International Journal of Cooperative Information Systems
© World Scientific Publishing Company

BLENDING AD-HOC AND FORMAL WORKFLOW MODELS IN SUPPORT OF DIFFERENT STAKEHOLDERS NEEDS

ROBERTO SILVEIRA SILVA FILHO

Siemens Corporation, Corporate Technology
755 College Road East, Princeton, NJ USA 08540
roberto.silva-filho@siemens.com

MONICA MCKENNA

Siemens Corporation, Corporate Technology
755 College Road East, Princeton, NJ USA 0854
monica.mckenna@siemens.com

KEVIN MCDEVITT

Siemens Corporation, Corporate Technology
755 College Road East, Princeton, NJ USA 08540
kevin.mcdevitt@siemens.com

Received Day Month Day
Revised Day Month Day

Workflow Management Systems (WfMSs), also known as Business Process Management Systems (BPMSs) are increasingly popular in today's large organizations. In spite of this popularity, many processes are still supported by ad-hoc systems based, for example, on spreadsheets and homegrown databases. In particular, there is a lack of flexible process automation approaches that are able to bridge the gap between these ad-hoc solutions and large-scale systems. This paper reports on a flexible workflow management system and approach that blends formal and informal workflow modeling and execution, thus supporting different types of processes. We validate our work by discussing its design and implementation, and by analyzing its use in four different use cases within Siemens business units. We also discuss the role of action items as important flexibility mechanisms behind our model. Showing its ability to handle exceptions and ad-hoc workflows.

Keywords: CSCW, Flexible Workflow Management Systems, System Design, Experience Report.

1. Introduction

Workflow Management Systems (WfMSs), also known as Business Process Management Systems (BPMSs), have been around for over two decades. They employ process representations involving tasks, people and activities in the automation of coordination activities in business settings[1]. The goal of WfMSs is to better support users coordination by automating the task of collecting and disseminating information required for a given task[2, 3]. Another important benefit of these systems is the ability to track, report on, and monitor existing process instances, supporting the management and auditing of complex processes[4].

As any technology that strives to support humans in their complex coordination activities, early WfMSs suffered from various flaws. Problems such as: rigidity due to excessive process specification, inappropriate handling of exceptions, poor user interface designs, and divergences between the way work is actually done versus how it was specified were common[5–7,8]. Lessons learned in the early days of workflow automation technology resulted in improvements in their design and the consequent adoption of these systems by many organizations. In particular more adaptive and flexible workflow definition languages[9], and automation approaches[2, 3, 10] have been proposed.

Recently, WfMSs have been popularized in many organizations in the form of Enterprise Resource Planning (ERP) systems that automate well-regulated business processes in domains such as health care, accounting and banking. Commercial systems in these categories include SAP, Microsoft Dynamics, and Oracle to cite a few (*http://whatiserp.net/erp-comparison/erp-vendor-evaluation-2010/*). The successful adoption of these systems, however, come at high customization and maintenance costs, and not always support the flexibility required by some processes[8]. Hence, in spite of the availability of such tools, many business processes are still supported by simpler yet flexible solutions based for example on spreadsheets, home-grown databases, word processing documents and ad-hoc e-mail exchange. While these approaches are generally cheap and fast to develop, they lack the scalability and reliability necessary by many organizations. They are also difficult to manage and audit. Thus, the gap between ad-hoc home grown solutions and the complexity of existing ERP systems have motivated the research and development of light-weight, cheaper and more flexible approaches that automate business-specific processes at reduced costs[11], while provide the flexibility required to evolve, adapt, and handle exceptions in a flexible, yet accountable way.

In this paper, we report on the design and adoption of CMT (Change Management Tool), a flexible WfMS that bridges the gap between ad-hoc and ERP workflow automation approaches. This is achieved by the use of a flexible process model that supports structured, unstructured (ad-hoc) and semi-structured workflow execution. The key insight of our approach is the use of formal workflow models enriched with the concept of optional activities, role distribution lists, and user-defined, spontaneously triggered *actions items*. When combined with automatic tracking of decisions, e-mail notifications and Web-based user interfaces, these characteristics provide the flexibility required by users along with the accountability required by the organization, thus supporting change without jeopardizing compliance and auditing requirements.

This paper is organized as follows. In section 2, we provide a brief overview of the evolution of flexible and adaptive workflow systems. In section 3, we discuss our motivation in the context of Siemens business needs. In section 4, we present the main elements of our approach. Section 5 presents implementation details of our approach as the CMT tool. Section 6 discusses four case studies where the tool has been applied. Section 7 takes a closer look of the use of action items. Section 8 presents related work and Section 9 concludes.

2. Background and Motivation

Over the last three decades, WfMSs research and practice have undergone different stages, combining efforts from both industry and academia.

Early WfMSs systems relied on rather detailed and descriptive process models. The workflow engine and process languages used to orchestrate these processes had poor exception handling mechanisms, resulting in rather rigid processes that often failed to support the actual work activities and practices of the organization. In particular Suchman[7] illustrates these problems by analyzing the complexity of interactions involved in office work. Her work discusses why everyday activities cannot be completely captured and formalized in the predefined set of steps of existing formal process languages.

These findings motivated the study for more flexible workflow management systems, focusing on adaptive process languages and exception handling approaches[5]. In particular, Schmidt[12] suggests that, instead of striving to capture all possible aspects of work, process descriptions should be seen as high-level “maps” and “scripts”. i.e. elements that prescribe the work and provide orientation, rather than describing rigid sets of steps that must be followed in the right order. According to Divitini & Simone[13], *"The idea that formal constructs, and hence workflows, incorporate working practices both as maps and as scripts implies that, as maps, the rules are partially specified (under specified) and formulated in such a way that they can be modified, adapted at a reasonable cost, in relation to the changing requirements of the organization and application domain."* Hence, Divitini & Simone suggest that the adaptability of technology can be achieved by the combination of language (the language should fit the process at hand. Multiple languages can be used), visibility (support adaptability through process awareness) and modification of formal constructs in order to better match the domain. In another work, Heinl et al.[14] proposes a set of concerns to be considered in the design of adaptive workflow systems. These include Infrastructure, Resource, Process and Domain.

These approaches informed the development of sets of systems. In particular, a collection of research in flexible (or adaptive) workflow was published in the journal of CSCW special issue on Adaptive Workflow Management Systems[9], which provides different tools and insights on how to design and support variability in this domain. The majority of the approaches discussed in Ref. 9, however, are based on the prescriptive nature of workflow modeling languages. In other words, they rely on process models that are able to describe exceptional conditions, and assume that the workflow engine is solely responsible for interpreting and enacting the workflow model. In these approaches, even though work can deviate from this original plan, to a certain extent, any deviation must be first formalized, during an exception handling procedure, and incorporated in the process description.

More recently, Jorgensen[10] proposed a radically different approach, named interactive enactment. It is based on incomplete or non-completely defined models that are tailored to each process instance as it executes. In this approach, workflow enactment is partially automated, and users are responsible for resolving process ambiguities by making informed decisions based on the current state of the process. For example, whenever an activity is not formally specified, the system interacts with users in order to decide: which user(s), when, and which resources to use.

Whereas these research results represent promising approaches for the design of more flexible WfMS, many of these ideas have not been proven in industrial settings. Moreover, flaws in workflow management systems design and adoption are still common place in organizations[8]. In fact, from an industrial perspective, workflow automation has been supported by either large one-size-fits-all customizable infrastructures, or by loosely automated ad-hoc processes.

In particular, Siemens is a large organization composed of many business units within divisions within sectors such as: Energy, Industry, Health Care and City Infrastructures. Each business unit manages a large number of projects. Project management is important for a number of reasons including auditing, transparency, fine-grained understanding project costs, and budget management.

Towards the end of 2005, Siemens made the decision that it would be useful if all its business units collected NCC (Non-Conformance Cost) information about projects within their divisions on a wide-scale level in a more systematic way than was currently being followed. The overarching applicability of the process needed for this data collection motivated the development of a flexible WfMS: one that could meet the specific NCC processes needs of each organization. In particular we identified the needs and issues described in the paragraphs below.

From an organizational point of view, the adoption of new processes must face two main challenges: platform heterogeneity and the need for low adaptation costs. Hence, instead of adopting ad-hoc solutions, based for example on spreadsheets or home-grown databases, developed for each business unit, we opted for designing a flexible workflow engine, that could be integrated with existing ERP systems, and which process could be flexible enough to fit the needs of different business units and their processes, yet accountable to the point of enforcing key procedures and norms. Besides interoperability and process flexibility, the following issues motivated our design:

Need for consistent project information. Project changes were sometimes not being tracked appropriately. One of the reasons was the inability of corporate ERP systems to capture and coordinate project-specific data. As a result, for some locations, project change tracking was performed in an ad-hoc manner: using spreadsheets and various file shares, which resulted in inconsistent reports.

