Work Distribution & Resource Management in BPEL4People: Capabilities & Opportunities[*]

Nick Russell and Wil M.P. van der Aalst

Department of Technology Management, Eindhoven University of Technology
GPO Box 513, NL5600 MB Eindhoven, The Netherlands
{n.c.russell,w.m.p.v.d.aalst}@tue.nl

Abstract. The BPEL4People and WS-HumanTask extensions to the BPEL proposal define the state of the art in resource management and work distribution in business process execution languages. In this paper, we use the workflow resource patterns as an evaluation framework to assess the capabilities of BPEL4People and WS-HumanTask in these areas and identify several areas where there is opportunity for further improvement.

Keywords: BPEL4People, WS-HumanTask, Resource Patterns

1 Introduction

One of the major objectives of workflow systems (and process-aware information systems (or PAIS) more generally) is to facilitate the distribution and coordination of work amongst the group of human resources associated with a process. There has been explosive growth in the commercial offerings available to support this objective as organisations seek out more effective ways in which to deploy their business processes across their workforce in a predictable, reliable and controlled manner. With the rise of the internet came a consequential extension of the underpinning technologies to embrace cross-organisational processes and the concept of the web service was born together with the notion of service oriented architectures which aim to facilitate business processes on the basis of loosely coupled (and potentially widely distributed) execution capabilities.

BPEL [11] was one of the first standards initiatives that attempted to establish a common processing framework and language that distinct execution engines could adopt in order to make the notion of a distributed business process based on disparate web services a viable possibility. It met with significant commercial interest and quickly established itself as the major standards initiative in this area. Developed by an industry consortium, it is perhaps not surprising that it met with early success as many of its contributors also had

[*] This research is conducted in the context of the *Patterns for Process-Aware Information Systems (P4PAIS)* project which is supported by the Netherlands Organisation for Scientific Research (NWO).

specific commercial interests that were directly furthered through its publication and broad adoptance. It is ironic therefore given the level of commercial input into the overall development of the BPEL standard that it had two major omissions: (1) a lack of recognition that business processes are generally hierarchical in form (resulting in the omission of the notion of subprocesses) and (2) a lack of consideration that business processes generally have some form of human involvement. Although these may have been deliberate omissions, they limit the applicability of BPEL in real-life processes.

The WS-BPEL Extension for Sub-Processes [9] proposal resolved the first of these issues. In an attempt to address the second, the BPEL4People [4] and WS-HumanTask [3] proposals have been released. They attempt to provide a series of extensions to WS-BPEL 2.0 [11] that integrate human resources into the overall execution of business processes. As these are early stage proposals, they are still open to comment in order to ensure that they meet with general acceptance before being finalised as standards. The focus of this paper is to review the conceptual foundation of BPEL4People and WS-HumanTask using the resource patterns as an evaluation framework. Through this examination, we hope to determine where the strengths and weaknesses of these proposals lie and what opportunities there may be for further improvement.

The *resource patterns* [12] were selected as the basis for evaluating the BPEL4-People and WS-HumanTask proposals as they offer a means of examining their capabilities from a conceptual standpoint in a way that is independent of specific technological and implementation considerations. The resource patterns were developed as part of the *Workflow Patterns Initiative,* an ongoing research project that was conceived with the goal of identifying the core architectural constructs inherent in workflow technology. The original objective was to delineate the fundamental requirements that arise during business process modelling on a recurring basis and describe them in an imperative way. A patterns-based approach was taken to describing these requirements as it offered both a language-independent and technology-independent means of expressing their core characteristics in a form that was sufficiently generic to allow for its application to a wide variety of offerings. To date, 126 patterns have been identified in the control-flow [13], data [14] and resource [12] perspectives and they have been used for a wide variety of purposes including evaluation of PAIS, tool selection, process design, education and training. The workflow patterns have been enthusiastically received by both industry practitioners and academics alike. The original Workflow Patterns paper [1] has been cited by over 600 academic publications and the workflow patterns website is visited by more than 300 individuals each day. Full details can be found at `http://www.workflowpatterns.com`.

The resource patterns form part of a surprisingly small body of research into resource and organisational issues in PAIS. Relevant research in the context of this paper includes early work by Bussler and Jablonski [5] which identifies a number of shortcomings of workflow systems when modelling organisational and policy issues. Du and Shan [6] present a design for a resource manager for a workflow system which includes a high level resource model together with proposals for resource definition, query and policy languages. Similarly in [8], Huang et al. propose a means of facilitating policy-based handling of resource assignment in a workflow context. The RBAC (Role-Based Access Control) model [7] describes a security framework for workflow that allows suitable users to be determined for a task. In [10] zur Muehlen presents a comprehensive overview of the organizational aspects of workflow technology. Several researchers [2,10] have developed meta-models describing the relationships between various workflow concepts, including aspects of work allocation, however these meta-models typically do not describe the dynamic aspects of work distribution.

