Business Conversation Manager: Facilitating People Interactions in Outsourcing Service Engagements

Hamid R. Motahari-Nezhad, Sven Graupner, and Sharad Singhal

Hewlett Packard Labs
Palo Alto, USA
{hamid.motahari,sven.graupner,sharad.singhal}@hp.com

Abstract. People involved in outsourcing services work through *collaboration, conversations and ad-hoc* activities and often follow guidelines that are described in best practice frameworks. There are two main issues hindering the efficient support of best practice frameworks in outsourcing services: lack of *visibility* into how the work is done that prevents *repeatability*, and conducting best practice processes that are *ad-hoc* and *dynamically defined and refined*. In this paper, we present Business Conversation Manager (BCM) that enables and drives business conversations among people around best practice processes. It supports the dynamic definition and refinement of a process in a collaborative and flexible manner. The ad-hoc processes are backed with a semi-formal process model that maintains the model of interactions and an execution engine. We present the implementation of a prototype BCM and its application in outsourcing services. It supports making processes from best practices among people more transparent, repeatable and traceable.

Keywords: Ad-hoc Business Processes, Collaborations, Outsourcing Services.

1 Introduction

Outsourcing services are offered through organizations of people [1]. The service design and delivery entails several lifecycle phases, in which business artifacts are generated and transferred from one phase to another. People involved in service delivery often work through collaboration, conversations and ad-hoc activities. Currently, it remains hard to provide *visibility* on how a service is created and delivered across all lifecycles and *track* it as information about work results is scattered across many systems such as document repositories, project management systems, and emails. This leads to efficiency issues in the process, timing and economics of service delivery.

There is a push towards facilitating and streamlining processes followed in delivering outsourcing services so that they are delivered in a repeatable, traceable and cost efficient manner. While ad-hoc, the interactions among people in service delivery are not random: they often follow some high-level process flow described in process frameworks (such as ITIL [2]) or in best practice documents in repositories. For instance, the Transformation Services Catalog (TSC) from HP Enterprise Services provides collateral templates (e.g., sales brochure, deployment guide, precedence

B. Benatallah et al. (Eds.): ICWE 2010, LNCS 6189, pp. 468–481, 2010.

diagram, etc) for different lifecycle stages of several services. The templates are used as guidelines for people engaged in advertising, designing and delivering services.

While the main activities conducted in delivering services are often given in templates or defined by project participants, not all of these activities are known in advance nor are they fixed: new activities may be introduced or existing ones in the template not needed in a given engagement. Let us refer to the interaction among a group of people to discuss and work on a business problem as a *business conversation* (or conversation for short). The workplan of a given conversation should be able to be *adapted* in an *ad-hoc* and *flexible* manner, often through brainstorming and *collaboration* between people engaged in the service delivery. The activities within a conversation are often inter-related, so that some may need to be done before others. In addition, different conversations in an engagement are also related. Therefore, there is a need for capturing *relationships* and *dependencies* of activities and conversations. Addressing these problems could make outsourcing service delivery significantly more cost efficient and scalable.

Current workflow management systems do not support such ad-hoc interactions among people as often workflow systems need a well-structured and rigid definition of processes ahead of execution time [15, 16, 21]. Document management systems [18] such as Microsoft Sharepoint are passive repositories of documents and tasks and do not drive interactions among people. Collaboration tools simplify the communication between people and creating and sharing content in a collaborative manner, however, they are unaware of the work context.

This paper presents a system called *business conversation manager* that supports the *guided* interaction of people in a business context in a *flexible, adaptive* and *collaborative* manner. It is capable of establishing business conversations among a number of people (and from a template), drive work between people, and enable them to conduct and adapt the workplan collaboratively (*as they do the work).* The system does not necessarily need a starting template and can be used to start and drive ad-hoc business conversations among an agile-formed group of people. It builds on top of existing document management systems and collaboration systems. We envision that users would use the business conversation manager besides and in integration with productivity tools such as email, MS SharePoint and communication tools.

The conversation manager is backed with an engine that builds and maintains a formal model of the workplan in the form of a task dependency graph that allows nested task modeling, and enables automatic work allocation, progress monitoring and dependency checking through the analysis of the underlying graph model. The conversation manager is based on a minimal number of concepts. The core concept is *business conversation* which is a conceptual container for the interactions among people in order to achieve a business goal. It includes participants (*person* and *role*), *documents* that are consumed (input) or produced (output) in the conversation as well as a *workplan* for achieving the goal. A workplan consists of a set of *tasks* and their *dependency* relationships. Participants of a business conversation can use a number of communication channels (e.g., chat, web-based dialog mechanisms and email) to interact with each other and update the system on the progress of tasks.