Need for approval history accountability. Histories of approvals (who, why, when, where) are necessary pieces of information required for audits and change management in general. When this approval process was based on emails, faxed documents, and phone calls, it was very difficult to produce proof of the approvals for an audit.

Need for increased chain of approval awareness. Chains of approval define the main workflow being automated. They integrate data items, activities (or steps), roles, and users. Originally, it was difficult to understand the chains of approval of non-conformance costs, and it was difficult to see if the chain was actually followed. E.g.: for any given non-conformance cost (or value), who is the person that must approve it? Who else should be informed? When is it ok to pick somebody else? Who else can I pick?

Need for organizational change awareness. Changes to Delegation of Authority (or DOA) were also painful. Delegation of authority defines a hierarchy of management, approval and roles. They define who must approve certain type of non-conformance cost, based on its value, for example. As in any organizational hierarchy, as the DOA changes over time, it requires means to update all affected parties. This was typically performed via emails and process documentation updates distributed to all users. The result was a lack of a central database, where all this information could be easily found and applied to the situation.

User-friendly UIs. People also need familiar user interfaces. Older versions of ERP systems, that still run in some business units, have outdated UIs, for example, text-based interfaces that employ terminologies that are specific to their systems, as opposed to each user domain. There was a clear need for usable interfaces that could fit a broader audience, including not only experts but also non-experts. Putting it in simple terms, many ERP systems have interfaces that "do not speak the language of their users".

Need for better means of coordination. People need to coordinate in different situations. For example, when selecting dates for a specific event to occur, both physical resources as well as people availability must be considered; and the approver selection for each date event may be different depending on the type of event. Trying to coordinate this by exchanging emails or by phone calls was very time-consuming and difficult to organize.

3. Approach

In response to the needs previously discussed, we developed an adaptive workflow management approach and a tool called CMT (Change Management Tool). The central goal of the CMT is to centralize organizational and process data, decisions and communication in a single workflow management system that can be customized to support the needs of different business unit stakeholders within Siemens.

The CMT implements an approach combining the set of elements depicted in Fig. 1. It includes a data model, composed of individual work items of different user-defined types (non-conformance costs, opportunities, change orders, reconciliations, etc.); a process model (business logic) that prescribes a workflow process called chain of approval, which defines activities that must be performed in the life cycle of each work item, and an organization model that represents the Delegation of Authority organizational hierarchy, including approvers in each role. This allows the process model to be role-based and tied to the organization model.

The process and the data models are integrated by means of events i.e. originating messages that are distributed to users, according to a set of delegation of authority rules. Work items represent units of data being tracked by the system.

The interface between the system and their users is Web-based. From any workstation in the organization, users can configure workflow processes, modify its attributes, query the current status of a work item, locate other relevant items, and generate reports. The Web interface not only supports the ubiquitous use of the system but also supports the users' language. In this approach, the process model and language details are hidden from end-users. Instead, users implicitly configure the system by defining distribution lists, item attributes, and a chain of approval via forms in the UI. The system them maps those to an internal process representation that gets executed by the process engine, and a scheduler that periodically reminds users of open activities.

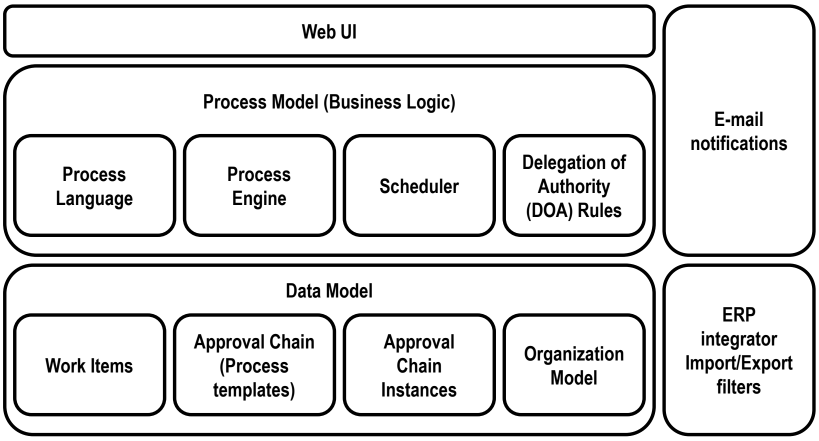

Web UI
Process Model (Business Logic)
Process Language
Process Engine
Scheduler
Delegation of Authority (DOA) Rules
E-mail notifications
Data Model
Work Items
Approval Chain (Process templates)
Approval Chain Instances
Organization Model
ERP integrator Import/Export filters

Fig. 1 Main elements of the approach

The flexibility of the system is supported by a data model that stores contextual information about work items and the process, and a workflow model that accommodates three main usage modes: structured, unstructured (or ad-hoc) and a blended modes further discussed in section 5. The system allows tasks to be reassigned, created and enacted on demand. It then automatically captures the history actions performed to work items associated to those activities. Hence, instead of requiring users to first change the workflow model, and then execute an exceptional case, the CMT system tracks the work assignments (action items, and optional activities) as they go. In this way, the workflow becomes more fluid; not requiring extra restructuring stages, while the work still remains accountable, being recorded as it progresses.

The dynamic operation of the system is orchestrated by a mix of process enactment, which assigns tasks and work items to users, and by e-mail messages that notify users of work assignments. The system notifies all required parties when users modify the content or status of a work item (e.g. posting a new conformance cost or modifying an existing one), this includes the user herself and the process listeners, owners and managers. Work item changes and approval requests are then sent to their assigned users via e-mail notifications, which include a description of the work item nature, status and required action. Finally, the system can periodically remind users to respond to open actions, resulting in recurring e-mail notifications.

From a theoretical standpoint, The CMT approach goes in line with the ideas of “maps and scripts” proposed by Schmidt[9] and Jorgensen[7] interactive enactment. In the CMT approach, the process definition and the organizational structure work both as "maps" of the organization and "scripts" of the chains of approval a work item must pass through. This information is provided in context, allowing users to decide who to request approval from, and what steps to take from a current state. The workflow model supports optional activities that are automatically included/excluded from the workflow instance based on the state or content of a work item. For example, if a work item has a value higher than a threshold, an extra approval step, involving high management becomes a requirement in a chain of approval. Finally, CMT also supports interactive enactment by means of ad-hoc activities called *Action Items*. Action Items are spontaneously defined tasks that users can create throughout the life of a process. They can be used to delegate tasks to other users, ask for comments, mark important events of the systems, and so on. They work as building blocks for ad-hoc user-defined workflows, adding an extra layer of flexibility to the system. Action Items will be later described in this paper.

In the following sections, we describe in more detail, the basic elements of the CMT approach.

3.1. *Data model*

The system data model, illustrated in Fig. 2, encompasses work items, the delegation of authority hierarchy, workflow templates (known as approval chains) and their respective instances.

Users belonging to a delegation of authority are responsible for creating, analyzing, approving, and updating work items according to a possibly complex workflow.

3.1.1. *Chain of Approval*

A chain of approval of a work item defines general workflow process to be followed during the evolution of a data item from the time it is created, opened, modified and closed. The process model is based on optional and mandatory steps (or activities), having roles defined in terms of an existing delegation of authority organizational hierarchy. There is usually one general chain of approval template per business unit per data item. The chain of approval defines "fixed parts" (mandatory steps) and "soft parts" (optional, listener steps and action items), together with the role of users that perform each step in this chain. The actual execution of the approval chain may vary, and is recorded in each work item. The roles and the order of activities are fixed. When roles are assigned to steps, any user within that role can be selected for that task. Activities can be dynamically selected based on different item content criteria, for example, total order value, risk, and other attributes.

3.1.2. *Work items*

Work items, illustrated in Fig. 2 are the main data elements being tracked by the system. Their life cycle is regulated by specific instance of the approval chain, and its history. The specific instance of the approval chain is the actual description of the approval workflow for a given work item.

Originally, the first type of work item supported by CMT system was non-conformance costs (or NCCs). In subsequent releases of the system, the set of supported work items was diversified to represent: change orders, risks, opportunities, customer concessions, meeting schedules and other business related data. Each variation of the item has a different set of data fields, and some data fields are given special meaning as cost, location, meeting time, etc.