The remainder of this paper proceeds as follows: Section 2 provides an overview of the BPEL4People and WS-HumanTask proposals. Section 3 presents an assessment of the two proposals using the workflow resource patterns as an evaluation framework. Section 4 discusses the results of the evaluation and identifies a number of areas where future possibilities exist for strengthening the proposals and Section 5 concludes the paper.

2 BPEL4People: overview and background

In this section we examine the intention and coverage provided by the BPEL4People and WS-HumanTask proposals from various perspectives, starting with their intention and relationship with related proposals and standards and then examining their informational and state-based characteristics on a comparative basis against those described by the workflow resource patterns.

2.1 Motivation and related standards

The stated intentions of the BPEL4People proposal and the closely coupled WS-HumanTask proposal are as follows [4,3]:

- BPEL4People: to support a broad range of scenarios that involve people within business processes.
- WS-HumanTask: to provide a notation, state diagram and API for human tasks as well as a coordination protocol that allows interaction with human tasks in a more service-oriented fashion, and at the same time control task autonomy.

In order to achieve these objectives, the BPEL4People proposal assumes the services of a number of related standards. Figure 1 illustrates the relationship between the various standards that are required in order to support the BPEL4People proposal. It is interesting to note that whilst BPEL4People has the most visibility, it provides minimal new capabilities from a resource perspective and essentially acts only to extend the notion of an Activity to that of a PeopleActivity hence enabling the definition of inline and local tasks carried out under the auspices of a human resource. The bulk of the new features associated with work items, work distribution and state management are actually provided

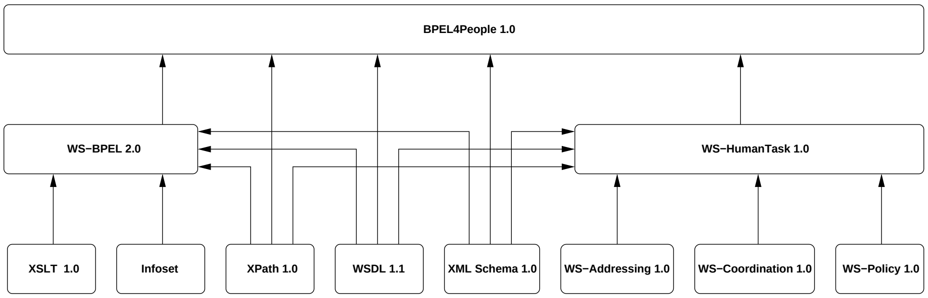

Fig. 1. Web services standards hierarchy

by the WS-HumanTask proposal which also introduces the notion of a standalone task (i.e. a task whose implementation is defined outside of the context of the BPEL process) that is undertaken by a human resource. Consequently, much of the remainder of this document will tend to focus on the capabilities defined by the WS-HumanTask proposal.

2.2 Information coverage of the WS-HumanTask extension

A significant insight into the overall capabilities of the WS-HumanTask extension can be gained from an examination of the data elements that make up the associated schema. Figure 2 illustrates the major data elements that make up the workflow resource patterns and the WS-HumanTask extension in terms of UML class diagrams and identifies the major correspondences between them. Much of the information content is common to both proposals, although there are some noteworthy distinctions between them.

The resource patterns:

- assume a richer organisational model both to capture relationships between resources, job and organisational units, and also allow this information to be used as the basis for work distribution directives (see ❶ in Figure 2);
- include the notion of execution history (where the execution outcomes of activities in multiple concurrent cases are permanently logged) and allow this data to be used in work distribution directives ❷;
- support the notion of extensible resource descriptions (via capabilities) which can be used when making decisions about distributing work items ❸; and
- provides a comprehensive authorisation framework which strictly defines the work item privileges available to individual resources at runtime ❹.