The rest of paper is structured as follows. Section 2 presents the definition of concepts introduced in business conversation manager. Section 3 presents the functionalities of business conversation manager from a system point of view. In

Section 4, we present the architecture and implementation of the business conversation manager. We discuss related work in Section 5. Finally, we conclude and present areas of future work in Section 6.

2 Business Conversation Manager: Concepts and Design

2.1 Characteristics of Business Processes for People Services

People services are offered through collaborative work of a group of people. These services have a lifecycle that includes several phases such as inception, design, delivery and operation. Several business processes are involved in each lifecycle phase. These business processes are often described in best practice documents provided either by the vendor corporation or coming from standards such as ITIL (IT Infrastructure Library) [2]. By their very nature, these business processes do not include precise workflows and a strict definition of what activities and in which specific order they have to be executed. Rather, the processes define high level descriptions of activities, some of which are optional in service engagements, as well as a coarse-grained description of constraints on the ordering of activities.

The main characteristics of these processes include: they are (i) *non-structured,* i.e., no strict or formal definition of the process exists, (ii) *non-deterministic*, i.e., the execution order of the activities is not well specified, and the order may change in different engagements, (iii) *adaptive*, i.e., the identified activities may be updated in engagements and at runtime, some may be skipped and new ones added. In general, there is no separation between definition and execution phases of the process, (iv) *templated or ad-hoc*, i.e., there may exist templates for such processes that suggest an initial set of activities, however, such processes may be defined at runtime by the people in an ad-hoc fashion, and (v) *collaborative*, i.e., both the definition as well as execution of the process may be performed in a collaborative manner between involved people.

There are studies showing that capturing definition of processes in a complete and accurate way is often not practical [3] due to sometimes incorrect and incomplete information from various sources. Having adaptive and flexible mechanisms to define processes enables updating the process models at runtime to account for such incompleteness as well as in-accuracy. We refer to such semi-structured processes as *people processes* in the context of designing and delivering people services.

2.2 People Processes: Basic Concepts

We define a minimal number of concepts for people processes informally described in the following. The core concept is *business conversation* (conversation, for short) which is a conceptual container for the interactions among people in order to achieve a business goal. A conversation includes a number of participants that are either real persons or roles, and a set of documents that are consumed (input) or produced (output) in the conversation. Finally, a conversation has a workplan for achieving the goal.

A workplan consists of a set of *tasks*. A task is defined by the set of its input documents and output documents. A task can have a state of “new”, “assigned”, "pooled" (can be picked up by one of participants), "enabled" (ready to be performed), "started", "completed" and "in-active". Tasks can be composed of other tasks in a hierarchical manner. Therefore, tasks can be either "composite" (having subtasks) or atomic. Atomic tasks are executable, i.e., a participant may perform it and it is completed, however, composite tasks are abstract and their completion requires the completion of all its abstract and composite subtasks.

Tasks may have dependencies on one another. We define the *dependency relationship* between two tasks as their data dependency. That is an input document of dependent tasks may be produced by the depended task. The data dependency between tasks is used to draw implicit control flow dependencies between tasks. This frees users from explicit identification of control flows. Indeed, this is an important feature that minimizes the amount of information needed from users to specify the process. It comes from the lesson learned from the observation that most users involved in a project in outsourcing deals do not like the burden of specifying a control flow model (in other words a process model), but rather they are more concerned about their own role and function in the process.

There are two types of task dependencies: "start" and "completed". In the "start" type, the dependent task cannot start until the depended task is started, as well. The "completed" type specifies that the dependent task can start only if the depended task is completed. We define the concept of communication channel to represent a mechanism (email, chat, web-based dialog systems) through which participants of a business conversation interact with each other and the system to perform work and report on the progress of tasks. Similar to tasks, conversations may have dependencies of the same types.

Finally, a participant (a role or a person) can take one of four levels of involvement in a task: "Responsible", "Accountable", "Consulted" and "Informed" (similar to RACI chart [4] for assigning roles and responsibilities in a project plan). The role/person that is responsible should perform the task, while the accountable role/person is ultimately required to make sure of good performance of the task. People with consulted role are those who can be approached for brainstorming or information, and finally, people with informed role have an interest to be informed of the progress and the result of performing the task. Note that not all these roles have to be assigned for a given task, but any task should have a role/person assigned as responsible (which in this case is by accountable as well).