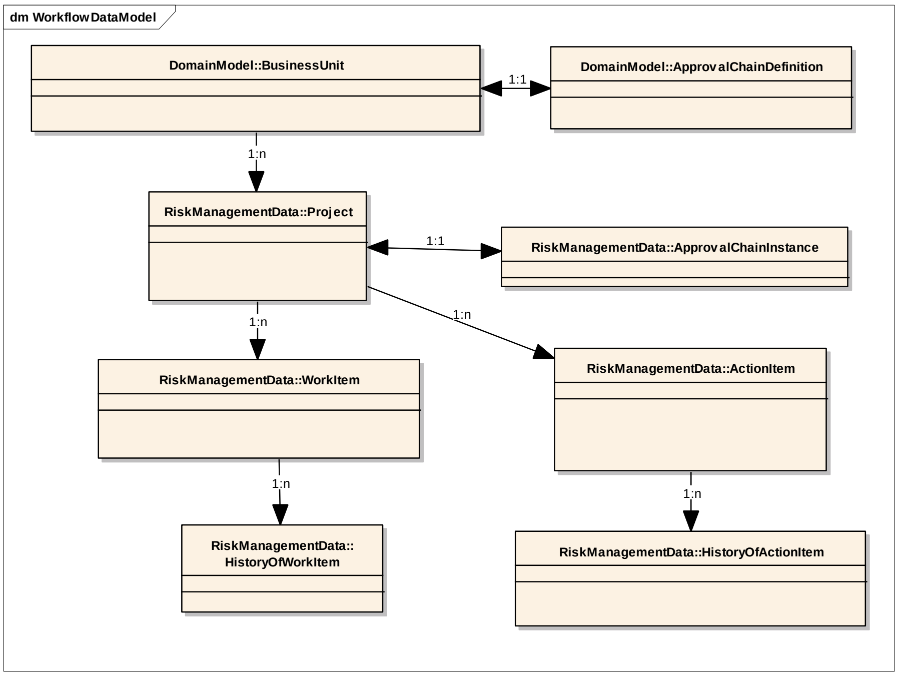

dm WorkflowDataModel
DomainModel::BusinessUnit
1:1
DomainModel::ApprovalChainDefinition
1:n
RiskManagementData::Project
1:1
RiskManagementData::ApprovalChainInstance
1:n
1:n
RiskManagementData::WorkItem
RiskManagementData::ActionItem
1:n
1:n
RiskManagementData::
HistoryOfWorkItem
RiskManagementData::HistoryOfActionItem

Fig. 2 Data model

3.1.3. *Contextual data model*

One key insight in the design of the system is the support for the main user needs represented by the three ‘W’s and one ‘H’: who, when, where, and how. Many users are interested in a data model that allows the answering of questions involving these subjects. For example, who was responsible for that purchase, when it was performed, and how much is the total today? In order to support those queries, the data model stores key contextual information for each work item in the database.

Besides the contextual information, the work items in the model also supports user-defined fields (typed attribute/value pairs), allowing the model to be tailored to different application domains and purposes.

3.1.4. *Automatic tracking of work items history*

Every work item can have its history traced. A history of a work item is its set of attributes events and chain of approval. As such, the automatic tracking of item history is the mechanism, together with the e-mail-based integration, that supports implementation of the different workflow modalities of the system. As every action is recorded, accountability is guaranteed.

3.2. *Workflow model*

The workflow model defines the basic elements of the language used to describe the processes enacted by the business processes that are automated by the system.

3.2.1. *Organizational modeling and role-based workflow*

In order to support the approval process, the system keeps track of organization structure with its roles and actors that fulfill these roles. The roles are defined based on the business process being automated, for example, project manager, resource deployment manager, coordinate, financial officer, CEO. Users are then assigned to the different roles. Each role can have one or more assignees.

Roles are also used for privilege and access control to different software features. For example, administrators can define the initial process for a business unit, and can configure the set of modules supported by the system.

The system also allows role transfer. I.e. users have the ability to transfer or reassign other users to different roles. For example, if an activity was assigned to a person in a role incorrectly, that person can transfer the item to a different person within that same role.

3.2.2. *Delegation of authority (DOA) rules*

Represent a set of optional rules that allow the system to assign an approver for an activity. The tool automatically enforces some rules, while other rules are enforced by convention. For example, the name of a role associated to an activity may include a monetary value. With that information, users know, for example, that if an NCC item amount is over $100,000, the item must be routed to the CEO for approval.

3.3. *User interface*

During workflow execution, events are triggered when users approve items, or when they create or change the status of an action item. These events will re-compute the item’s values and statuses, which will then trigger additional activities in the workflow instance and generate new email notifications as needed.

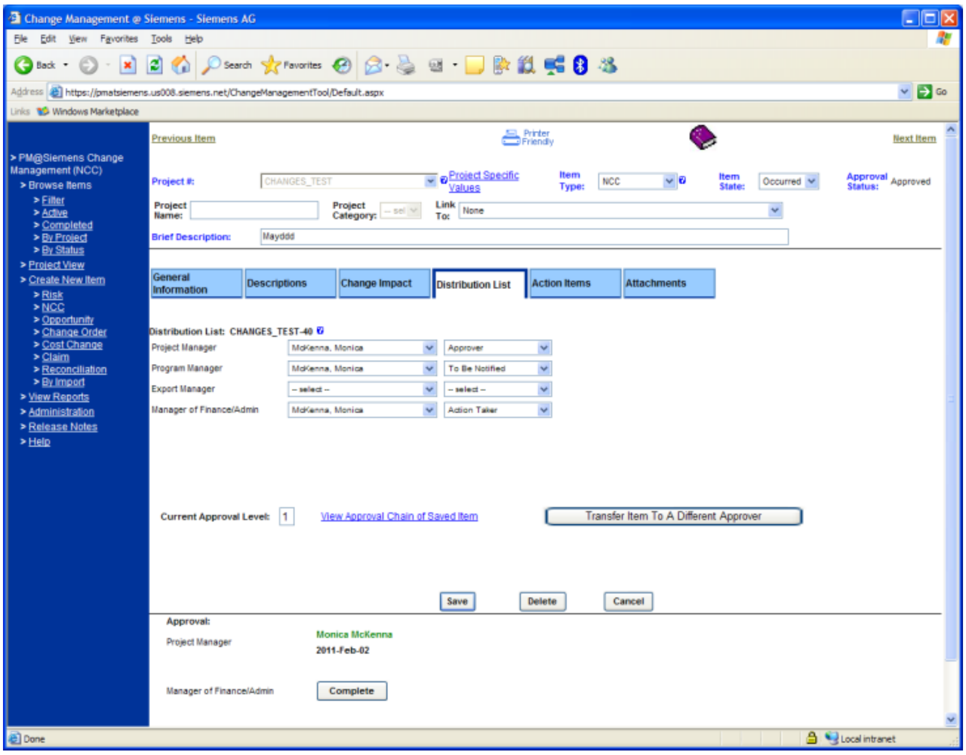
3.3.1. *E-mail based notification*

The interaction of the system with its users is done by means of the CMT web application and e-mail notifications. E-mails with hyperlinks describing the work items are sent to the users. These hyperlinks lead users to the CMT web page where the work item is described, and where the user can take the appropriate action. For example, when approving a request or generating a report, the user is given a URL of that item history, with approval options. After inspecting its values, she decides to approve it. Upon completion, the system notifies the interested parties associated to that work item, e.g. the originator of the report and her manager.

Another important role of notifications is to remind the users of open action items. CMT will produce periodic notifications whenever incomplete action items over an extended period of time are found. This is done with the help of the scheduler process.

3.3.2. *CMT Web-based UI*

While e-mail is used as the main notification mechanism of CMT, the end-users interaction with CMT system is mainly Web-based. i.e. e-mail notifications contain links to Web pages such as those of Fig. 3 where users can log-in and select, for example, among a list of approvers (distribution list) for a task.



Project #: CHANGES_TEST

Item Type: NCC

Item State: Occurred

Approval Status: Approved

Project Name:

Project Category: -- sel

Link To: None

Brief Description: Mayddd

Distribution List: CHANGES_TEST-40

Project Manager	McKenna, Monica	Approver
Program Manager	McKenna, Monica	To Be Notified
Export Manager	-- select --	-- select --
Manager of Finance/Admin	McKenna, Monica	Action Taker

Current Approval Level: 1

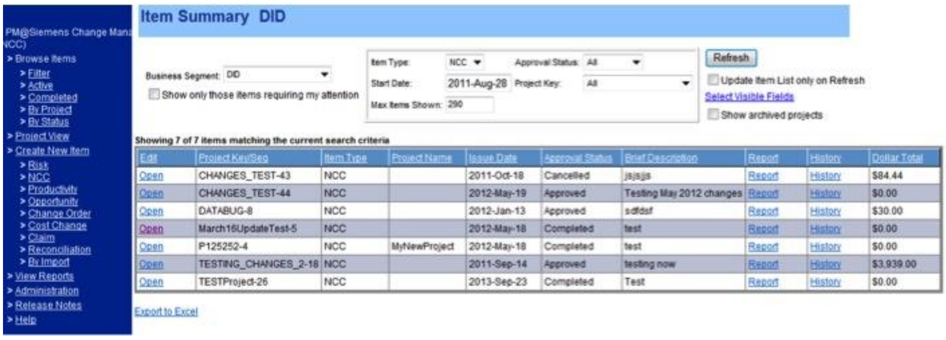
Approval:

Project Manager Monica McKenna 2011-Feb-02

Manager of Finance/Admin

Fig. 3 CMT UI: Example of a process rules definition page

The UI task structure is organized around the concept of Items. A workflow template (or chain of approval) can be defined for each item. When a new item is created, a new workflow instance is automatically started. Using the same UI, users can verify existing and past item records, and can generate reports. Thus the UI design reflects an item-based structure (as shown on the left tab of Fig. 4). By selecting different items, users can verify each item current status, and perform operations that change their content.