The BPEL4People/WS-HumanTask proposals:

- distinguish between a series of distinct task implementation strategies (local, remote, etc.) ❺;

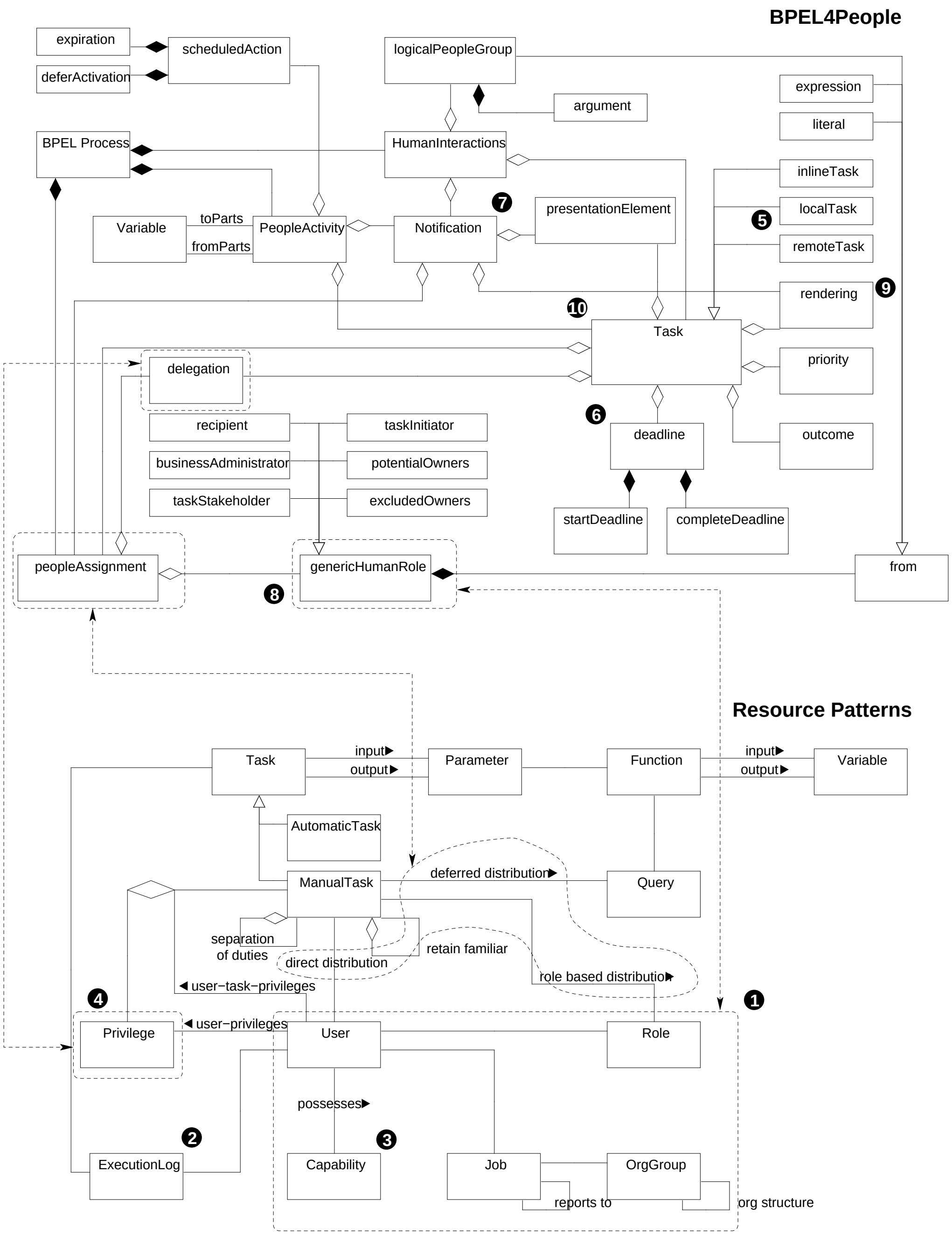

Fig. 2. Comparison of information coverage in WS-HumanTask [3] and the workflow resource patterns [12]

- incorporate facilities for defining commencement and completion deadlines for tasks along with the actions that should be taken when the deadline is reached. Similar capabilities exist for specifying escalations ❻;
- support a series of notification capabilities to advise resources of adverse work item execution circumstances ❼;
- include a series of designated roles for each task that describe specific privileges. These include task initiator and task stakeholder ❽;
- incorporate the identification of rendering facilities for each task which describe the potential user interfaces that will be presented to resources undertaking the task ❾; and
- include a means of representing data specific to a task instance (although interestingly, individual task data instances are only referenced by an id field and it is unclear how data elements are related to specific task instances in a specific case) ❿.

