

WS-BPEL Extension for People – BPEL4People

A Joint White Paper by IBM and SAP
July 2005

Authors (alphabetically)

Matthias Kloppmann, IBM (matthias-kloppmann@de.ibm.com)

Dieter Koenig, IBM (dieterkoenig@de.ibm.com)

Frank Leymann, IBM (ley1@de.ibm.com)

Gerhard Pfau, IBM (gpfau@de.ibm.com)

Alan Rickayzen, SAP (alan.rickayzen@sap.com)

Claus von Riegen, SAP (claus.von.riegen@sap.com)

Patrick Schmidt, SAP (patrick.schmidt@sap.com)

Ivana Trickovic, SAP (ivana.trickovic@sap.com)

Abstract

Human user interactions are currently not covered by the Web Services Business Processes Execution Language (WS-BPEL), which is primarily designed to support automated business processes based on Web services. In practice, however, many business process scenarios require user interaction. This paper describes scenarios where users are involved in business processes, and defines appropriate extensions to WS-BPEL to address these.

Copyright Notice

© Copyright SAP AG and International Business Machines Corp 2005. All rights reserved.

No part of this document may be reproduced or transmitted in any form without written permission from SAP AG (“SAP”) and International Business Machines Corporation (“IBM”).

This is a preliminary document and may be changed substantially over time. The information contained in this document represents the current view of IBM and SAP on the issues discussed as of the date of publication and should not be interpreted to be a commitment on the part of IBM and SAP. All data as well as any statements

regarding future direction and intent are subject to change and withdrawal without notice. This information could include technical inaccuracies or typographical errors.

The presentation, distribution or other dissemination of the information contained in this document is not a license, either express or implied, to any intellectual property owned or controlled by IBM or SAP and/or any other third party. IBM, SAP and/or any other third party may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give you any license to IBM's or SAP's or any other third party's patents, trademarks, copyrights, or other intellectual property.

The information provided in this document is distributed "AS IS" AND WITH ALL FAULTS, without any warranty, express or implied. IBM and SAP EXPRESSLY DISCLAIM ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR TITLE. IBM and SAP shall have no responsibility to update this information. IN NO EVENT WILL IBM OR SAP BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

SAP is a registered trademark of SAP AG.

IBM is a registered trademark of International Business Machines Corporation.

Other company, product, or service names may be trademarks or service marks of others.

Contents

1	Introduction	4
1.1	Problem	4
1.2	Goal	4
2	Scenarios	5
2.1	People Activities	5
2.2	People Initiating Processes	5
2.3	People Managing Long-Running Processes	6
2.4	Transition between Human and Automatic Services	6
2.5	Advanced Interaction Patterns.....	6
2.5.1	4-Eyes Principle.....	7
2.5.2	Escalation	7
2.5.3	Nominations.....	7
2.5.4	Chained Execution	7
3	Concept.....	8
4	Features.....	10
4.1	People Integration.....	10
4.1.1	Generic Human Roles	10
4.1.2	People Links	11
4.1.3	People Resolution	11
4.2	People Activities	12
4.3	Tasks	13
4.3.1	Properties	13
4.3.2	Operations for Client Applications	13
4.3.3	States	14
4.3.4	Inline Tasks and Standalone Tasks.....	14
4.4	Context	15
4.5	User Interface	16
4.6	Services Implemented by People	16
4.6.1	Motivation from a Business Process Perspective.....	16
4.6.2	Motivation from a Web Services Perspective	17
4.6.3	Characteristics.....	17
5	Summary.....	17
6	Glossary.....	17
7	References.....	18

1 Introduction

1.1 Problem

The ongoing work on the Web Services Business Process Execution Language, version 2.0 (WS-BPEL 2.0, or BPEL for short) focuses on two parts. One is the model for executable business processes that is used to specify automated business processes that orchestrate activities of multiple Web services, and which may be interpreted and executed by compliant engines. The other part is the observable behavior of Web services. The language encompasses features needed to describe complex process control flows, including error handling and compensation behavior. Those constructs are seen and used in multiple process models and are needed to build complex processes, which can be executed by underlying software. However, business processes go beyond the orchestration of activities exposed as Web services. In addition to the orchestration of Web services, the process definition typically incorporates people as an additional possible type of participant, since they can also take part in business processes and can influence the execution of processes. The aspect of how people interact with business processes must be properly modeled.

1.2 Goal

The BPEL specification focuses on business processes, the activities of which are assumed to be interactions with Web services with no additional prerequisite behavior. But the spectrum of activities that make up general purpose business processes is much broader. People often participate in the execution of business processes introducing new aspects, such as human interaction patterns. Workflow tools already cater for the orchestration of user interactions.