Item Summary DID

Business Segment: DID
□ Show only those items requiring my attention

Item Type: NCC
Approval Status: All
Start Date: 2011-Aug-28
Project Key: All
Max Items Shown: 290

□ Update Item List only on Refresh
□ Show archived projects

Showing 7 of 7 items matching the current search criteria

Edit	Project Key/Seq	Item Type	Project Name	Issue Date	Approval Status	Brief Description	Report	History	Dollar Total
Open	CHANGES_TEST-43	NCC		2011-Oct-18	Cancelled	jsjsjs	Report	History	$84.44
Open	CHANGES_TEST-44	NCC		2012-May-19	Approved	Testing May 2012 changes	Report	History	$0.00
Open	DATABUG-8	NCC		2012-Jan-13	Approved	sdfdsf	Report	History	$30.00
Open	March16UpdateTest-5	NCC		2012-May-18	Completed	test	Report	History	$0.00
Open	P125252-4	NCC	MyNewProject	2012-May-18	Completed	test	Report	History	$0.00
Open	TESTING_CHANGES_2-18	NCC		2011-Sep-14	Approved	testing now	Report	History	$3,939.00
Open	TESTProject-26	NCC		2013-Sep-23	Completed	Test	Report	History	$0.00

Fig. 4 CMT UI: Example of items assigned to a user

The use of Web-based interface is also important for its ubiquity. From any modern Web browser, approvals can be triggered, and the current status of an item can be verified.

3.4. *Process model*

In this section, we discuss, in detail, the process model behind our approach. In CMT, the workflow model is defined in terms of different activities or steps. A process definition is known an *approval chain* or *distribution list*. We use both terms interchangeably.

3.4.1. *Approval chains*

In CMT, a process description template is called an approval chain. Every business unit where the tool is deployed is responsible for defining its own general approval chain within the constraints allowed. Each data item that is created and tracked has an associated instance of that general approval chain.

Approval chains are workflow processes descriptions composed of steps (shown as solid, dashed or dotted boxes). Each step represent a work item assigned to a user or a list of users within a role. Steps have dependencies one each another, being organized into control flows with begin and end as shown in Fig. 5. As previously discussed, each process instance has a data item associated to it. Special types of steps called action items are represented as ellipses. Processes can be executed in sequence and in parallel. Parallel processes can be synchronized with AND and OR types of join points.

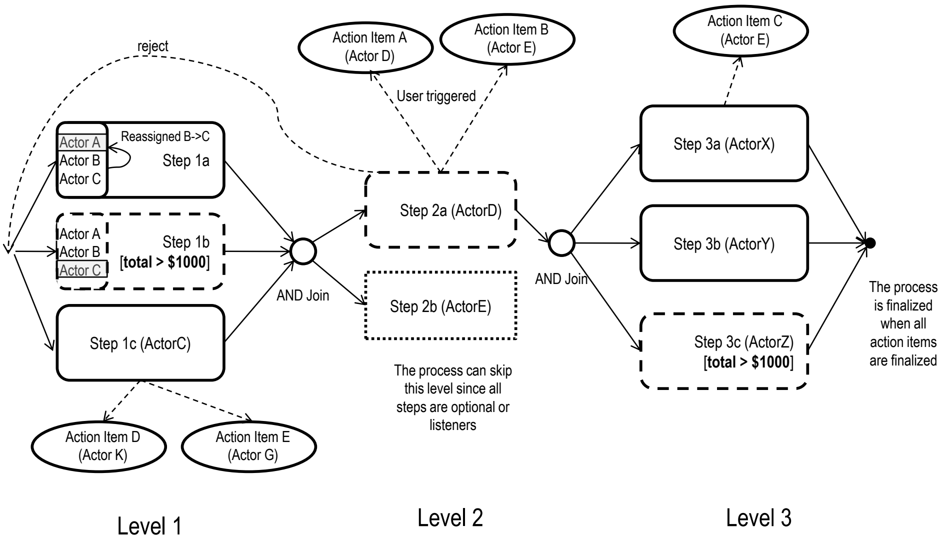


Fig. 5 Workflow process instance A

For illustration purposes, Fig. 5 depicts an approval chain having three levels. Each level is separated by an implicit synchronization point (and join), which has an implicit approver.

Different types of steps are supported. These include (1) required steps with no pre-selected actor (approver), (2) optional steps with no pre-selected actors, and (3) listener steps with no pre-selected actors. We also support all of these types of steps with pre-selected actors, in cases where there is only one choice for that step. A full list of steps and their variations are shown in Fig. 6.

Overall, we use a process model and notation similar to that proposed by the Workflow Management Coalition (http://www.wfmc.org), but with adaptations to express: actors lists for each step, as well as optional, listener and action item steps. We further define the types of steps as follows.

A *required step,* represented as round solid boxes, indicates a mandatory task in a process. Required steps may or may not specify lists of actors to perform them. Bindings of steps to actors are performed at runtime. When a list of actors is provided, the system assigns one of the listed users to perform a task according to different policies. If a specific actor is already specified for a task (list of actors size = 1), no user intervention is required. If no actor is specified, the user who initiated a process instance must select someone to perform those unassigned steps before the step starts. In the CMT system, this selection is typically assisted, i.e. the user picks a person from the list of options that they are given (which is defined for the particular business unit – users in a business unit are each assigned to 0 to n roles).

In other words, a required step can be expressed by the following Java style pseudo code algorithm:

```
required_step_for_item(ItemType I, Actors A, String desc) {
    bool accepted = false;
    while (size(A) > 0 && ! accepted) {
            // different selection policies apply
            actor = selectNextActor(A);
            String reasonOfDenial =‘“’;
            accepted = assignTask(I, actor, reasonOfDenial);
    }
    if (! accepted) {
            returnTaskToOwner(I, desc, reasonOfDenial);
    }
    // the user will receive an e-mail notification with the
    // description of the task
    String failureReason =‘”’;
    bool success = performTask(I, actor, desc, failureReason);
    if (! success) {
            returnTaskToOwner(I, desc, failureReason);
    }

    if (isListenerStep(this)) {
            notifiyListeners();
    }

    // by default, will notify approvers, owner
    // and will start next step

    return true;
}
```

Optional steps, represented by dashed rounded squares, are used to specify tasks that are not always required in a process instance. For example, an extra verification step, or the ability to express that that many roles can book a room. Similar to mandatory steps, an *optional step* with no pre-selected actor or actor list indicates that when an item is created, the item creator may select someone to be in that role from the provided list of users. For example, if they know an item is related to a sales issue, they may add the sales manager to the approval chain for this item. For a case that is not a sales issue, they may leave the role selection blank.

Optional steps can be expressed by the following algorithm:

```
optional_step_for_item(
    ItemType I, Actors A, String desc, Expression cond) {
    // verifies if step was selected
    // or if data trigger condition is met
    if (activitySelected || evaluateCondition(cond)) {
            return required_step_for_item(
                    ItemType I, Actors A, String desc);
    }
    return false; // means it does note require execution
}
```

Data triggered steps. Some optional steps can change from optional to required based on a data trigger. For example, if a NCC (non-conformance cost) process, if a data item total dollar amount is over a specified value, an approval step becomes mandatory, and an approval from the General Manager of the process will be required. Data triggered steps follow the same actor assignment rules as optional and mandatory steps, except that the user assigned is performed when the trigger value becomes true.

A *listener step,* represented by doted rectangles, are used to allow selected actors to be notified via email when tasks and properties associated to a data item change. Though users assigned to a listener step can view and modify information about an item, they have no approval or rejection authority. Those steps are used for auditing and overseeing purposes.