Some of the distinctions outlined above are related to scope, others may indicate potential areas for improvement or enhancement and are discussed at greater length later in the paper. One observation that can be made at this point is that the WS-HumanTask proposal considers implementation aspects for individual tasks (e.g. presentation elements, interface details and deadlines) in addition to issues associated with work distribution. In contrast, the resource patterns operate at a conceptual level and focus strictly on issues of resource management and work distribution. There is currently no consideration of functional details associated with task enactment in the workflow patterns framework and this raises the question of whether there should be further investigations into the potential for a set of *operational patterns* describing task implementation.

2.3 Dynamic coverage of the WS-HumanTask extension

The state models that underpin the resources patterns and the WS-HumanTask proposal are analogous. Figure 3 illustrates the state transition diagrams for both of them. A major difference between them is that WS-HumanTask also includes broader consideration of error states and allows tasks that haven't yet started to be suspended (as shown by ❶ and ❷ respectively). In contrast, the resource patterns differentiate between work items offered to single and multiple resources (shown by ❸) and support a slight wider range of detour actions (as illustrated by the bold arcs).

3 Capabilities: an assessment of resource pattern support

In the following section, we provide an evaluation of the capabilities of BPEL4People and WS-HumanTask from a resource perspective. This assessment utilises the workflow resource patterns as an evaluation framework thus providing a technologically agnostic means of examining the capabilities of the two proposals. There are seven distinct groups of resource patterns as follows:

