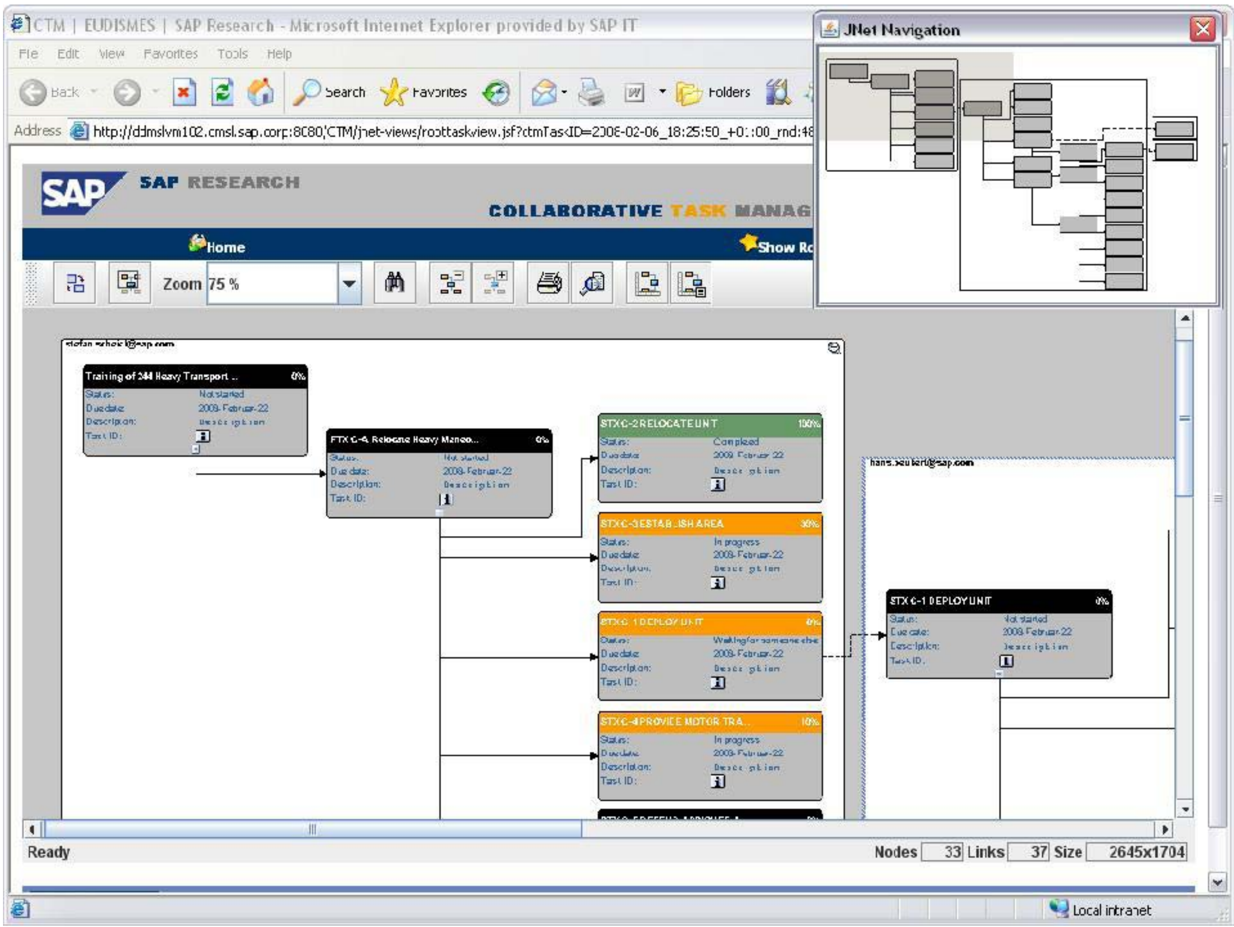

Figure 7: Graphical end-to-end process overview based on information tracked on the backend system.