Step execution/approval and rejection. Actors are also know as approvers in the CMT system, this comes from its original role as a non-conformance tracking system, where steps were used to approve o reject an item based on its cost. In the CMT system, any approver (or actor), whether in an optional or required activity, can stop the whole process (or approval chain) by rejecting (not approving) a step. This resets the item's workflow process and sends it back to its initiator (though the history of why it was rejected and the previous approvals that did happen is stored). For example, a purchase order can be defined requiring two steps: evaluation and purchase. Once started, the order gets approved by one user, but rejected by the other user that purchases it due to insufficient data. The process is then aborted and the originator of the order is notified with a reason of denial. Fig. 5 illustrates this situation. In this case, a Workflow Process Instance A, the dashed line labeled "reject" shows this workflow event being triggered. Once this rejection occurs, the item can then be started again on this approval chain, or left in the rejected state.

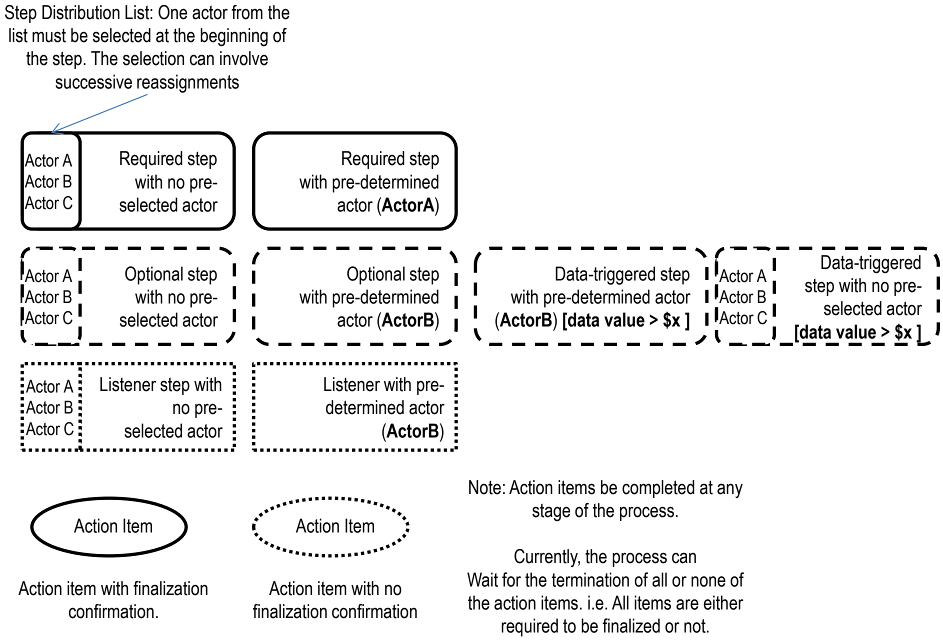

Step Distribution List: One actor from the list must be selected at the beginning of the step. The selection can involve successive reassignments
Actor A
Actor B
Actor C
Required step with no pre-selected actor
Required step with pre-determined actor (**ActorA**)
Actor A
Actor B
Actor C
Optional step with no pre-selected actor
Optional step with pre-determined actor (**ActorB**)
Data-triggered step with pre-determined actor (**ActorB**) **[data value > $x]**
Actor A
Actor B
Actor C
Data-triggered step with no pre-selected actor **[data value > $x]**
Actor A
Actor B
Actor C
Listener step with no pre-selected actor
Listener with pre-determined actor (**ActorB**)
Action Item
Action Item
Action item with finalization confirmation.
Action item with no finalization confirmation
Note: Action items be completed at any stage of the process.
Currently, the process can Wait for the termination of all or none of the action items. i.e. All items are either required to be finalized or not.

Fig. 6 Overall workflow model step representations

Actor assignment policies. The lists of actors in each step work as priority lists for task assignment. When a selected user cannot perform a task, the system can automatically select the next user in the list. It's also possible for an actor, once they get the item, to re-assign their task to someone else, if reassignment option is selected for a particular step. For example, in Fig. 5, in Step 1a, we can see actor B reassigning its task to actor A. Different policies can be adopted: random assignment within the list, priority lists, or as previously discussed, manual assignments of unassigned tasks when an item process is started.

Join points. We follow the Workflow Management Coalition notation for joins, allowing AND and OR types of joins, with similar semantics. OR joins allow the following step of a join to proceed when one or more incoming branches are completed; whereas AND joins require all the incoming branches to complete before the following set of activities can start.

3.4.2. *Action items*

Besides the use of mandatory and optional activities and the ability to reassign a task to another actor with a step, action items can be used to add an extra layer of flexibility to the approval chain by allowing actors to create new steps in the workflow. They can be started at any time in the life of an approval process, from the creation of the item up until any time before the item is completed. Users can configure whether or not all action items for an item need to be approved before the item is completed.

The algorithm for an action item is simple and general: if the next approval will result in the completion of an approval chain (so that all required approvals are made), check for open action items. If there are open action items, do not allow the final approval, and show an error message.

Action items can be assigned to any user of the business unit registered in the CMT. They represent a much more free form than approval chain roles. The creation of action items also triggers e-mail notification to the assigned actor. As with other types of steps, e-mail reminders are periodically generated while the action item is not followed up on.

A general action item can be expressed by the following pseudo code:

```
action_item( ItemType I, Actor a, String desc, bool confirm) {
    String reasonOfDenial =‘“”’;
    accepted = assignTask(I, a, reasonOfDenial);
    if (accepted) {
            waitForUserCompletion();
    } else {
            returnTaskToOwner(I, desc, reasonOfDenial);
    }
    if (confirm) {
            issueConfirmation(I, a, desc);
    }
    return true;
}
```

4. Implementation and usage details

The CMT tool is an ASP .NET web application with several libraries and is backed by a SQL Server database. The web application and libraries are on the order of 45,000 lines of C# code. The SQL Server database has roughly 50 tables, and over 100 stored procedures and user-defined functions.

The system has currently 675 active users, out of existing 2800 registered accounts over the last two years. It has been actively used on 23 different business segments within Siemens, among those, 10 are on trial stage.

5. Case studies

The CMT Tool has been used at different Siemens divisions and their business units for multiple purposes including tracking of NCC's and project related costs as: risks, opportunities, cost changes, and order changes in general. An initially unforeseen use of the tool was as a meeting schedule coordinator (use case 4).

Users report that while many tools exist to support workflow and document management, a key advantage of CMT tool is its customizability. By adopting CMT, a business unit can tailor the tool for their processes and their business without incurring additional development costs. This is supported by the flexibility of process, data and exception handling of the tool that can be configured via Web interface.

Another reported advantage of adopting the CMT tool is the reflective analysis about the process that it requires. In order to configure the tool, one must first understand the process being automated, which usually results in better awareness, planning and even some restructuring of the organization.

This section present anonymized case studies where the tool was successfully employed, showing the benefits of the different characteristics of the tool in supporting these processes. We haven't had so far any case where the CMT tool was not able to support process adequately.

5.1. *Case 1: Tracking of Non-Conformance Costs*

Some business units use the tool just as it was originally designed i.e. to track non-conformance costs (NCCs). In other words, they follow a normal chain of approval, generate monthly and quarterly reports, and are able to show an audit trail for the approval of their NCCs. This chain of approval as illustrated in Fig. 7 shows the use of listener steps and required steps in support of these common scenarios.

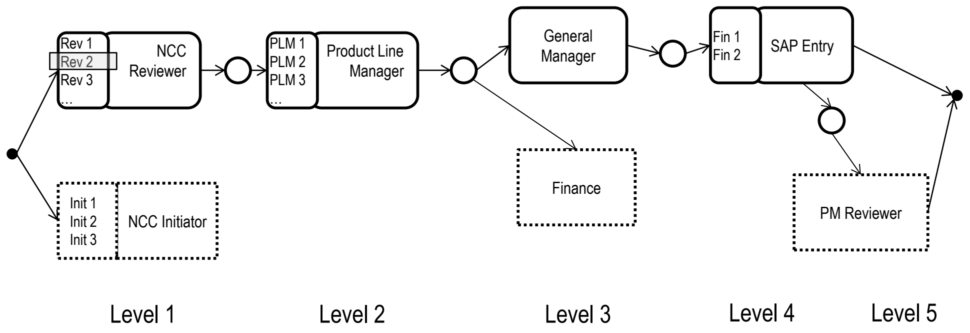


Fig. 7 Case Study 1 process description

In Fig. 7, the first level approver is an *NCC Reviewer*. Also at the first level, the NCC initiator is an optional listener (who receives e-mail notification when the process starts) – this is for the case when the person initiating the NCC was not the person who entered it into the tool – this listener step allows NCC initiators to be informed that the item was entered and is now in process. Note that we use roles instead of specific user names in this chain of approval. Roles add the flexibility to choose among existing qualified users.

The second level approver is the *Product Line Manager* where the appropriate person among the list of possible managers is chosen for the item. *Finance* is then optionally informed at the third approval level, but never approves or disproves an NCC. The *General Manager* always approves for this case. Finally, the results are sent to an *PM Reviewer* for final approval (and update the ERP system). They are also responsible for a final, as a last step, in order to verify that all data is correct for the item.