2.3 Towards a Formal Model for People Processes

In this section, we formally define the notion of people processes starting by the concept of "business conversation".

Definition 1 (Business Conversation). A business conversation c is a triple $c = < P, D, W >$ in which P is the set of participants, D is the set of documents manipulated in the conversation and W is the workplan.

The participants $P = \{ p \mid p \in R \vee p \in M \}$ where R is the set of roles and M is the set of people in the enterprise. D is the set of documents that are either consumed or generated in the conversation, and typically stored in document repositories such as MS SharePoint. We define a workplan *W* as follows:

Definition 2 (Workplan). A workplan *W* is a hierarchical directed graph represented with tuple $W = < T, X >$ where *T* is the set of tasks (nodes in *W*), and $X \subseteq T \times T$ is the set of transitions. A task $t \in T$ is defined with the tuple $t = < I, O, s >$ in which $I, O \subseteq D$ is the set of input (outputs) respectively, and $s \in$ *{new, assigned, pooled, enabled, started, completed, in-active}* is its status. A transition *x* is represented as tuple $< t_1, t_2, q > \in X$ meaning that the execution of task t_2 depends on that of t_1 with the dependency type $q \in \{start, completion\}$. If *q="start"* then t_2 is not enabled unless t_1 is started, and if *q="completion"* then t_2 is not enabled until t_1 is completed. A task can be "composite" or "atomic". To a composite task *t* a child workplan W' is associated.

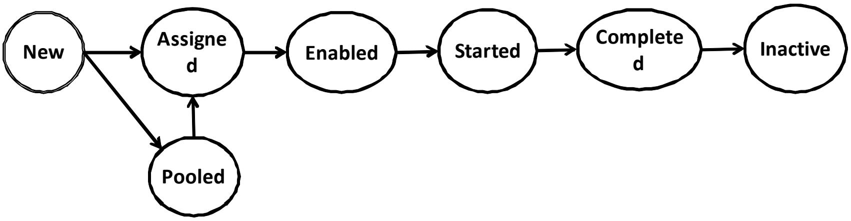


Fig. 1. The lifecycle of an atomic task

Figure 1 shows the lifecycle of an atomic task, i.e., a task that is performed by a participant. A composite task observes only some of these states, i.e., "new", "enabled", "started", "completed" and "in-active". A composite task is "enabled" when all its dependencies are resolved (started or completed respectively to the type of each dependency). It is "started" when at least one of its atomic sub-tasks (in any level) is started, and becomes "completed" when all its sub-tasks are completed. Note that the progression (the status) of a task is updated and for composite tasks can be over-written by participants.

2.4 Execution Semantics of People Processes

Conceptually, we map the workplan model of a business conversation into a colored, hierarchical Petri net [5] and therefore we adopt the execution semantics of such a model for a workplan model. Mapping to the concepts of Petri-net, tokens could be used to represent documents flowing in a business conversation. In a colored Petri net, tokens can take values that are defined by a simple or complex type (historically the value of a token is referred as its color). Workplans allow hierarchical definition to enable having multiple levels of abstraction in the process model. This simplifies the work for different participants such as managers, workers, etc and allows gradual development of the plan with more details concerning specific higher-level tasks. In order to model this aspect of workplans, the semantics of hierarchical Petri nets (HP-Net) [6] are adopted.

Note that this mapping is conceptual in the sense that we incorporate these semantics in our execution engine. Adopting this model allows capturing and reacting to events related to the status update or progression of tasks and document exchanges by humans in a business conversation. Note that the dependency graph of a workplan may form a set of disconnected sub-graphs. Each connected component in this model is related to a set of dependent tasks. When evaluating the dependency model from the execution semantic perspective, we can form a single HCP-net by creating a fake initial place and transitions through which all sub-graphs are connected to form the overall HCP-net representing the whole workplan.

2.5 Templates

In many domains and particularly in services outsourcing context, there are corporate repositories offering templates of documents that are used in the engagements as well as activities that have to be followed in each stage. In some areas such as IT service management, there are standards such as ITIL [2] that describe such processes and activities at a high level. In our previous work (reported in [7]), we have taken an approach to formalize part of these processes as templates that are used to initiate working processes among people based on ITIL processes. Process templates are described as RDF graphs that are refined: more details are added to them as the work progresses among people.

In this paper, we build on our previous work for encoding and formalizing processes in the context of outsourcing engagements as templates. In particular, we have taken collateral templates (high level description) for different lifecycle stages of service delivery from HP TSC catalog. We formalize them as templates and capture the knowledge in those processes as RDF graphs. These templates are made available in the system to be used by participants (e.g., managers) in a business conversation as the initial workplan that could be tailored for a specific engagement.