User interactions range from simple scenarios, such as manual approval, to complex scenarios where data is entered by the user. Imagine a bank's personal loan process. This process is made available on the internet site of the bank using a web interface. Customers can use this interface to enter the data for their loan approval request and to start the approval process. The process performs some checks, and eventually informs the customer whether his or her personal loan request has been approved or rejected. Processing is often automatic and does not require any human involvement. However, there are cases that require bank staff to be involved. An example of such a case is if the online check of a customer's creditworthiness returns an ambiguous result. In this case, instead of declining the request automatically, a bank clerk could check the request and determine whether to approve or decline it. Another example would be if a request exceeds the amount of money that can be approved automatically. In this case, a manual approval step is required, in which a member of the "approvers" group either approves or declines the request.

User interactions in business processes are not limited to approval steps. They also may involve data. An example of a user interaction that involves data is when an e-mail from an employer is manually attached to the process instance, or when the summary of an interview with an applicant is keyed into the process via a simple form or custom-built application.

To support a broad range of scenarios that involve people within business processes, a BPEL extension is required.

BPEL4People is defined in a way that it is layered on top of the BPEL language so that its features can be composed with the BPEL core features whenever needed. We envisage that additional BPEL extensions may be introduced which may use the BPEL4People extension introduced here.

This paper discusses scenarios and outlines features that need to be supported by BPEL4People. It is organized as follows: Section 2 discusses scenarios. The underlying concepts are explained in section 3. Section 4 discusses the features required to extend BPEL to support user interactions. It also considers the aspects of standalone human tasks that do not necessarily belong to BPEL processes. A summary of the work on BPEL4People is provided in section 5.

2 Scenarios

This section describes various scenarios capturing human perspective in business processes to illustrate the scope of BPEL4People.

2.1 People Activities

People can be involved in business processes as a special kind of implementation of an activity – a communication step which may be called people activity. From the user's perspective, the people activity is a task that is assigned to a user and requires the user to perform some action. Users are made aware of this assignment by a distribution mechanism that adds a work item to their task list. A work item denotes an assignment of a task to a user. Work items specify that a certain user is eligible to perform specific actions on a task.

The task representing the people activity appears in the user's task list, together with, for example, to-dos from Lotus Notes, tasks from Microsoft Exchange, or collaboration tasks from SAP. As well as being involved in tasks, people may be involved in a more simple communication step called notification, which is simply the transmission of information to an interested party. The most important difference between tasks and notifications is that a task holds up the process until it is completed.

2.2 People Initiating Processes

In addition to involving people as the implementation of a BPEL activity, there are scenarios where it is desirable to define which people are eligible to start a certain business process. An example of this is the travel approval process of a company's employee self-service portal. All people belonging to the group of regular employees are entitled to use the travel approval process directly, but part-time employees and work students have to go through their respective managers. In practice, the attributes associated with people, such as their organizational assignment, often play a significant role in the direction that the process takes, or in the assignment of people to work items generated in the process (for example, determining the responsible manager). To support such scenarios, it is useful to associate people with the BPEL constructs that are used to initiate a business process.

2.3 People Managing Long-Running Processes

During the lifetime of a long-running business process, conditions that require human involvement can occur. An example of this would be if a process is stuck because no one has been assigned to perform a particular task. Another example would be if a process waits for input from human participants or Web services, and the input must be collected within a certain number of hours. If the timeout occurs, a user must be notified to decide how the process should proceed. There might be several possible solutions: continue process using data that has been received so far; extend the deadline and wait for missing data; or feed the process with the missing data and proceed with the process execution. Based on BPEL's semantics today, the consequence would require undoing parts of the business process. But if we assume that the business process has already run for multiple days, invoking partner operations, collecting results, and so on, compensation may not be desirable, since this wastes resources and efforts already spent. In this case, it is desirable to have a business administrator associated with the process, who can decide what corrective action is necessary.

Business administrators are people who use their business knowledge to ensure that the processes run smoothly with the best possible outcome. To support such a scenario it is required to define people assignments for the BPEL process that define which people are allowed to act as business administrators of the process.

From the perspective of the business administrator, a task is defined and created. This task allows the business administrator to interact with a certain process instance by retrieving process state, and by performing various operations on it, such as entering missing data, resolving failed people resolution, prioritizing, suspending or slowing down the process instance execution in order to synchronize with other processes, or even canceling the execution.

Process monitoring and technical administrators who are in charge of the process monitoring and the underlying software infrastructure (for example, runtime engine) are out of scope of BPEL4People.

2.4 Transition between Human and Automatic Services

Consider a service that takes place out-of-sight of the initiating process. In many circumstances, it may be immaterial as to whether the service is performed with or without user interaction, for example, a document translation service. In addition to the requirement to support the execution of human tasks, the transition between human and automatic services (and vice versa), must be non-disruptive.