5.2. *Case 2: Tracking of Projects*

Some business units use multiple item types, not just NCCs, and track their projects, from the moment an order is accepted through its completion. Fig. 8 describes a typical workflow template in this case study, and illustrates the use of optional activities that are triggered based on item attributes.

In Fig. 8, the 'level 1' approver must always be chosen, and is the *Project Manager* for the project that this item is related to. Before approving, the *Project Manager* must verify that the item has been correctly entered with the correct dollar values, cause codes, and milestones, and that the approval chain going forward has the correct selections.

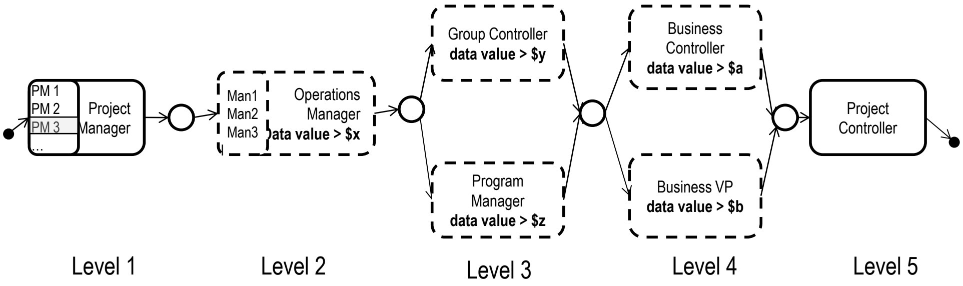


Fig. 8 Case Study 2 process description

If the dollar value is greater than x (and the work item is a certain type, such as an NCC or a Change Order), then the operations manager becomes a required approver, and the appropriate *Operations Manager* for this project is chosen. Once that approval is given, the approval chain moves to Level 3.

If the monetary value is greater than y, the *Group Controller* becomes a required approver, otherwise they can be optionally selected as an approver. If the dollar value is greater than a value z, then the *Program Manager* becomes are required approver. Once they both approve (if required), only then the item goes to the next level.

Again on level 4, a monetary value and item type is used to determine if the approvers are required (*Business Controller* and *Business Vice President*). The level 5 approver is always required, where this is the project controller who is then responsible for entering the item into the ERP system.

Hence, for a process instances with low monetary values, the approval chain might be just two steps, the first approver (the *Project Manager*) and the level 5 approver (the *Project Controller*).

Note that the initiator of the work item, the work item's creator, will set up the initial approval chain for the item. Once an item has been created for a specific project, the creator need only push a button, and the system will search for the "typical" approvers for this type of project, so the approval chain need only be verified.

Some roles are given the ability to further modify the chain of approval. In this case, the *Project Manager* has this ability. For example, suppose the creator of an item did not explicitly assign the *Program Manager* since the dollar value for this item was low. However, since the item is related to a project where the *Program Manager* has requested to be kept very closely informed on, the *Project Manager* can add herself as an approver for that item at a time during the item life cycle.

5.3. *Case 3: Blended Structured-Unstructured Workflow*

A business unit who had used the tool to track NCCs eventually had the monetary aspects of NCC tracking integrated into their ERP system, and therefore no longer needed to track the budget effects within the CMT tool. However, having gotten used to the workflow of the CMT tool, they continued to use the tool to track the evolution of NCC items, using both structured and unstructured workflows. They also relied on the tool's ability to collect and report on additional information about the items that is still not available within their ERP system.

This business unit's approval chain, shown in Fig. 9, starts out with many approvers, all optional, at the first level. These options include an engineering manager, a production manager, an operations manager, a quality engineer, and a product manager. Also at the first level, the marketing manager can be informed (as a listener only).

Any of these roles (as well as the item creator) can create action items. Some examples of action items in this business is to contact a supplier for additional information for the item, or to update a change form that is used in their process, or to verify that a specification related to a project has been correctly updated as related to the NCC item, or that an engineering drawing was updated as related to the NCC item. All of these actions fall outside the normal approval chain process, and may go to users in the system that are not anywhere on the approval chain.

After all select first level approvers have been chosen; the item goes the second level approvers. In this step, the item initiator is responsible for approving it. An item initiator may be the creator, or someone who asked that the item be created in the system. They can then see any comments or actions that were added by any of the first level approvers, and verify that the item has been correctly entered and that rest of the approval chain has been correctly specified as they feel is necessary.

All the next roles in this case's approval chain are optional. These approvers are all optionally selected, and each is defined in its own level (i.e. they are required to complete before the next activity may start); and roles as project engineer or database administrator may all become approvers for any given item, and may initiate additional action items.

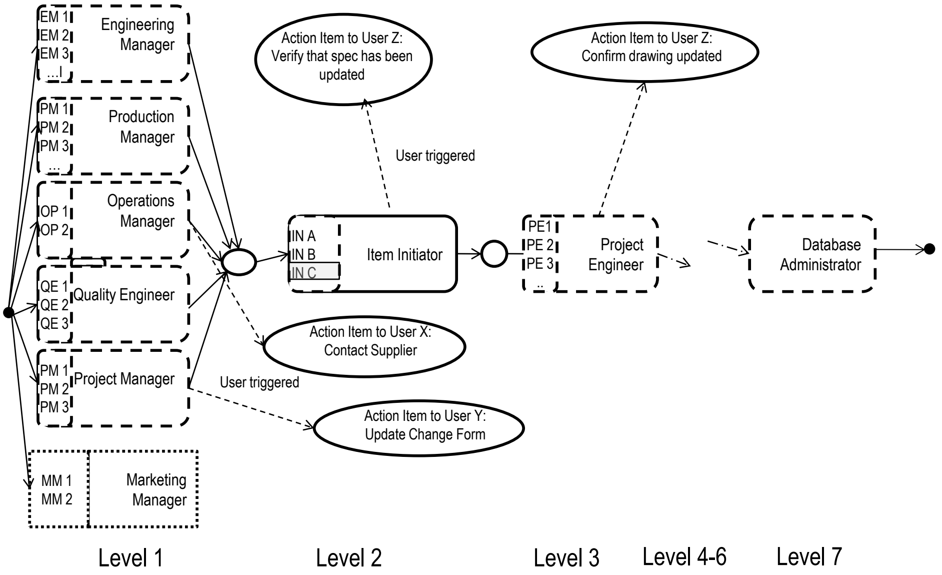


Fig. 9 Case Study 4 process description

5.4. *Case 4: Scheduling*

A completely unanticipated use of the CMT tool was scheduling. In this case, users utilized items to agree on data and place for an event e.g., a meeting or a presentation. For that, they utilize data fields of an item to record event properties, e.g. the event data, and a location. They then route the item to various approvers to get approval for the event. This action produces automatic e-mail notifications, send for each approver. Weekly reports are generated from the CMT tool from which the actual schedule is easily derived. A record of the approval process history for the event date is also available, allowing users to understand the rationale behind the final decision. The unstructured workflow is also sometimes used in this case when actions outside of the structured workflow approval need to take place.

This business unit's approval chain is similar to that shown in case 2, except that all approvals other than the first level approver, are optional, and the last level has four choices of roles instead of just one. This business unit also makes use of the action items created at any stage of the approval process. For this business unit, the work item cannot be completed until all open action items are completed.

6. A Closer Look at Action Items

A distinctive characteristic of the CMT process model is the ability to spawn action items at any point in time, supporting the spontaneous construction of workflow processes within the limits of the optional and required steps of a process template. This section discusses the use of action items in the different case studies where CMT was applied.

6.1. *Qualitative analysis: uses of action items*

Action items represent a very versatile feature in our approach. They are used by our client base to fulfill different roles in different processes, allowing users to create their own personalized workflow by handling exceptions and dealing with different ad-hoc situations. In order to better understand the role of actions items in the existing processes, we analyzed the data records of our case studies, looking at existing process instances logs, and derived a list of situations where action items are mostly used. In the business units studied, action items are mostly used to:

6.1.1. *To record the occurrence of corrective actions on items*

- E.g. providing additional training/documentation to new employees to reduce reoccurrence (frequent).
- Adding training sessions for all employees involved in the process.
- Updating documentation/drawings specific to the item (often).
- Updating process documentation, not always as related to this specific item but to the overall process.
- Review procedures for calculating information.
- Change the installation process or physical setup (frequent).

6.1.2. *As a record of actions taken with respect to a work item:*

- Contacting a supplier to inform them about an item.
- Contacting a vendor about replacement/repair.
- Contacting a supplier to hurry up delivery of a part.