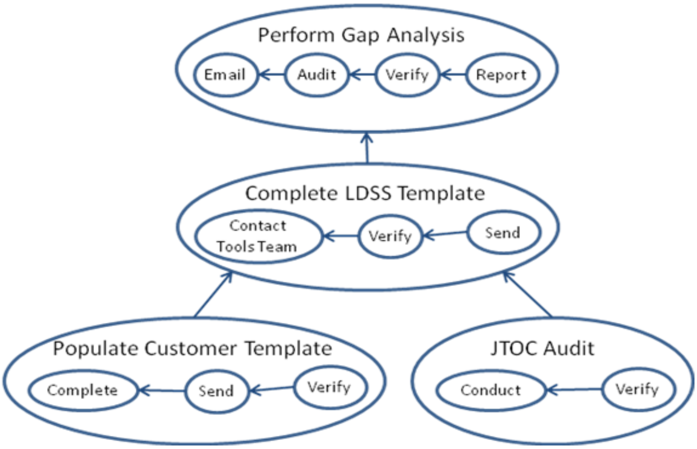


Fig. 2. Part of the hierarchical task dependency model for a workplan

For example, Figure 2 shows part of the (high-level) hierarchical dependency graph for the workplan of an “assessment” business conversation in the context of an incident management process. The ovals represent tasks and the links between them the dependencies between them. For details on how this information is encoded in RDF, please refer to [7].

3 Business Conversation Manager

In this section, we describe our system called “business conversation manager” (BCM) for the establishment, management as well as adaptation of business conversations among people.

3.1 Establishing Conversations and Implicit Dependency Model Management

While business conversations in BCM are supported with formal models, it is very important to note that these models are not exposed to users in BCM. Indeed, BCM creates and maintains the task dependency model of the workplan in the backend automatically. Therefore, participants do not work with the workplan model explicitly. Participants are concerned with the definition and progression of individual tasks and if a given task maintains dependency relationships with others. This is consistent with the nature of the job that they are doing. The intention is to introduce the least amount of overhead of the automation tool for participants.

One of the main features of BCM is that it allows gradual and level-wise definition of the model by different participants with different expertise and levels of knowledge about the process. Therefore, there is no need for apriori definition of the whole process. The participants can start with a high-level and incomplete definition of the tasks. The hierarchical feature of the model allows the participants to refine abstract tasks into more finer grained ones in a level-wise fashion by people who are responsible for the next level of detail. Another important feature of this system is that there is no separation between the definition and the execution of the model. The process is considered executing right from the time that the highest level tasks are defined. Therefore, the runtime and design of the process are interleaved.

In identifying the dependencies between the tasks, BCM does not mandate identifying which specific input document(s) of the dependent task depends on that (those) of a depended task. This is to provide more flexibility for participants. Instead, BCM establishes this relationship after the participants of the dependent tasks pull and use documents produced (or manipulated) during the performance of the depended task. Internally, this is managed in BCM by tagging documents as “input”, “manipulated” or “output” for documents that are used or produced in the context of a task. However, the dependency between two tasks must be explicitly specified. It should be noted that dependency is specifiable between tasks at the same level in the hierarchy.

3.2 Management and Adaptation of Conversation Workplans

People processes are often ad-hoc and require adaptation, especially if the workplan of a conversation is defined according to a template. BCM provides the following facilities for adapting definition of processes.

Adding and updating a task: This method allows the definition of a task in a workplan. For each task, properties such as *start-date*, *due-date*, *end-date*, *status*, *actorIDs*, *documents*, *dependsOn*, *parent, type* and *method* can be specified. ActorIDs provides the list of the participants involved with their role(s) in accomplishing the task. A task can be one of atomic or composite types, and can be accomplished using one of two methods: "human" or "automated". By default tasks are human-offered. If the method of a task is "automated", the API interface details are needed and an adapter to call the Web service at the runtime is generated. "dependsOn" takes the list of other tasks that the current task depends on. A task may not have any dependencies.

The parent property takes as value the workplan or another composite task if it is its immediate child. A composite task is a place-holder for a set of other tasks that are not yet known. The update method enables updating various properties of a task at runtime.