2.5 Advanced Interaction Patterns

In addition to the simple task selection and execution actions, there are more complex patterns in the way humans interact with the process instances, and these need to be handled by BPEL4People. The interaction patterns described below are all typical when user interaction is required, so BPEL4People needs to support them.

2.5.1 4-Eyes Principle

The 4-eyes principle, often referred to as “separation of duties,” is a common scenario in the medical or banking environment when a decision (such as granting a loan) is made by two or more people independently of one another. This is often done for security reasons, so in extreme cases the users may not even be allowed to know who else is involved to rule out any collusion between them. In other cases, it is simply enough to obtain a second opinion. Being able to exclude users from performing an activity needs to be supported by BPEL4People.

2.5.2 Escalation

Tasks represent work to be performed by humans. Tasks can be modeled to express the expectation that the task is to be started or completed within a certain time frame. If a task is not progressing as expected, for example, if a person working on such a task suddenly becomes ill, or is just overwhelmed by work, an escalation mechanism is required.

Escalation takes place if a task does not meet its modeled time constraints. If this occurs, a notification is sent to one or several people specified as escalation recipients using a people assignment definition. Notifications are sent to escalation recipients by e-mail, SMS, or instant message. Escalation recipients can then decide what they need to do to get the task back on track.

Escalations can also be tasks themselves and escalations can be chained. The first escalation could go to the first line-manager, the second escalation, which is only triggered if the task is still behind schedule, would go to the second line-manager, and so on. However, once the main task is completed, the escalation chain must terminate immediately to avoid redundant effort. So it is important to be able to synchronize between the original task, its modeled deadlines, and the escalation procedure.

2.5.3 Nominations

Sometimes it is not clear who should perform the task in hand. This is the case when schedule or workload constraints dictate that a supervisor manually assigns tasks to colleagues. In other cases, a supervisor will nominate ownership of a task to the colleague with the expertise that best matches the task. In the most complex scenario, this task delegation may even be a collaborative decision among peers.

2.5.4 Chained Execution

Chained execution is a process fragment where a sequence of steps is executed by the same person. This is sometimes known in advance, but can sometimes only be determined during execution. It would be clumsy to force the user to return to the task list and refresh it after executing each task, so a mechanism to ‘short-circuit’ the user interaction with the task list is desirable. The person working on it performs subsequent actions of the form “complete and claim next task” within the scope of one and the same process instance. This gives the illusion of a wizard-style interaction like typical Web-based order processes when performed by a single person, who in most cases is also the process initiator. In addition to the fact that the people assignments in such a process refer to the process initiator, chained execution has an interesting usage pattern: Associating UI definitions with the

activities in such a chained-execution workflow gives the illusion of using such a workflow for the dialog control of a UI. But when several users are involved in the process, it executes without any loss in process logic using the familiar task list interaction instead of the wizard illusion.

3 Concept

This section explains the design decisions and introduces the basic concepts used throughout this paper. This is done by using a simple example process: The “Brochure Creation” process defines the steps needed to create a brochure. It consists of the “Create” step, in which the brochure is created, then the “Approve” step, in which the brochure is either approved or not approved, and then an additional “Revise” step if the brochure has not been approved in the first place, and thus requires more work. The tasks required to create a brochure are performed by humans. Consequently, the “Brochure Creation” process contains activities that are performed by humans. Bear in mind that this process is for illustrative purposes only. In reality, the process would probably have additional steps that we will ignore here to keep things simple.

The left-hand side of Figure 1 shows the process modeling the brochure creation. The right-hand side shows the organizational directory of the HR system of the company that creates the brochure. The organizational directory contains the organizational structure of the company.