6.1.3. *As way to request actions to a responsible person:*

- Someone notices something on the item is not filled in correctly, and uses an action item to inform the person with rights to change the incorrect item (frequent).
- Asking an administrator to have an item cancelled (happens in the event scheduling case only).
- Request to an approver to speed up an item approval.

6.1.4. *To obtain/request additional information about an item*

- Query sent to a user in the business unit asking a specific question about an item (many instances).
- Questions asked if an item is really correctly defined.

- Questions asked if a process related to an item was followed, or the status of that process.

6.1.5. *To invoke processes outside the normal item workflow*

- Start a separate billing issue investigation.
- Ask about hardware associated with the item.
- Assign a person to call another person to discuss something related to the item.
- Ask someone to take action in another tool (also tracking things related to this item) such as in a fault report or billing tool (frequent).

6.1.6. *To store additional information about an item and notify someone of places where information is located*

- Website links are added this way.
- Information about assignments to related tasks have been added this way.
- Inform someone that a meeting had been held (and at the same time record the information that meeting was held), or that an email was sent. (This is frequently used).

6.1.7. *To inform someone, not on the normal workflow chain, about the status of a work item*

- Action item is used just to make sure someone outside the normal workflow looks at an item.

6.2. *Discussion*

These situations can be classified in three major types of use: to handle exceptions, as when a user cannot fulfill a role at a given time, as an organizational record of both data items and workflow history, and as a way to improve the organizational awareness, as for example, by the sending of notifications to interested parties.

In fact, an important feature of the CMT system that contributed to the success of the tool in the organization was its ability to periodically generate e-mail reminders for currently opened action items. This was a feature requested by the users in the early stages of the development of the system. Even though this feature can be disabled, it is usually enabled in all case studies we analyzed. Many users appreciate this feature, and like the fact that the tool automatically reminds them of opened steps in the process, which helps in the timely response of tasks assigned to these users.

Another important aspect of our approach is accountability. As the system records the actions of its users, and notifications are produced to interested parties, the changes in the process are automatically recorded. Reports can be easily generated, while the auditing trail of the process of change and approval is preserved.

6.3. *Quantitative analysis: frequency of action items*

Another important question we would like to answer is how frequent or important are the use of action items in each use case. The analysis of the action items frequency reveals how common spontaneous or exception handling actions are in the types of processes studied, and indicate the relative need for this feature in common business workflows.

We compared the total number of work items with the total number of action items per case study as shown in Table I. We first observe that action items are only used in case studies 3 and 4. In cases 1 and 2, the flexibility provided by organizational roles, data triggered optional steps, and listener steps was sufficient for their users.

Table I Total number of work items and action items per case study

Use Case	1	2	3	4
Total number of work Items	249	170	764	3690
Total number of action items	0	0	92	293
Average action items /work item	0	0	0.12	0.08

In use cases 3 and 4, we see that action items are roughly used in 10 percent of the workflow instances of a certain work item types if we consider only absolute values of these use cases,. However, if we classify the work items by the number of action items (in Table II), we see a better picture of their use. I.e. even though the total number of action items are relatively low (Table I), many process instances do rely on their use, and have at least one action item.

Table II Total number of process instances with 1 through 5+ action items

Case Study	1	2	3	4
1 Action items	0	0	6	203
2 Action items	0	0	4	72
3 Action Items	0	0	6	15
4 action items	0	0	8	4
5+ action items	0	0	12	0

In the schedule case study (case 4) the workflow model based on optional activities was sufficient to formalize the variability in the process. Action items were mainly used to handle exceptions, thus its relatively low usage: about 8%.

In the blended workflow case study (case 3), users were many times required to interact with clients outside of the organization. This interaction was performed in a case-by-case basis, requiring constant exchange of e-mail with representatives. The need to handle these ad-hoc interactions was fulfilled by action items, thus their higher importance in that case study.

6.4. *Quantitative analysis: use of action items over time*

Another important factor to be considered is the learning curve of the system, and its features. Hence, we also analyzed the use of action items over time as shown in Table III. In this table, we see that users adoption of the tool have increased over time, having stabilized in the last quarters. We attribute this fact to the familiarization of the users with the system processes, and the adjustment of the core workflow capability to capture this core feature.

Table III Action item usage over time (quarters and years)

2009				2010				2011
Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1
14	46	77	93	110	95	93	69	87

It is also important to note that, in all case studies, action items are used as a complement to existing processes. No case was found where this mechanism was the only strategy adopted by users to enact the workflow. This shows that, in practice, there is a need for a balance between structured activities, belonging to the core process definition, and ad-hoc activities, required to handle exception and more spontaneous interaction.

6.5. *Summary of findings*

Action items fulfill an important role in the studied use cases. They simplify the treatment of exceptions in existing process, allowing more concise chains of approval. For example, one does not need to think about all the possible situations a chain of approval may handle, instead, they can focus on the required and some optional steps, leaving the handling of ad-hoc situations to be carried out in a case-by-case basis.

As noticed in the case studies, the use of action items varies with the nature of the process. When a process is well understood, more traditional mandatory and optional steps can be used. In many cases, roles and data triggers provide sufficiently flexible mechanisms for handling user and data variability, while the required steps are being enforced. It is on less structured situations that action items provide more value. Hence, instead of being a replacement to existing processes, action items are used as a compliment, achieving higher levels of flexibility in a usable and useful way.

It is also important to note that action items are not used to subvert existing processes, but to address the complexity and unpredictability of different processes. The use of e-mail notification and the automatic recording of item evolution and approval history provide basic elements for accountability, awareness, and coordination, making it possible for users to create custom, on-demand processes as those of use case 3.

7. Related Work

Existing work in this field of workflow automation encompasses systems in three main categories: structured or process-driven systems, un-structured or ad-hoc workflow management systems, and blended (combining structured and structured workflow approaches), which includes our own approach. As a blended approach to workflow automation, CMT provides features common to existing systems. This section describes existing related work in these fields.

7.1. *Structured approaches*

Different approaches have been proposed to support change and exception handling in structured workflow management systems. In particular, the approaches discussed in the Journal of CSCW special issue in adaptive workflow management systems[9], including Endeavors[15], and the work of Cugola et al.[16] support adaptability in the process description level, requiring dynamic changes in the process definition of a workflow instance before it can be enacted by the system.

Compared to these, our approach, on the other hand, is based on partially defined process models, that get constructed during process execution time. Instead of being only prescriptive, our approach can also be dynamically constructed with the help of action items.

7.2. *Un-structured approaches*

In general, unstructured approaches employ monitoring strategies to capture the workflow as they get executed, groupware tools such as e-mail, notification systems, shared repositories, and task lists to support coordination.

Some insights in the development of CMT system trace back to the Coordinator System[17]. In this system, the process is defined in terms of conversation moves, via e-mail. These conversational interactions are then recorded by the system, and making them available to users via a group calendar. As shown in our use cases, our approach can be used to support such coordination scenarios as well as more formal processes. The role of workflow in support of group awareness and self coordination mechanism was first analyzed by Dourish[4]. In our studies, users utilize item history change and approvals to self-coordinate and to assess the state of a process.

Another interesting research on contextual collaboration, utilized in the development of Activity Explorer (or AE) [18], supports the recording of complex collaborative sessions. In this system, unstructured work involving chat, screen sharing, video sessions, electronic meetings, and other synchronous interaction are automatically captured into activity threads, providing accountability and organizational memory. There is no formal process to be followed as activities are recorded as they unfold. Similar to AE, Our system employs the recording of events and the idea of notifications into action items and combines them with more formal processes and policies around data items.

More recently, a framework for supporting unstructured business processes is proposed by [19]. It utilizes e-mail, together with group and personal task lists to allow end users to create their own processes. Users can create hierarchical to-do lists by breaking down tasks into sub tasks. Tasks can be delegated, over email to other users. The recipients of tasks can further break down and assign tasks and (sub)tasks to other users. Changes of individual tasks in the personal end users' to-do lists are tracked by the system, and a web interface is provided.

7.3. *Blended approaches*

Different systems have strived to support a combination of structured and unstructured workflow automation. For example, one of the first systems to blend structured processed with ad-hoc non structured work was the *InConcert* system[3]. Some features of this system such as shared information space, e-mail notifications, support for monitoring and reporting, process mutability, and selective automation are some of the characteristics are common to CMT. The *InConcert* approach for expressing blended workflows, however, is programmatic. i.e. it is based on complex process definition languages. The lack of usability of such approaches have motivated projects such as the ADEPT [20], which discusses the usability challenges involved in supporting flexibility in workflow systems. A key difference of our approach is the user-centered control of process and tasks, via a simpler Web interface; and the support for formal, informal and blended workflow automation.