Remove a task: This method introduces a "consensus-based removal" approach for removing tasks in a collaborative manner between conversation participants. When the removal request for a task is made by a participant, BCM triggers an event so that a notification message is sent to all participants with "accountable" and "responsible" roles from the list of actorIDs of this task and all the tasks that depend on it directly to inform them of the request for its removal. The message asks them to react if they object to the removal of the task. If nobody objects to its removal within a pre-specified timeline, it is removed from the workplan. Upon removal, the dependency list of tasks that depends on the removed task is updated. The status of the removed task is set to "in-active" meaning that it is not part of on-going conversation. It is not shown in the list of tasks of the workplan but is maintained in the back-end repository for historical reasons, as well, it can be restored to its last state before removal if needed by the participant who removed it.

Join a business conversation: This method allows a person in the enterprise to request to join a team involved in a business conversation. The membership request is sent to the participants with "accountable" role in the business conversation for approval (approval of one suffices). BCM creates a workspace for the new participants in which he can review the history of the business conversation progression.

Add/remove a participant: This method allows the participants in a business conversation to invite new people to join the business conversation, and allows participants with "accountable" role to remove a participant from the conversation. In case the person is removed, the workspace of the person gets the status of "in-active" and maintained in the repository for historical reasons.

4 Architecture and Implementation

We describe the architecture and implementation of prototype business conversation manager in the following.

4.1 Architecture

The business conversation manager is offered as a service that exposes a set of APIs. There are three main categories of components in BCM: those supporting the definition and execution of workplans, and those related to communication channels, and the client-side portal. The system's architecture is shown in Figure 3. It has the following components:

The service APIs: The APIs expose the functionality of business conversation manager as a Web service.

Business activity portal: This is a Web-based software component that implements the user interface and supports ad-hoc user interactions to define, view and update the workplan/activity details.

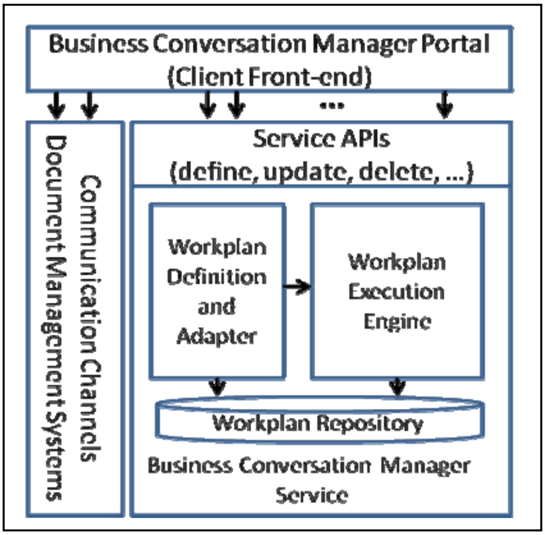

Business Conversation Manager Portal
(Client Front-end)
...
Communication Channels
Document Management Systems
Service APIs
(define, update, delete, ...)
Workplan Definition and Adapter
Workplan Execution Engine
Workplan Repository
Business Conversation Manager Service

Fig. 3. The architecture of business conversation manager as a service

Workplan repository: This is the component that stores information about workplan templates from best practices for process frameworks such as ITIL as well as people service catalogs such as TSC. It also stores the information of on-going conversations.

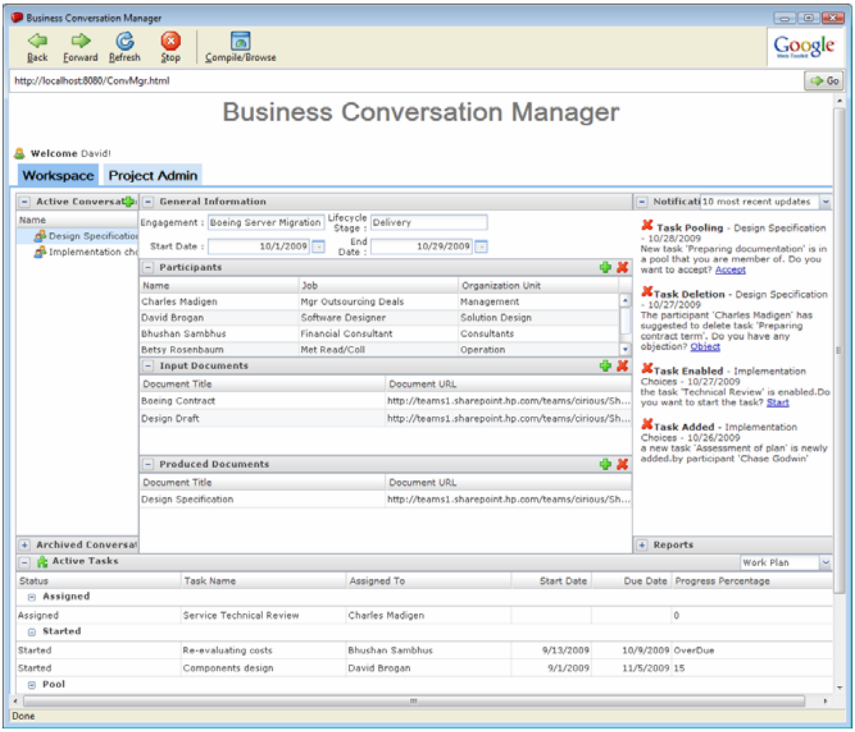
Workplan definition and adapter: This component enables definition of workplans and updating their definitions through methods such as adding/removing tasks, as well as updating the workplans definition in an ongoing conversation (at runtime) initiated by events triggered from the portal.