This information is offered in a visualization enabling easy navigation and drill down functionality (Figure 7). This process view offers a highly intuitive process representation. It enables users to easily recognize their role in the overall process. The process view can be inspected by a recipient of a delegation request before his/her decision to accept, decline or negotiate the request in order to find out what has to be done exactly and how this task is related to other tasks. The process view also shows information about:

- who is assigned to which task
- what is the status of the task (not started, in progress, completed)
- the due date of the task
- the description of the task
- the documents attached to the task

The information in the process view helps to identify potential bottlenecks. In addition it is possible to track the status of a process. All persons involved in the process can examine who else is involved. In addition users easily get an end-to-end overview over the whole process telling them the actual state. By this it is possible to find out who is actually working on which task with which status. If person A is for example waiting for another person B to complete a specific task and he sees that person B depends on another person C, who is ill, person A knows why he / she has to wait. This speeds up task execution as delays can be detected easily.

8 Agility via Support of Ad-hoc Process

In contrast to other workflow systems, the CTM approach always offers the possibility of ad-hoc deviations from the process model defined in the task pattern repository. Such deviations may be necessitated by exceptional circumstances. CTM leaves the responsibility for ad-hoc deviations by the acting people. Whenever task patterns that match an exceptional situation are available, such patterns offer guidance and can be composed in an agile manner to collaborative processes resolving the exceptional situation.

If no suitable task pattern is available, free ad-hoc task structuring and collaboration can be driven out of the office application. Due to the tracking functionality, previous ad-hoc processes can be

- analyzed on the backend system for guidance in a current process,
- analyzed on the backend system to discover tacit recurrent practices that may be candidates for explicit best practices,
- stored as task patterns in personal or shared repositories using the TPEE which allows the editing and optimization of task patterns.

Thus CTM enables ad-hoc deviations from best practices given in task patterns in a way that is easy to use but also offers all possibilities of process support available in CTM. Mission training now can be tailored to very specific situations.

From all of the attributes of the proposed approach, we can then map them to the developed C2 maturity model [23]. Specifically, the individuals involved in the planning collaborate on developing the interdependencies associated with the sub-tasks and negotiate a mutually acceptable course of action in a transparent manner. The approach allows for dynamic re-planning and adjustments based on newly available information along with the insertion of ad-hoc elements whenever this is necessary. We purport that this approach leads to “position 5” – shared understanding and dynamically adaptive – in the model. In addition, the framework provides an ability to reflect on the execution versus the plan and thus develop lessons learned and corrective actions to future similar situations.

9 Conclusions and Outlook

With CTM it is possible to define mission training plans in the form of task patterns. These task patterns have the following advantages:

- Users’ working tasks (e.g. actions needed in a STX) can be created and managed ad-hoc in hierarchical action lists in the individual users’ workspaces by inheriting all structure and context information in the defined task patterns. Task delegation can be done via email. Thereby recipients of collaborative tasks can receive further guidance based on task pattern suggestions. The personal task hierarchies of different process participants are integrated to end-to-end processes through tracking the email exchange for task delegation. Tracking of individual tasks in the personal workspaces reflects the latest task state in the process overview. This enables transparency into parties involved in a process, their tasks, and the status thereof.
- Task patterns, as alternative to textual mission and training plans, represent adaptable, reusable building blocks of collaborative processes. They can be used to construct task hierarchies for complex collaborative processes. This means that the US Army sample FTXs and STXs derived from their documentation can be written down in the form of

task patterns. Task patterns are reusable task structures, comprising a task and its subtasks with the complete task context. Task patterns can be adapted to the current situation, in which a certain mission training has to be done.