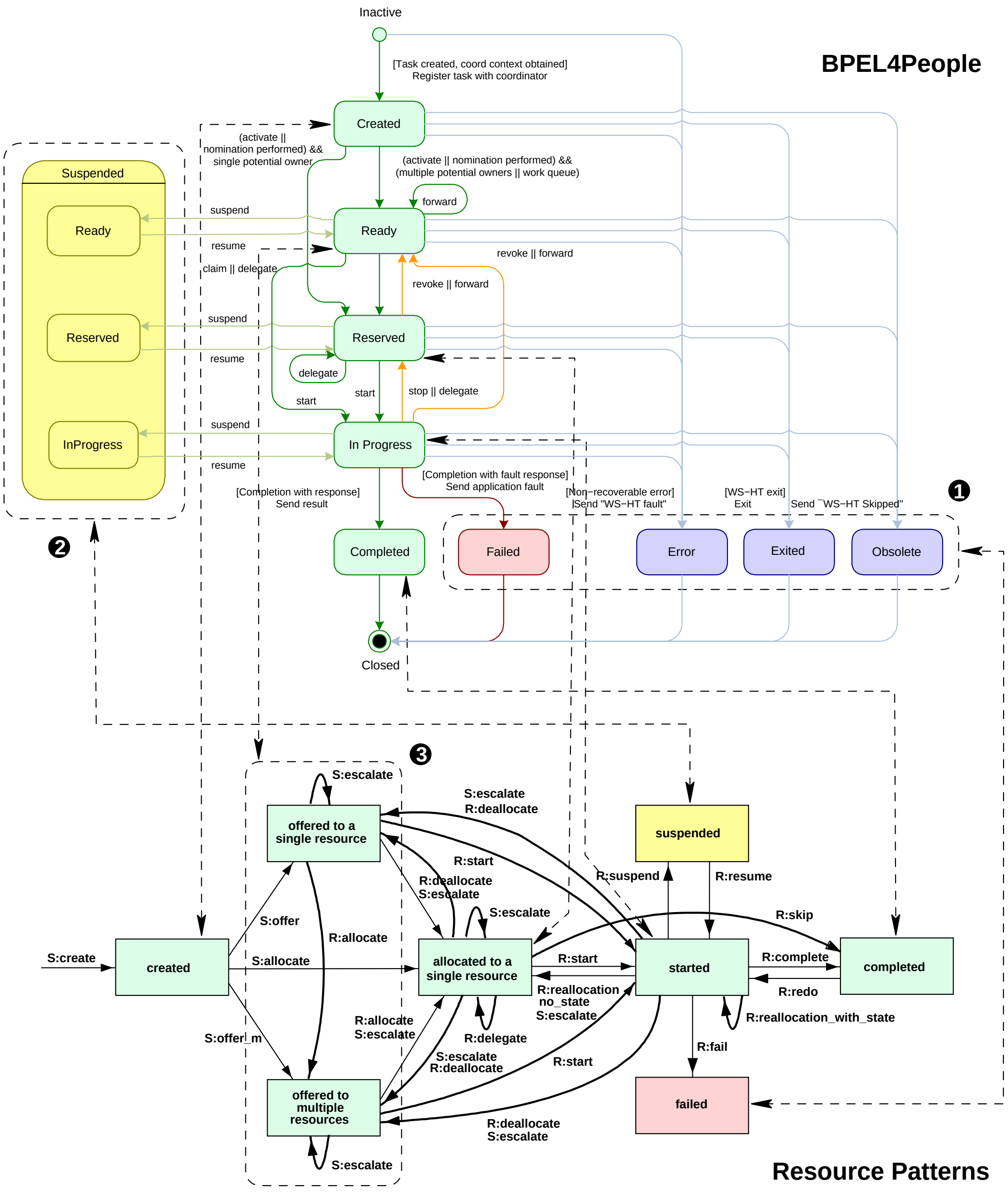

Fig. 3. Comparison of supported states in WS-HumanTask [3] and the workflow resource patterns [12]

- *creation* patterns – which correspond to limitations specified in the design time model on the manner in which a work item is executed by resources;
- *push* patterns – which characterise situations where newly created work items are proactively offered or allocated to resources by the system;
- *pull* patterns – which correspond to situations where individual resources take the initiative in committing to and undertaking available work items;
- *detour* patterns – which refer to situations where work allocations that have been made for resources are interrupted either by the system or at the instigation of individual resources;
- *auto-start* patterns – which relate to situations where the execution of work items is triggered by specific events in the lifecycle of the work item or the related process definition;
- *visibility* patterns – which describe the various scopes in which work item availability and commitment are able to be viewed by resources; and
- *multiple resource* patterns – which characterise situations where the correspondence between the resources and work items in a given allocation or execution is not 1-1.

The following sections describe the support for each of these patterns by the BPEL4People (B4P) and WS-HumanTask (HT) proposals in detail[1].

3.1 Creation patterns

The intention of the BPEL4People and WS-HumanTask proposals – to support a broad range of scenarios that involve people within business processes – is immediately reflected by the range of creation patterns that are supported as illustrated in Table 1. As the original BPEL proposal provided no guidance in this area, the relative change is significant.

Resources are identified within the context of a BPEL process and work can be distributed directly to them by name or indirectly via role-based groupings or based on the results of queries. Through the use of these queries, separation of duties and retain familiar constraints can be specified between work items within a case. Less well-supported however is the ability to specify more precise work distribution requirements for a task in terms of organisational or history-based criteria. The organisational model supported with the BPEL4People/WS-HumanTask framework is relatively simplistic and does not explicitly identify job roles, reporting lines or relationships between organisational groupings hence these cannot be used when distributing work. Similarly, it is only possible to use the execution characteristics of work items in the same case when framing historical work distribution requirements. There is no support for adding further descriptive criteria to individual resources (i.e. capabilities) and using these when distributing work items. An additional shortcoming relates to the limited ability within BPEL4People/WS-Human-Task to impose an authorisation framework

[1] Details of individual pattern realisations in BPEL4People and WS-HumanTask can be found in the companion technical report BPM-07-11 at `www.BPMcenter.org`.

Table 1. Creation patterns support

Nr	Pattern	Rating	Rationale
1	Direct Distribution	+	Supported by literal assignment of potential/actual task owners (HT)
2	Role-Based Distribution	+	Supported by logical people group assignment of potential/actual task owners (HT)
3	Deferred Distribution	+	Supported by assignment of potential/actual owners based on expressions (HT)
4	Authorisation	+/–	Limited support for nominating delegation and skipping on a per task basis but no general support for user privileges (HT)
5	Separation of Duties	+	Supported via excluded owners attribute for <peopleAssignment> elements (HT)
6	Case Handling	–	No support for case handling
7	Retain Familiar	+	Supported by assigning actual owner to the same value as the actual owner of another task (HT)
8	Capability-Based Distribution	–	No support for resources to have additional capability attributes
9	History-Based Distribution	+/–	Expressions can utilise details associated with task instances for a given user via the getMyTasks function although its unclear how this can be generalised to broader history-based queries (HT)
10	Organisational Distribution	+/–	The organisational model only identifies group membership and role participation for individual resources (HT)
11	Automatic Execution	+	Directly supported by BPEL

on resources and the range of actions that they are able to undertake with respect to overall process execution (other than for delegate and skip actions). Similarly, it is not possible to constrain the resources that individual tasks can be distributed to in a guaranteed way (e.g. a work item could ultimately be delegated to any resource not just one that satisfied the distribution criteria associated with the task).