Workplan execution engine: This component supports the execution of the workplan of a business conversation and the execution semantics introduced in Section 2.4. This component coordinates the flow of tasks among people based on task dependencies.

Communication channels: The communication channels are those that enable the communications between participants such as chat and email, and document management systems such as Microsoft SharePoint that enable storing information used in the context of BCM.

4.2 Implementation and Use

We have implemented the components for the prototype business conversation manager service in Java. The client-side portal has been implemented using Google Web Toolkit (GWT) (code.google.com/webtoolkit/).



Engagement : Boeing Server Migration
Lifecycle Stage : Delivery
Start Date : 10/1/2009
End Date : 10/29/2009

Participants

Name	Job	Organization Unit
Charles Madigen	Mgr Outsourcing Deals	Management
David Brogan	Software Designer	Solution Design
Bhushan Sambhus	Financial Consultant	Consultants
Betsy Rosenbaum	Met Read/Coll	Operation

Input Documents

Document Title	Document URL
Boeing Contract	http://teams1.sharepoint.hp.com/teams/cirious/Sh...
Design Draft	http://teams1.sharepoint.hp.com/teams/cirious/Sh...

Produced Documents

Document Title	Document URL
Design Specification	http://teams1.sharepoint.hp.com/teams/cirious/Sh...

Task Pooling - Design Specification - 10/28/2009
New task 'Preparing documentation' is in a pool that you are member of. Do you want to accept? Accept

Task Deletion - Design Specification - 10/27/2009
The participant 'Charles Madigen' has suggested to delete task 'Preparing contract term'. Do you have any objection? Object

Task Enabled - Implementation Choices - 10/27/2009
the task 'Technical Review' is enabled.Do you want to start the task? Start

Task Added - Implementation Choices - 10/26/2009
a new task 'Assessment of plan' is newly added.by participant 'Chase Godwin'

Status	Task Name	Assigned To	Start Date	Due Date	Progress Percentage
Assigned					
Assigned	Service Technical Review	Charles Madigen			0
Started					
Started	Re-evaluating costs	Bhushan Sambhus	9/13/2009	10/9/2009	OverDue
Started	Components design	David Brogan	9/1/2009	11/5/2009	15
Pool					

Fig. 4. The screenshot of the frontend of Business Conversation Manager prototype

In the prototype BCM tool, the participants of a business conversation are provided with a workspace within which they can manage the conversations that they are participating in. The workspace enables participants to monitor the status and progress of tasks as well as update the workplan. The workspace also provides highlights such as tasks that are (over-) due.

Figure 4 shows the screenshot of the workspace for a user called "David". In the left panel it shows the list of conversations (active and archived). The middle panel shows the information about a given conversation including which service engagement and lifecycle stage it belongs to, as well as the start and end date of the conversation. The list of participants and the input and output documents are also part of the middle panel. The bottom panel shows the list of tasks in the conversation. David has the option of limiting the task view to the ones allocated to him or all tasks in the workplan of the conversation. The right panel shows updates from all conversations in which David is participating (e.g., enablement of a particular task because its dependent tasks are completed, and new tasks that are in the pool and David is one of the nominated people to take it). The notification messages are visible in the workspace and can be optionally sent by email to participants, as well.

When an outsourcing manager comes to the business activity portal, he/she can create a new business conversation using a wizard. The system asks the manager whether the new conversation is defined based on a pre-existing template, in which case she can select from the list of already stored templates. Using the wizard the people involved in the conversation are also invited. A business conversation is considered active from the moment that it is defined.