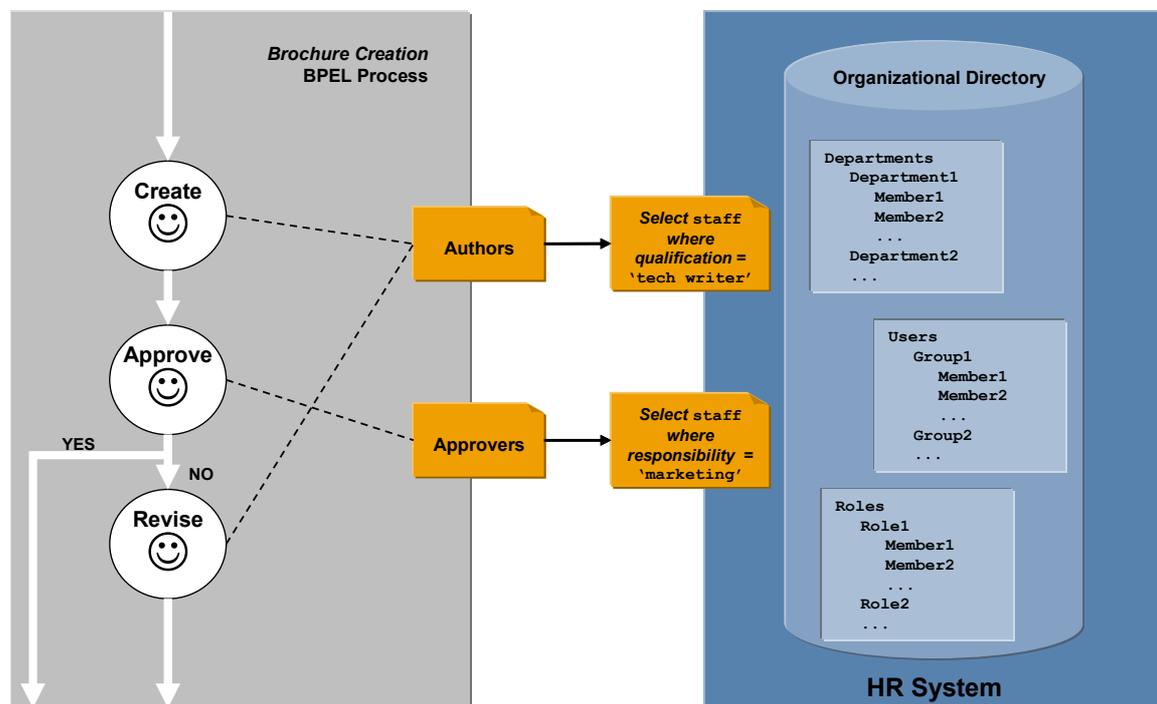


Figure 1. Illustration of Basic Concepts

The process is initiated by triggering the first step in the process, which is the “Create” step, by the process initiator. In this example the initiator happens to be the stakeholder, too. The stakeholder is a person with an interest in creating and completing a brochure, and who may track the progress of this particular process

instance. The “Create” step is performed by the author of the brochure. The author is the person from the “Authors” group who has chosen to perform the “Create” task. The “Authors” group is bound to a query against an organizational directory, which selects all the people qualified as a “technical writer”. When the process reaches the “Create” activity, this query is executed. As a result, the list of users with the “technical writer” qualification is returned, and the “Create” activity is assigned to them. The query uses parameters that were specified when the process was modeled.

The next step is the “Approve” step. It is similar to the “Create” step, except that another group of people is eligible to approve the brochure. As outlined below, the group “Approvers” is bound to a query that returns people with marketing responsibilities. In addition, if the brochure has a financial nature, and, for example, discloses results of recent acquisitions that affect the shareholder value of the company, then the approver must have passed the necessary training to fulfill the requirements (such as Sarbanes-Oxley). This is done so that shareholders are not inadvertently misled. Electronically describing the qualifications or certification required to perform this approval activity is out of scope for BPEL4People, but provision is made to include textual descriptions of such requirements to make the BPEL process that encompass user interactions portable.

The third step is the “Revise” step. It is only processed if the brochure has not been approved in the first place. In the example above, the “Revise” step is performed by the same group of people that did the original authoring. However, it does not prescribe that it must be necessarily performed by the same author. In a real life example, we might see a process that at first gives the original author a chance to work on the findings that led to the rejection of the document. While this is possible to achieve using the concepts described in this paper, we choose instead to use the simpler scenario for this example. This allows us to illustrate that the same group of people (“Authors”) that is bound to the same people query can be used multiple times in a process.

The following design decisions have been made. The BPEL process definition has the *people activity* as a new activity type that incorporates user interactions. People activities are implemented by *tasks* performed by users. Particular users who perform a task may be specified at design time, at deployment time, or at runtime. In the case of the latter, organizational directories are usually used to determine the users who are associated with a particular task. Organizational directories define collections of people, often structuring them into groups, departments, and so on. Organizational structures typically exist independently of business processes, and are managed in separate infrastructures such as LDAP. Therefore the definition and description of organizational structures, and the methods used to access them are out of the scope of this paper.

Although the organizational directory with the underlying organization model is out of scope of the business process definition, the link between the people activities and organizational directory is needed and is provided using a *people link*. A people link represents the group of people who are associated with a people activity. A people link may be reused across multiple people activities within the same process. It can be resolved as early as at design time or as late as at runtime.

To determine the actual set of people involved in the process, people links are bound to *people queries*. Note that the pseudo query used in the figure above is just an example. An actual query would use an LDAP filter, a SQL query, an XQuery, or a Java method, for example. The people queries and binding of people links are out of scope of BPEL4People. From the perspective of this paper, the only thing that is important is that there is an organizational directory that allows information to be retrieved about the people in a company by using queries or an alternative access method. The kind of organizational directory used, its structure, and the query mechanisms used are irrelevant.