3.2 Push and pull patterns

The work distribution model in WS-HumanTask is based on work being advertised to individual resources and those resources making a decision on what work they will commit to undertaking and when they will start it. The degree of support for specific push patterns is illustrated in Table 2. Work items can be offered to multiple resources or allocated to one of them, however it is not possible to offer a work item to a single resource on a non-binding basis. There is no support for randomly selecting a resource to undertake a work item or for distributing work on a round robin (i.e. an equitable) basis, however it does appear that the possibility may exist to distribute work on a shortest-queue basis where there are multiple potential resources for the same work item (although the precise means of implementing this using the provided function set is a little unclear). All work is distributed at the time the task with which it is associated is enabled. As indicated previously, under the WS-HumanTask proposal, work

Table 2. Push and patterns support

Nr	Pattern	Rating	Rationale
			Push patterns
12	Distribution by Offer - Single Resource	–	Not supported. If there is only one potential owner for a work item, then it is allocated to them
13	Distribution by Offer - Multiple Resources	+	Supported by setting multiple potential owners for a task instance in the Created or Ready state (HT)
14	Distribution by Allocation - Single Resource	+	Supported by setting a single potential owner for a task instance in the Created or Ready state (HT)
15	Random Allocation	–	Not supported
16	Round Robin Allocation	–	Not supported
17	Shortest Queue	+/–	It would appear that this pattern can be supported by using an expression to set the actual owner for a task instance to the potential owner with the shortest work list, however its unclear if this can be implemented with the supported functions (HT)
18	Early Distribution	–	Not supported
19	Distribution on Enablement	+	Potential owners are notified of tasks when they are created
20	Late Distribution	–	Not supported
			Pull patterns
21	Resource-Initiated Allocation	+/–	Supported via the claim function providing the work item is offered to more than one user. It is automatically started if only offered to one resource (HT)
22	Resource-Initiated Execution - Allocated Work Item	+	Supported via the start function (HT)
23	Resource-Initiated Execution - Offered Work Item	+	Supported via the start function (HT)
24	System-Determined Work Queue Content	–	No ability to limit or order the work queue for a resource
25	Resource-Determined Work Queue Content	+	The simple and advanced query functions provide the ability for resources to restrict and format the content of their worklists (HT)
26	Selection Autonomy	+	Resources can choose to start any task instance available available to them (HT)

is advertised to resources and they commit to undertaking work items of their choice and can choose the time of commencement. The degree of support for specific pull patterns is illustrated in Table 2. There is provision for a resource to execute multiple work items simultaneously and to order and select the content of their own work queue via queries however it is not possible for the system to impose a default ordering or content for work queues.

3.3 Detour, auto-start, visibility and multiple resource patterns

Detour patterns provide the ability for resources (and potentially the system) to alter the normal sequence and manner in which work items are distributed for execution. A variety of distinct "detours" are supported, as illustrated in Table 3, although there is no ability to undertake work items outside of the

Table 3. Detour, auto-start, visibility and multiple resource patterns support

Nr	Pattern	Rating	Rationale
Detour patterns			
27	Delegation	+	Supported via the delegate function (HT)
28	Escalation	+	Escalations can be specified for tasks. Both commencement and completion deadlines are supported together with logical conditions that restrict their application (HT)
29	Deallocation	+	Supported via the release function (HT)
30	Stateful Reallocation	+	Supported via the the forward function (HT)
31	Stateless Reallocation	–	Not supported
32	Suspension/Resumption	+	Supported via the suspend and resume functions (HT)
33	Skip	+	Supported via the skip function (HT)
34	Redo	–	Not supported
35	Pre-Do	–	Not supported
Auto-start patterns			
36	Commencement on Creation	–	Not supported. Task instances must be explicitly started by an owner
37	Commencement on Alloc.	–	Not supported. Task instances must be explicitly started by an owner
38	Piled Execution	–	Not supported
39	Chained Execution	–	Not supported
Visibility patterns			
40	Configurable Unallocated Work Item Visibility	+/–	The advanced query function seems to support this but its operation across process instances and also for querying work items not allocated to the requesting resource is unclear. Also it is not a mandatory part of the proposal (HT)
41	Configurable Allocated Work Item Visibility	+/–	The advanced query function seems to support this but its operation across process instances and also for querying work items not allocated to the requesting resource is unclear. Also it is not a mandatory part of the proposal (HT)
Multiple resources patterns			
42	Simultaneous Execution	+	Directly supported (HT)
43	Additional Resources	–	Not supported. There can only be one resource for a task instance