5 Related Work

A novel contribution of the proposed business conversation manager is that is builds a bridge among structured and rigid process-centric systems, completely ad-hoc and unstructured conversations between people, and the use of productivity tools in work environments to facilitate the efficiency of conducting best practice processes such as ITIL. Therefore, related work spans across the areas of best practice frameworks; business processes; knowledge and document management systems; and collaboration environments.

Best practice frameworks. Processes from best practice frameworks such as ITIL [2] and eTOM [9] have been often described as textual descriptions. There have been efforts to support people in formalizing and following best practice frameworks. For example, some approaches propose using semantic Wiki [8] and also ontologies [10] to represent processes. However, these efforts only look at this representation as a knowledge base rather than actionable processes.

Business processes. Business process modeling and management tools such ARIS, SAP or standards such as BPEL4People [11] allow definition of well-defined and structured business processes. However, many processes in the enterprise, especially in the context of outsourcing services, involve human interactions that are semi-structured and ad-hoc. In the same line of work, [13] formalizes ITIL processes as precise business process models expressed in process modeling languages such as BPMN.

Definition of ad-hoc and flexible processes has also gained attention recently. For instance, Caramba [14] enables definition of ad-hoc processes for virtual teams. In this work, the process needs to be explicitly defined by the team members using graphical process modeling tools prior to its execution. However, in the context of best practices: (i) processes are not well-specified to enable formal definitions directly, (ii) process users are knowledge workers that are only familiar with productivity tools; they find it difficult to work with formal process models, And (iii) in our approach the process models are used in the backend to support the user but not explicitly exposed to users.

Change management for adaptive and dynamic workflows is studied in the literature [15, 16, 21]. Adaptable workflows address changes that affect the workflow definition (structure, type, etc), while dynamic workflows are concerned with changes to runtime instances of workflows. ADEPTflex [15] enables operators to (manually) change the running instances of a statically defined workflow, while ensuring its correctness. In [16] a high level definition of a workflow is assumed and the concept of *worklet* is introduced to assign concrete activities from a library to realize tasks in the predefined, high level workflow. We do not assume availability of a library of tasks that could be used to realize high level tasks, as most often changes to tasks in process templates are specific to the project context. Our proposed model enabled both static and dynamic changes. The changes to the ad-hoc processes and instances are made as the participants work.

Knowledge and document management systems. Many existing requirement capture and management tools and business process analysis tools such as ProVison simplify the tasks of gathering, documenting, tracking and managing requirements and process definitions in an enterprise [17]. Typically these tools help document requirements and processes, and in some instances simulate the impact of changes. They are geared towards implementing and executing projects and processes in IT systems not among people.

Collaboration approaches and tools. The proposed system differs from wiki-based collaboration systems (e.g. Semantic Media Wiki [8]) as Wikis provide a passive knowledge base. DOMINO [23] and OpenWater [22] are examples of early efforts to support more flexible and cooperative processes in organizations. Unified activity management [12] is another thread of work which aim at providing an integrated work environment for all activities of a person across various productivity tools and organizing them and supporting the collaborations of people around activities. Business Conversation Manager takes a step forward by combining informal interactions about the process with the semi-structured definition of the processes while supports ad-hoc (best practice) processes which was not the focus in earlier works. Recently, there has been a rapid growth in social collaboration tools and techniques such as Google Wave (wave.google.com) as well as Web 2.0 types of collaboration techniques. These tools and techniques are complementary to our work, and in our platform they play a role as communication channels between conversation participants. There has been also recently some works that allow collaborative definition of processes, e.g., based on Google Wave platform (e.g. Gravity [19] from SAP Research that allows collaborative process modeling) or Workflow-on-Wave (WoW) [20]. These tools aim at defining the business process model prior to their execution (in a collaborative manner), however, we do not assume the existence of a business process model (other than templates) ahead of execution time and the process definition emerges and becomes updated on the fly (while people work) in a flexible and collaborative manner among people.

6 Conclusion and Future Work

In this paper, we have presented an enterprise-grade system for establishing, managing and conducting business conversations to support ad-hoc people processes from best practice frameworks such as ITIL. We have implemented a prototype system in the context of supporting people processes for delivering outsourcing services. Our main aim has been to reduce the burden of using the system as well as its overhead for knowledge workers in terms of amount of process-related information that they need to learn. At the same time, we have designed the system so that it is backed with formal modeling and execution semantics of processes and makes uses of them in a transparent manner for users.