4 Features

This section describes the features that need to be supported by BPEL4People.

4.1 People Integration

The relationship between people and processes is complex so this section focuses on:

- The different ways that people interact with processes ('generic human roles')
- How the process identifies the people to interact with ('people links')

4.1.1 Generic Human Roles

People activities and processes have well-defined generic human roles defining what a person or a group of people resulting from a people query can do with them. This perception is independent of the actual business scenario the activity is used in, and is orthogonal to the organizational roles mentioned earlier. Based on the scenarios in the previous sections, the following process-specific semantic roles can be observed.

The *process initiator* is the person who actually creates an instance of the process.

The *process stakeholder* is a person who can influence the progress of a process instance, for example, by adding ad-hoc attachments, forwarding a work item, or simply observing the progress of the process instance.

People activities have *potential owners*. Potential owners are entitled to claim and complete the activity. A potential owner becomes the actual *owner* of an activity by explicitly claiming it.

Processes may define a *business administrator* for a business process. Business administrators are allowed to perform administrative actions on the business process, such as resolving missed deadlines. The business administrator, in contrast to the process stakeholder, has an interest in all process instances of a particular process type, and not just one. For example, the stakeholder is keen on seeing that the underlying process involving the brochure he or she commissioned is executed quickly, and that the brochure is printed, whereas the business administrator simply ensures that all brochure requests are processed efficiently, irrespective of whether the request is accepted or not.

4.1.2 People Links

Within a process or a people activity, certain groups of users are important from a business standpoint. For example, a loan approval process requires a group of “loan approvers”. People links are used to represent the different groups of people who participate in the execution of the process. The group of people associated with a generic human role is determined by a people link.

In order to determine the actual individuals associated with a people link, a query against an organizational directory is used. This query is bound to the people link. For example, the generic human role *process owner* could be qualified by the people link *marketing manager*, which is bound to the query “select head of department, where department name is 'marketing’”. Typically, people queries are defined using descriptions based on an organizational model. A very simple organizational model would assume that there are persons and groups. It is out of the scope of BPEL4People to define a standard for organizational models.

It is possible to parameterize people links to pass data from the process instance to the people query. The above example could be refined by taking the geographical location into account. If the company has offices in different countries, the request for approval is sent to the regional marketing manager.

There is a need to specify the parameters and their binding to concrete data, either at design time or at deployment time. The 4-eyes principle is one of the scenarios where needed parameters are specified at modeling time. The owner of the first approval activity is passed as a parameter to the people link of the second approval activity so that this owner can be excluded.

A set of parameters may need to be refined at deployment time. For example, a standardization consortium may specify processes that can be applied to various industries and across multiple companies of different sizes. However, companies spanning the globe may want to customize the standardized process definition. They may refine the parameters to be more specific about people who should perform people activities. In the above example, the geographical location could be taken into account, and the request for approval is sent to the regional marketing manager. This information is not known at design time, but it is known when the process is realized by a particular company.

4.1.3 People Resolution

People resolution is the act of identifying the people responsible for a particular generic human role. For example, the potential owners of a task are determined by people resolution. People resolution is performed by using queries assigned to people links.

The following are some examples for people queries used for people resolution:

- The persons with the names ‘Ivana’, ‘Claus’, ‘Dieter’, ‘Frank’, ‘Gerhard’, ‘Matthias’, ‘Alan’, and ‘Patrick’
- The persons with the user ID ‘gpfau’ and ‘klm’
- All persons belonging to the group of ‘approvers’, except the person who processed the previous approval step

- The manager of an organizational unit
- Any sales engineer located in New York
- The three people assigned to a particular work center

4.2 People Activities

People can be involved in business processes as a special kind of implementation of an activity. To facilitate this, a new BPEL activity type called people activity is required. From the point of view of the BPEL business process, a people activity is a basic activity, which is not implemented by a piece of software, but realized by an action performed by a human being. Therefore, the actor of a people activity is determined by a people link, as introduced before (see section 4.1.2 “People Links”). A people activity can be associated with different groups of people, one for each generic human role (see section 4.1.1 “Generic Human Roles”).

A BPEL engine must process people activities differently from activities invoking Web services. When a people activity is initiated by a BPEL engine, the engine creates work items (“to-dos”) for this activity and distributes them to all people eligible to perform it. Eventually, one of these eligible users decides to work on the activity by claiming it. This provides the user with unique access to the activity. The user can read the input data, perform whatever actions are needed, create the result data of this work as the activity’s output data (or a fault), and pass that data back to the BPEL engine (“check-in”), or make changes to the data in the underlying document. In addition, the nature of a knowledge-workers interaction dictates that there must be a provision for ad hoc attachments as a necessary part of the process. For example, the user attaches a sketch to qualify a decision that needs to be visible to participants downstream in the process. Once the user has performed his or her task, the work item is completed and the business process continues.