normal execution sequence (i.e. redo/pre-do) or to rollback their execution state (i.e. stateless reallocation). Auto-start patterns correspond to mechanisms which attempt to speed up the overall throughput of work in various ways. As indicated in Table 3, BPEL4People and WS-HumanTask do not provide any capabilities in this area. Visibility patterns describe mechanisms within the workflow system for limiting the visibility of upcoming or in progress work items to selected resources. As indicated in Table 3, WS-HumanTask potentially provides support in this area, however it is unclear how the query function operates in the context of multiple concurrent processes. Multiple resource patterns characterise situations where the work item - resource relationship is not 1-1. As indicated in Table 3, WS-HumanTask supports the notion of simultaneous execution (i.e. one resource running multiple work items) but only allows a work item to be allocated to a single resource.

4 Opportunities

The BPEL4People and WS-HumanTask proposals provide comprehensive support for incorporating tasks undertaken by human resources within the overall process execution framework that BPEL provides. There is a broad range of ways in which human resources can be represented and grouped: individually, via roles, groups and also as a result of query execution. These strategies can also be used as the basis for work assignments. Moreover there are a number of distinct ways in which human tasks can be implemented, ranging from inline activities in which both the task definition and the associated work directives form part of the same node in the process through to standalone tasks (defined elsewhere) which are coordinated by a PeopleActivity node in a BPEL process.

Nonetheless, *the patterns evaluation undertaken in the previous section identifies a number of potential opportunities* that these two proposals could pursue to further strengthen their ability to support human resource involvement in business processes. These issues are discussed in the following sections. In order to give an indication of effort associated with addressing each of them, we have rated their complexity from ✯ (minimal effort) to ✯✯✯ (significant effort).

4.1 Non-binding offers to a single resource ✯

There is no ability in the context of WS-HumanTask to offer a work item (i.e. not allocate) to a single resource. Where a newly created work item is identified as having a single potential owner, then it is assumed to be allocated to that resource (i.e. reserved) on a binding basis. There is no option that allows the resource to decline to undertake the offered work item.

4.2 Automatic selection of a resource ✯

Where multiple potential resources are identified when seeking to distribute a work item, there is no means of selecting a single resource to whom it should be allocated. Common means of selecting a suitable resource where several are identified include round-robin (i.e. distribute work evenly), least busy user (e.g. shortest queue) and random selection.

4.3 Distinguishing execution instances ✯

There is minimal distinction made between tasks and task instances. Whilst this is inconsequential when specifying a static process model, many of the elements in the enhanced BPEL4People/WS-HumanTask proposals require specific addressing e.g. invoking a remote task requires knowledge of the remote endpoint, the process name, task name, the specific process instance and task instance being sought. Similarly, data elements are specific to a process instance (not all process instances) hence they also need to be named accordingly. Moreover there seems to be no notion of process instance or task instance identifiers in these naming schemes that facilitiate navigation to a specific instance that is currently in progress (e.g. for delivering a notification).

4.4 Richer resource descriptions ✩✩

There is no support for more detailed definition of specific resources (e.g. via capabilities) or for the use of resource characteristics when distributing work. This limits any possibility for differentiating between specific resources on the basis of characteristics that they possess when distributing work. In effect, all resources are treated as being identical when making a decision about where to route a work item. Note that multiple processes and organisations may want to share information about resource capabilities and requirements. BPEL4People/WS-HumanTask could play a prominent role here were they able to utilise and mediate more detailed resource definitions held in distinct systems (e.g. X.500 style directory services, ERP/HR systems) for work distribution purposes.

4.5 Inclusion of an organisational framework ✩✩✩

The organisational model provided is relatively minimalistic and does not take common concepts such as jobs, reporting lines, organisational groups etc. into account nor can these characteristics be used for work distribution purposes or for identifying or grouping resources in a generic sense. As the relationships between resources cannot be described in terms of the organisational context in which they operate, it is not possible to describe a variety of common approaches to work distribution, e.g. offer the work item to a clerk reporting to the manager. Moreover, work distribution cannot be framed in terms of organisational positions or jobs. This is a common approach to describing work responsibilities in many organisations as it minimises the need to change the work distribution directives associated with a process when staffing arrangements change.

4.6 Work distribution based on historical information ✩✩✩

Within a process instance, there is minimal access to historical information (and at that, only that referring to preceding work items in the same case). Moreover it is not clear to what extent this can be used for work distribution purposes. This is an obvious area where further clarity can be added to these proposals. The use of historical information, particularly that based on a multitude of previously completed process instances, provides a useful means of targetting suitable resources when distributing work items and allows approaches such as “allocate to the most experienced resource” or “offer to the person who did it least recently” to be implemented.