Existing research on business process modeling languages has studied the combination of formal (imperative) and policy driven (declarative) models as ways to support exception handling and ad-hoc process change [21]. In particular, language-based approaches as declarative modeling [22], generic process models [23], and the language patterns discussed in Ref. [24] and programming models such as *worklets* [25] have also been proposed as a way to capture commonality and variability in workflow processes. In those approaches, flexibility is achieved by a combination of process definition and policies. While the optional and mandatory parts of CMT process model requires formalization, our approach to ad-hoc action items allow users to perform complex activities without any formalization at all. This flexibility, however, comes at the costs of having other users involved in the chain of approval must then enforce workflow policies, which in our use cases have shown to be adequate.

The work of Dourish et al.[2] on the *Freeflow* prototype, and the work of Wainer et al.[26] both define a WfMS that are flexible towards the sequence of tasks proposed by a process. They show the importance of processes that separate the prescription of the dependencies between task activities from the actual temporal sequence of their performance. Flexibility is achieved by the use of constraint-based process language that supports the relaxation of activity order. In doing so, they support the execution of activities in different order than the process suggests. While our CMT approach does not fully support out of order execution of tasks across the different levels, this is possible within each level of activity. Our approach differs from these approaches by blending structured and unstructured (or ad-hoc) processes, allowing different degrees of freedom within certain limits of the process.

Finally, the work of r Stoitsev et. al. [27] supports collaborative end-user programming of business processes by the automatic recording of tasks, as they get enacted by users, It goes beyond ad-hoc workflow proposed in Ref. 19 by supporting the further formalization of those task hierarchies into process templates. These templates can then be instantiated by users, and customized at any time during a workflow instance execution. Users are enabled to deviate from a formal process instance by creating an ad-hoc task. That task is then incorporated into the existing workflow instance model. E-mail and to-do lists (of both individual and the workgroup) are used to assign tasks to users. Users can suggest list of users to perform the task, and data attachments can be incorporated in e-mails. An eclipse-based UI is provided supporting the transformation of personal task hierarchies into process models. Different from our CMT model, more complex tasks with data triggers are not supported. Also, there is no close integration with the corporate delegation of authority.

8. Conclusions

In this paper we discussed a workflow process model that allows users to cope with the variability of different workflow automation, and which distinctive feature is its support for action items, allowing users to spawn individual workflow tasks that are automatically monitored and managed by the system.

We present the design, implementation of the approach in the context of the CMT workflow management system, showing how it has been used within Siemens in a three-year period, describing examples from four business units, illustrating how the different types of activities and features from the approach are used in support of these scenarios. In particular, we analyze the role of action items in support of flexibility, and exception handling.

Our experience with the CMT tool shows that a combination of structured and unstructured workflow models provide the flexibility necessary for the support of different organizational processes, thus better supporting coordination activities within an organization where accountability, project and organizational awareness are important requirements. In particular, the use of formalized processes with required and optional steps, together with the triggering of spontaneous ad-hoc activities (action items), and the automatic recording of item history, have provided the flexibility necessary for handling exceptions and process variability our case studies. Thus, we believe such approach can be considered in the design of novel and more flexible workflow management systems in organizations in need of lightweight alternatives to more traditional ERP systems.

References

1 W. van der Aalst and K. van Hee, *Workflow Management: Models, Methods, and Systems*, 1st ed. The MIT Press, 2002.

2 P. Dourish, J. Holmes, A. MacLean, P. Marqvardsen, and A. Zbyslaw, “Freeflow: mediating between representation and action in workflow systems,” in *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, New York, NY, USA, 1996, pp. 190–198.

3 K. R. Abbott and S. K. Sarin, “Experiences with workflow management: issues for the next generation,” in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, New York, NY, USA, 1994, pp. 113–120.

4 P. Dourish, “The Dual Use of Workflow Technologies,” in *ACM SIG Group Conference*, 2001, pp. 52–60.

5 G. J. Nutt, “The evolution towards flexible workflow systems,” *Distributed Systems Engineering*, vol. 3, no. 4, pp. 276–294, Dec. 1996.

6 J. Grudin, “Why CSCW applications fail: problems in the design and evaluation of organizational interfaces,” in *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, New York, NY, USA, 1988, pp. 85–93.

7 L. A. Suchman, “Office procedure as practical action: models of work and system design,” *ACM Trans. Inf. Syst.*, vol. 1, no. 4, pp. 320–328, Oct. 1983.

8 T. Gross and S. Pekkola, “Three Levels of Failure: Analysing a Workflow Management System,” in *Reframing Humans in Information Systems Development*, vol. 201, H. Isomäki and S. Pekkola, Eds. London: Springer London, 2010, pp. 191–210.

9 M. Klein, C. Dellarocas, and A. Bernstein, “Introduction to the Special Issue on Adaptive Workflow Systems,” *Computer Supported Cooperative Work (CSCW)*, vol. 9, no. 3, pp. 265–267, 2000.

10 H. D. Jørgensen, “Interaction as a framework for flexible workflow modeling,” in *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, New York, NY, USA, 2001, pp. 32–41.

11 S. Brahe and K. Schmidt, “The story of a working workflow management system,” in *Proceedings of the 2007 international ACM conference on Supporting group work*, New York, NY, USA, 2007, pp. 249–258.

12 K. Schmidt, “Of maps and scripts - the status of formal constructs in cooperative work,” in *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge*, New York, NY, USA, 1997, pp. 138–147.

13 M. Divitini and C. Simone, “Supporting Different Dimensions of Adaptability in Workflow Modeling,” *Comput. Supported Coop. Work*, vol. 9, no. 3–4, pp. 365–397, Aug. 2000.

14 P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, “A comprehensive approach to flexibility in workflow management systems,” *SIGSOFT Softw. Eng. Notes*, vol. 24, no. 2, pp. 79–88, Mar. 1999.

15 P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi, and M. Bergman, “Techniques for Supporting Dynamic and Adaptive Workflow,” *Comput. Supported Coop. Work*, vol. 9, no. 3–4, pp. 269–292, Aug. 2000.

16 G. Cugola, “Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models,” *IEEE Trans. Softw. Eng.*, vol. 24, no. 11, pp. 982–1001, Nov. 1998.

17 F. Flores, M. Graves, B. Hartfield, and T. Winograd, “Computer systems and the design of organizational interaction,” *ACM Trans. Inf. Syst.*, vol. 6, no. 2, pp. 153–172, Apr. 1988.

18 W. Geyer, M. J. Muller, M. T. Moore, E. Wilcox, L.-T. Cheng, B. Brownholtz, C. Hill, and D. R. Millen, “Activity Explorer: Activity-centric collaboration from research to product,” *IBM Systems Journal*, vol. 45, no. 4, pp. 713–738, 2006.

19 T. Stoitsev, S. Scheidl, and M. Spahn, “A Framework for Light-weight Composition and Management of Ad-hoc Business Processes,” in *Proceedings of the 6th International Conference on Task Models and Diagrams for User Interface Design*, Berlin, Heidelberg, 2007, pp. 213–226.

20 P. Dadam and M. Reichert, “The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support. Technical Report. Fakultät für Ingenieurwissenschaften und Informatik, Ulm.,” Jan-2009. [Online]. Available: http://dbis.eprints.uni-ulm.de/487/. [Accessed: 03-Oct-2014].

21 M. Staender, A. Hadjakos, and D. Schreiber, “Adaptive Workflows in Smart Environments: Combining Imperative and Declarative Models,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, New York, NY, USA, 2012, pp. 1171–1174.

22 H. A. Reijers, T. Slaats, and C. Stahl, "Declarative Modeling–An Academic Dream or the Future for BPM?," in *Business Process Management*, F. Daniel, J. Wang, and B. Weber, Eds. Springer Berlin Heidelberg, 2013, pp. 307–322.

23 van der W. M. P. Aalst, "Flexible Workflow Management Systems: An Approach Based on Generic Process Models," in *Database and Expert Systems Applications*, T. J. M. Bench-Capon, G. Soda, and A. M. Tjoa, Eds. Springer Berlin Heidelberg, 1999, pp. 186–195.

24 B. Zimmermann and M. Doehring, "Patterns for Flexible BPMN Workflows," in *Proceedings of the 16th European Conference on Pattern Languages of Programs*, New York, NY, USA, 2012, pp. 7:1–7:9.

25 M. Adams, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows," in *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, R. Meersman and Z. Tari, Eds. Springer Berlin Heidelberg, 2006, pp. 291–308.

26 J. Wainer and F. de L. Bezerra, "Constraint-based flexible workflows," *Groupware: Design, Implementation, and Use*, pp. 151–158, 2003.

27 T. Stoitsev, S. Scheidl, F. Flentge, and M. Mühlhäuser, "From Personal Task Management to End-User Driven Business Process Modeling," in *Business Process Management*, M. Dumas, M. Reichert, and M.-C. Shan, Eds. Springer Berlin Heidelberg, 2008, pp. 84–99.