To define the implementation of a people activity, tasks are used. Tasks, as well as their different usage scenarios, with and without processes, are described in section 4.3 “Tasks”.

The people activity, like other BPEL activities, can have a name. It also has input and optional output to specify the data exchanged with the task. The people activity can also specify a priority that may be dependent on the data of the process. This priority does not determine when the activities are instantiated, but is used to prioritize the work for the people working on the tasks. A typical example is an airline reservation process that, when started by a “gold customer”, needs to ensure that the manual reservation tasks are performed with a higher priority than the same tasks of a parallel process instance started by a regular customer.

People activities allow the specification of deadlines. Deadlines are used to ensure service level agreements (SLAs) when dealing with work performed by people. Examples of deadlines are the maximum duration permitted until an activity is claimed, or the maximum duration until an activity has finished. These deadlines can be defined at design time, and can depend on data available to the process at runtime. If a deadline is exceeded, a BPEL fault could be raised.

4.3 Tasks

A task is an indivisible unit of work performed by a human being. The task specifies an action that a user must perform. This section describes the key characteristics of tasks, their properties, operations offered for task client applications, task states, and the interplay of tasks and people activities.

4.3.1 Properties

Tasks have a multilingual description of what the user needs to do when performing them. This description can be broken up into several parts, so that a one-line summary is displayed in a task list client, and a more detailed description is displayed when the user selects the task.

A task may also have a priority. Priorities are used to provide a sort criterion for tasks in the user's work list. If a task is used by a people activity that has also specified a priority, then the priority of the people activity overrides the priority of the task.

When initiated, tasks expect data, such as a spreadsheet, or details of a customer record that needs updating. Some data is essential, without which the task cannot be performed; some data is optional, such as a context for which a decision maker can access background information. This data is not just used during the execution of the activity, but can also be used for other aspects of task management, such as the people resolution, or within the task description that is displayed.

The task can have a set of deadlines associated with it, such as the time permitted before which it must be finished, or the time before a user claims it. If the deadlines are missed, escalations, such as sending notifications or performing an additional human resolution can be performed.

A user interface is associated with a task. This is used to present data needed to execute tasks, and to prepare data that is returned as the result of the task execution. Section 4.5 "User Interface" discusses different aspects of the user interface in more detail.

4.3.2 Operations for Client Applications

A client application (for example, task inbox) presents the tasks to the user. It must be able to interpret the task definition to render the correct user interface, such as calling a spreadsheet program or displaying a form for approving a customer record.

The client application has to be able to query a task's status and support its changes. It may interact with the task using the following operations:

- query available tasks – return a list of unfinished work or claimed tasks
- claim task – take over the responsibility for a task
- revoke claim – abandon the responsibility for a task (the task is returned to the potential owners)
- complete task – successfully finish work on a claimed task; task completion can be triggered either implicitly by the state change of the underlying document or explicitly by the user who completed work on the claimed task
- fail task – finish task indicating failure

These operations are described in more detail in the sections below.

Query Tasks

The query tasks operation is used by a client to retrieve a list of tasks the client might want to work with. The input of this operation is an expression in a query language such as XQuery or XPath. Examples of such queries are:

- Query all tasks I am currently working with
- Query all tasks I am a potential owner of

Claim Task, Revoke Claim

The claim operation is used by a user to assume responsibility for the execution of a task. The associated user becomes the owner of the task.

The revoke claim operation is used by a user who claimed a task before, and now wants to stop working on the task. The task can now be claimed by another potential owner.

Complete Task, Fail Task

The complete task and the fail task operations are called to finish work on a claimed task, providing as the result an output or fault message.

4.3.3 States

These are the most significant states that are used to describe the lifecycle of a task:

- ready – task has been created
- claimed – a user has claimed the task and has received its input data
- completed – a user has finished the task and provided its output data
- failed – a user has finished the task and provided a fault message

4.3.4 Inline Tasks and Standalone Tasks

Processes and tasks can be combined in different ways. Tasks can be used by processes either as *inline tasks* or as *standalone tasks*. The figure below shows the different models of interaction and describes how the coarse-grained distinction between standalone tasks and inline tasks can be refined into different sub-cases.