4.7 Resource privileges ✩✩

One notable absence is the ability to specify privileges defining what actions a resource can undertake. Ideally it should be possible to specify these on a per-task basis in order to restricting the range of actions that a resource can initiate in regard to a task (e.g. delegation, reallocation etc.).

4.8 Independent authorisation framework ✯✯✯

There is no provision for imposing an authorisation framework over the tasks in a process to limit the potential range of resources to whom they can be directed and that are able to ultimately execute them. This is particularly useful in an enterprise context in order to limit how a task can be executed, regardless of the process definition in which it appears or how it is routed.

4.9 User-initiated optimization ✯✯

In many situations, opportunities that may exist for optimising work throughput can best be identified by resources involved in the conduct of work associated with the actual process. There are a number of approaches to expediting the completion of a process instance, these include automatically starting tasks when they are created or allocated, automatically starting subsequent tasks and allocating to the resource that completed the preceding task in a process instance, and allocating all instances of a given task to the same resource regardless of the process instance in which they occur (i.e. chained and piled execution). None of these facilities are supported in the BPEL4People or WS-HumanTask proposals.

4.10 Provision of a worklist metaphor ✯✯

One of difficulties with the proposals is that there is an absence of a clearly defined user interface that describes how a resource interacts with the process engine when undertaking work items and what details associated with each work item are disclosed. Typically in workflow systems, this interface is termed a worklist handler although other metaphors are possible e.g. work queues. Access to work items is supported via user-initiated queries, however the operation of these queries is unclear when requesting work items in multiple process instances. Moreover the use of such queries also removes any potential for imposing a uniform view of work distributions to all users.

5 Conclusions

This paper has examined the support that the BPEL4People and WS-HumanTask proposals provide for extending the BPEL offering to deal with activities that are undertaken by human resources. It uses the workflow resource patterns as an evaluation framework to assess the capabilities of these proposals and in doing so identifies both their strengths and several areas where opportunities exist for further improvement.

References

1. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(3):5–51, 2003.

2. W.M.P. van der Aalst, A. Kumar, and H.M.W. Verbeek. Organizational modeling in UML and XML in the context of workflow systems. In H. Haddad and G. Papadopoulos, editors, *Proceedings of the 18th Annual ACM Symposium on Applied Computing (SAC 2003)*, pages 603–608, Melbourne, Florida, USA, 2003. ACM Press.
3. A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M Zeller. Web Services Human Task (WS-HumanTask), version 1.0, 2007. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf`.
4. A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M Zeller. WS-BPEL Extension for People (BPEL4People), version 1.0, 2007. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf`.
5. C. Bussler and S. Jablonski. Policy resolution for workflow management systems. In *Proceedings of the 28th Hawaii International Conference on System Sciences*, volume 4, pages 831–840, Wailea, HI, USA, 1995. IEEE Computer Society.
6. W. Du and M.C. Shan. Enterprise workflow resource management. In *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE'99)*, pages 108–115, Sydney, Australia, 1999. IEEE Computer Society Press.
7. D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3):224–274, 2001.
8. Y.N. Huang and M.C. Shan. Policies in a resource manager of workflow systems: Modeling, enforcement and management. Technical Report HPL-98-156, 1999. `http://www.hpl.hp.com/techreports/98/HPL-98-156.pdf`.
9. M. Kloppman, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic. WS-BPEL Extension for Sub-Processes: BPEL-SPE, 2005. `ftp://www6.software.ibm.com/software/developer/library/ws-bpelsubproc.pdf`.
10. M. zur Muehlen. Organizational management in workflow applications – issues and perspectives. *Information Technology and Management*, 5:271–294, 2004.
11. OASIS. Web Services Business Process Execution Language for Web Services version 2.0, 2007. `http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf`.
12. N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Workflow resource patterns: Identification, representation and tool support. In O. Pastor and J. Falcão e Cunha, editors, *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, volume 3520 of *Lecture Notes in Computer Science*, pages 216–232, Porto, Portugal, 2005. Springer.
13. N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar. Workflow control-flow patterns: A revised view. Technical Report BPM-06-22, 2006. `http://www.BPMcenter.org`.
14. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow data patterns: Identification, representation and tool support. In L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, editors, *Proceedings of the 24th International Conference on Conceptual Modeling (ER 2005)*, volume 3716 of *Lecture Notes in Computer Science*, pages 353–368, Klagenfurt, Austria, 2005. Springer.