The process model introduced in the paper based on dependency model offers a “lightweight” process modeling approach that supports collaborative definition and adaptation of the process compared to hard-coded or rigid processes that are hard to change after the process have been started. The business conversation manager (BCM) introduced in this paper builds on top of and allows users to utilize the existing systems that they are familiar with in their daily jobs such as MS SharePoint, and email. We introduce a minimum amount of abstractions in a simple and innovative manner to simplify the job of people in defining and managing people processes.

In terms of future work, we are planning to provide a catalog of conversations to users so that participants can find other related conversations within the project to that of their own conversations so that inter-conversation dependencies could be managed more efficiently. We are currently incorporating the capability to store the workplan of active conversations that are near conclusion as templates for future reuse. We are planning also to experimentally validate the system by having people use it in the context of service engagements.

Acknowledgement. Authors would like to thank Sujoy Basu and Susan Spence from HP Labs for their feedbacks and comments on earlier drafts of this paper.

References

1. Lee, J., Huynh, M.Q., Kwok, R.C., Pi, S.: IT outsourcing evolution—: past, present, and future. Commun. ACM 46(5), 84–89 (2003)
2. Hendriks, L., Carr, M.: ITIL: Best Practice in IT Service Management. In: Van Bon, J. (Hrsg.) The Guide to IT Service Management, London u. a, Band 1, pp. 131–150 (2002)
3. Hobson, S., Patil, S.: Public Disclosure versus Private Practice: Challenges in Business Process Management. In: Workshop on SOA, Globalization, People, & Work (SG-PAW 2009), Sweden (2009)
4. Project Management Institute, The Project Management Body of Knowledge, PMBOK (2000)
5. Jensen, K.: An Introduction to the Practical Use of Coloured Petri Nets. In: Reisig, W., Rozenberg, G. (eds.) APN 1998. LNCS, vol. 1492, pp. 237–292. Springer, Heidelberg (1998)
6. Choo, Y.: Hierarchical Nets: A Structured Petri Net Approach to Concurrency, Technical Report CaltechCSTR:1982.5044-tr-82, California Institute of Technology (1982)
7. Graupner, S., Motahari-Nezhad, H.R., Singhal, S., Basu, S.: Making Process from Best practices Frameworks Actionable. In: DDBP 2009: Second International Workshop on Dynamic and Declarative Business Processes, Auckland, New Zealand, September 1 (2009)
8. University of Karlsruhe, Semantic Media Wiki, `http://semantic-mediawiki.org`
9. TeleManagement Forum, Enhanced Telecom Operations Map (eTOM) – The Business Process Framework, `http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/1647/Home.html`
10. Shangguan, Z., Gao, Z., Zhu, K.: Ontology-Based Process Modeling Using eTOM and ITIL. In: CONFENIS, vol. (2), pp. 1001–1010 (2007)
11. WS-BPEL Extensions for People—BPEL4People: A Joint White Paper by IBM and SAP (July 2005), `http://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/cfab6fdd-0501-0010-bc82-f5c2414080ed`

12. Moran, T.P., Cozzi, A., Farrell, S.P.: Unified activity management: supporting people in e-business. ACM Commun. 48(12), 67–70 (2005)
13. Orbus Software, The iServer ITIL Solution, http://www.orbussoftware.com/business-process-analysis/products/itil-solution/
14. Dustdar, S.: Caramba — A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams. Distrib. Parallel Databases 15(1), 45–66 (2004)
15. Reichert, M., Dadam, P.: ADEPT flex Supporting Dynamic Changes of Workflows Without Loosing Control. Journal of Intelligent Information Systems 10(2), 93–129 (1998)
16. Adams, M., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Implementing dynamic flexibility in workflows using worklets. Report BPM-06-06. BPMCenter.org (2006)
17. Volere. List of Requirement Managemnt Tools, http://www.volere.co.uk/tools.htm
18. HCi Journal, List of Document Management Software, http://www.hci.com.au/hcisite3/journal/Listofdocumentmanagementsoftware.htm
19. Dreiling, A.: Gravity – Collaborative Business Process Modelling within Google Wave, SAP Research (September 2009)
20. Itensil, Workflow-on-Wave (WoW), http://itensil.com
21. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software Through Process Technology. WileyBlackwell (2005)
22. Whittingham, K., Stolze, M., Ludwig, H.: The OpenWater Project - A substrate for process knowledge management tools, AAAI Technical Report SS-00-03 (2000)
23. Kreifelts, T., Hinrichs, E., Klein, K., Seuffert, P., Woetzel, G.: Experiences with the DOMINO office procedure system. In: Second Conference on European Conference on Computer-Supported Cooperative Work, The Netherlands, September 25 - 27 (1991)