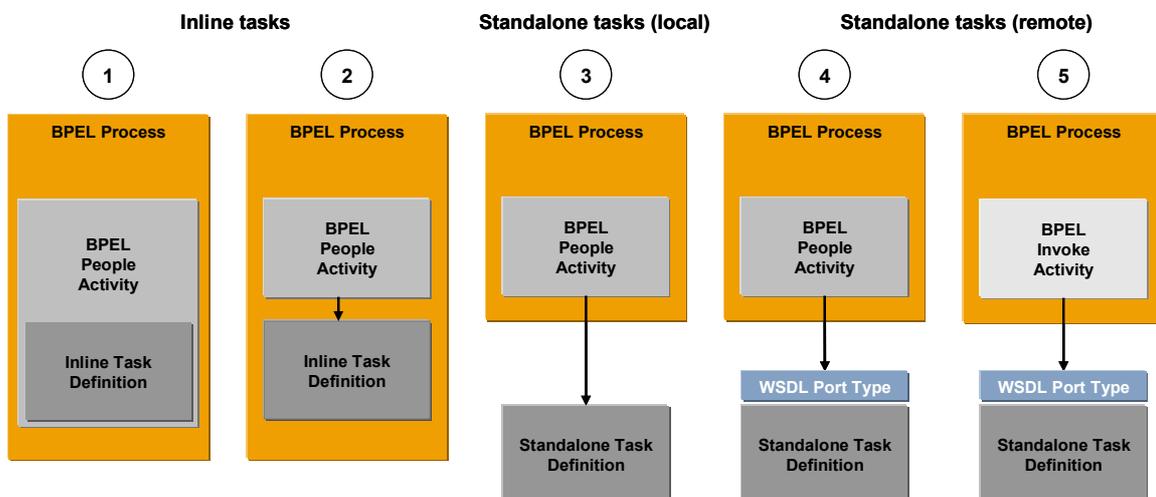


Figure 2. Models of Interaction between Tasks and Processes

Constellations 1 and 2 show models of interaction in which people activities (see section 4.2 “People Activities”) use tasks as *inline tasks*. An inline task can be defined as part of a people activity. In this case, the use of the task is limited to the people activity (constellation 1) encompassing it. Alternatively, a task can be defined as a top-level construct of the BPEL process. In this case, the same task can be used within multiple people activities, which is significant from a reuse perspective. Inline tasks have access to the context of the enclosing business process, for example, its variables (see section 4.4 “Context” below). BPEL processes that use tasks in this way are portable among runtime engines that implement BPEL4People.

Constellation 3 shows the use of a standalone task within the same environment, without the specification of a callable Web services interface on the task. This constellation is similar to constellation 2, except that the definition of the task is done independently of any process. As a result, the task has no direct access to process context. The task has no WSDL interface in this case, thus the task invocation is implementation-specific.

Another example of a standalone task is constellation 4. The major difference when compared to constellation 3 is that the task has a Web services interface specified using the Web Service Description Language (WSDL). A coordination protocol is used to communicate between process and task. Using this mechanism, state changes are propagated between task and process activity, and the process can perform life cycle operations on the task, such as terminating it. BPEL processes that use tasks in this way are portable across different BPEL engines that support BPEL4People. They are interoperable, assuming that both the process runtime and the task management implement the coordination protocol. This standalone task can be used independently of the business process, but other task initiators must support the coordination protocol.

Constellation 5 is the most generic case. A BPEL process uses a BPEL *invoke* activity to invoke a Web service that happens to be implemented by a standalone task. From the BPEL process’s perspective, the invocation of the task is transparent. In other words the BPEL process is not aware of the fact that the Web service is implemented as a task. Even BPEL engines that do not implement BPEL4People can use tasks in this way. Note that a remote standalone task can be used independently of any business process. It can be invoked by any Web services client. Section 4.6 “Services Implemented by People” provides more details on this constellation.

4.4 Context

Process context provides data needed to execute arbitrary activities, including people activities. For example, if a task must be performed by the plant manager, the information about which plant the process instance is dealing with must be known and used as input for the people query.

BPEL4People adds additional information to the standard BPEL process context. In addition to information such as variables and partner links, it adds information about users participating in the process and ad-hoc attachments as well as the means to access this information.

For example, using the process initiator derived from the process context enables explicit assignment of people activities to the person who started the process. Another example is the 4-eyes principle that requires access to the process context, to establish who performed a previous activity.

4.5 User Interface

People dealing with business processes do so by using a user interface. When people activities are used, the input data and output data must be rendered in a way that the user can interpret.

Examples of renderings are: a Web Dynpro provided by a certain service, a GUI dialog, an HTML form, an Excel spreadsheet, a voice message, or an SMS. For tasks that allow sending initiating messages to a process, a rendering is used to prepare the input message. If the corresponding operation of the process is a request-response operation, another rendering is used to display the result. Since the data presented to the user is well-defined, by using XML schema data types, for example, the renderings that present the data to the user and that allow the user to input data can be generated automatically, based on the message definitions. While this is sufficient in the general case, customizing is often desirable to apply the corporate look and feel to a UI, to improve usability, and so on.