For the future it is planned to enhance the recommendation system used in CTM. The recommendations as described in section 5.3 are very simple. To offer more complex recommendations two enhancements would be desirable: If during creation of a new task, it is found that there exists a pattern similar to the newly created task, this pattern should be recommended. In addition to offering suitable task patterns during task creation it would be interesting to be informed, if there exist certain typical deviations. This means: A user creates a new task. He is informed about a task pattern that might fit his needs. As the pattern is suitable he decides to reuse it. But the user is not only informed of a matching task pattern it is also pointed out to him that there exist typical deviations from the pattern. He then can further decide if his situation requires one of these typical deviations.

This mechanism could also be used to automate the evolution of task patterns: If, after a certain time, it turns out that a specific deviation is used more often then the original task pattern, an administrator could be informed that it might be useful to use the deviation as recommended best practice instead of the original one.

Acknowledgements

The authors thank Todor Stoitsev for his valuable support and so many useful comments.

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

References

1. Aalst, v. d. W, and Weijters, A., "Process mining: a research agenda", Computers in Industry, Vol. 53, Elsevier B.V., (2003).
2. Abbott, K.R., Sarin, S.K.: Experiences with Workflow Management: Issues for the Next Generation. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 113–120. ACM Press, New York, (1994).
3. Baecker, R. M., Grudin, J., Buxton, W. A. S., Greenberg, S.: “Readings in Human-Computer Interaction: Towards the Year 2000” (Second Edition) Morgan Kaufmann Publishers, Inc. (1995).
4. 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).
5. 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 (2005), pp. 89-138. Lawrence Erlbaum Associates (2005).
6. Bernstein, A.: How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier. In CSCW’00, ACM Press, New York, pp. 279-288; (2000).
7. Cypher, A.: Watch What I Do: Programming by Demonstration. MIT Press, (1993).
8. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A., Mehanjiev, N.: Meta-Design: A Manifesto for End-User Development. Communications of the ACM, vol. 47, no. 9 (2004)

9. Gartner Research. Person-to-process Interaction Emerges as the ‘Process of Me’. Gartner Inc. (2006).
10. Holz, H., Maus, H., Bernardi, A., Rostanin, O., “From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management”, Journal of Universal Knowledge Management, Vol. 0, No. 2, pp. 101-127, (2005).
11. Jorgensen, H. D., Interactive Process Models, Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, (2004).
12. Lieberman, H., Paterno, F., Wulf, V.: End-User Development. Springer (2006).
13. Lieberman, H.: Your Wish is My Command: Programming by Example. Morgan Kaufmann, (2001).
14. MacLean, A., Carter, K., Lövstrand, L., and Moran, T., “User-tailorable systems: pressing the issues with buttons”, Proc. CHI 1990, ACM Press: pp. 175-182, (1990).
15. Riss, U., Rickayzen, A., Maus, H., van der Aalst W.M.P: Challenges for Business Process and Task Management. In: Journal of Universal Knowledge Management 0, 2; pp. 77-100, (2005).
16. 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) (in German)
17. Stoitsev, T., Scheidl, S., Spahn, M.: A Framework for Light-Weight Composition and Management of Ad-Hoc Business Processes. LNCS, vol 4849, pp. 213-226. Springer (2007).
18. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: Enabling End Users to Proactively Tailor Underspecified, Human-centric Business Processes. In: Proceedings of ICEIS 2008. pp. 38-46, (2008).
19. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: Architecture for End-User Driven Composition of Underspecified, Human-Centric Business Processes. In: Proceedings ICEIS 2008. pp. 165-172, (2008).
20. United States Army: http://www.army.mil/usapa/doctrine/MTP_1.html.
21. United States Army: http://www.transchool.eustis.army.mil/New%20PSD/army_mission_training_plans.htm.
22. United States Army: Mission Training Plan for the Headquarters, Transportation Composite Group. Available under: http://www.transchool.eustis.army.mil/New%20PSD/MOSGuidereferences/ARTEP55_62.pdf (2002).
23. David S. Alberts and Richard E. Hayes: Planning Complex Endevors. CCRP, pp. 159-179, (2007)