In other situations the complete data is not available in the process. For example, a people activity “ReviewDocument” may include a document reference in its input data. A standard rendering would simply display the document reference as a string, but a custom client could contain the business logic to locate and include the referenced document in the rendering of the task. Thus, performing the review of the document becomes easier since the user does not have to copy and paste the document link and open the document manually.

BPEL4People does not stipulate how the data is rendered, but only provides a default method that shows the top level data of the message, for example, using HTML forms or XForms.

4.6 Services Implemented by People

Services implemented by people are provided by remote standalone tasks as discussed in constellation 5 in section 4.3.4 “Inline Tasks and Standalone Tasks”. A remote standalone task is a task defined as a separate entity outside of any process definition. It is a reusable entity. It can be reused in different BPEL processes, or by other service consumers outside a process definition. The standalone tasks model includes information needed to process tasks independently of any process (definition).

4.6.1 Motivation from a Business Process Perspective

When people activities are used in a process, it is explicitly clear that the implementation of such a step is performed by humans. While this is the right approach for some scenarios, especially for those where the people activity must be aware of the surrounding process, it is not suitable for every process. If the fact that a service is implemented by a human is something that does not need to be exposed, then a different approach is required. A solution for this scenario is to introduce

services that are implemented by humans. Using tasks as services, the process can transparently switch between services that have a “human implementation” and automatic services, without the need to change the process definition.

4.6.2 Motivation from a Web Services Perspective

From a pure Web services perspective, it is desirable to be able to invoke arbitrary services as Web services. Since people play an important role in many real-life scenarios, it is beneficial to provide the option to involve people in Web services applications.

4.6.3 Characteristics

To transparently switch between services with or without human implementation, a particular task offers the same interface and observable behavior that an automatic service offers. The task interface offers an operation with the same signature as an equivalent automatic service.

Note that there is a second interface, usually offered by the task infrastructure (task management engine), which is described in section 4.3.2 “Operations for Client Applications”. This interface allows client applications to provide a user interface to deal with tasks, that is, to query, claim, and complete them.

5 Summary

Although different bodies have worked to define interoperability standards for processes, including the user interaction with processes, this paper examines how interoperability can be pursued with respect to the BPEL language, which defines business process behavior based on Web services. Rather than defining a new standard for the user interaction, this paper shows that an extension of the existing standard can be achieved and is worth undertaking. A specification of BPEL4People that defines its syntax and semantics will follow.

6 Glossary

Business Administrator. A person with an interest in the smooth running of all process instances of a particular process type.

Generic human role. A descriptor identifying how a person interacts with the process. It includes the specific human roles such as potential owners, owners, and process initiator.

Inline task. A task defined within a BPEL process either as part of a people activity or within the scope of a BPEL process so that it can be reused in different parts of the same BPEL process.

Owner. A potential owner of a task becomes the owner when he or she claims the activity.

People activity. A new BPEL activity that assumes that people are involved in the execution of the process activity.

People links. A people link represents the group of people associated with a process or a people activity. It is evaluated at runtime using a people query.

People query. An expression based on an organizational model used to resolve users who take on the responsibility for a particular generic human role.

People resolution. The act of identifying the people who take on the responsibility for a particular generic human role.

Potential owners. People who are entitled to claim and complete an activity. This is one type of generic human role.

Process initiator. A person that creates an instance of a process, either directly or indirectly. This is one type of generic human role.

Process stakeholder. A person with an interest in the outcome of the process instance. This is one type of generic human role.

Standalone task. A task defined outside the scope of a BPEL process, but it can be referenced and invoked from different BPEL processes.

Task. An indivisible unit of work performed by a human being.

7 References

[BPEL4WS 1.1] Business Process Execution Language for Web Services Version 1.1, BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems, May 2003, available via <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>, <http://ifr.sap.com/bpel4ws/>

[WS-BPEL 2.0] Web Service Business Process Execution Language Version 2.0, Working Draft, July 2005, OASIS Technical Committee, available via <http://www.oasis-open.org/committees/wsbpel>

[WSDL 1.1] Web Services Description Language (WSDL) Version 1.1, W3C Note, available via <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XML Schema Part 1] XML Schema Part 1: Structures, W3C Recommendation, October 2004, available via <http://www.w3.org/TR/xmlschema-1/>

[XML Schema Part 2] XML Schema Part 2: Datatypes, W3C Recommendation, October 2004, available via <http://www.w3.org/TR/xmlschema-2/>

[XMLSpec] XML Specification, W3C Recommendation, February 1998, available via <http://www.w3.org/TR/1998/REC-xml-19980210>

[XPath 1.0] XML Path Language (XPath) Version 1.0, W3C Recommendation, November 1999, available via <http://www.w3.org/TR/1999/REC-xpath-19991116>