Evaluation of the BPEL4People and WS-HumanTask Extensions to WS-BPEL 2.0 using the Workflow Resource Patterns*

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

1 Introduction

One of the major objectives of workflow systems (and process-aware information systems 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 initiative 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 [4] was one of the first standards initiatives that attempted to establish a common execution 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 quite 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 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.

The WS-BPEL Extension for Sub-Processes [5] proposal addresses the first of these issues. In an attempt to correct the second, the BPEL4People [3] and WS-HumanTask [2] proposals have been released. They attempt to provide a series of extensions to WS-BPEL 2.0 [6] that integrate human resources into the overall execution of business processes.

*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).

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 [7, 10] were selected as the basis for evaluating the BPEL4People 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 [8], data [9, 11] and resource [7, 10] perspectives and they have been used for a wide variety of purposes including evaluation of PAISs, 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 150 academic publications and the workflow patterns website has been visited more than 80,000 times. Full details can be found at `http://www.workflowpatterns.com`.

2 Scope

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 Intent

The stated intent for the BPEL4People proposal and the closely coupled WS-HumanTask proposal are as follows:

- 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

2.2 Related standards

Figure 1 illustrates the relationship between the various standards that are required in order to support the BPEL4People proposal.

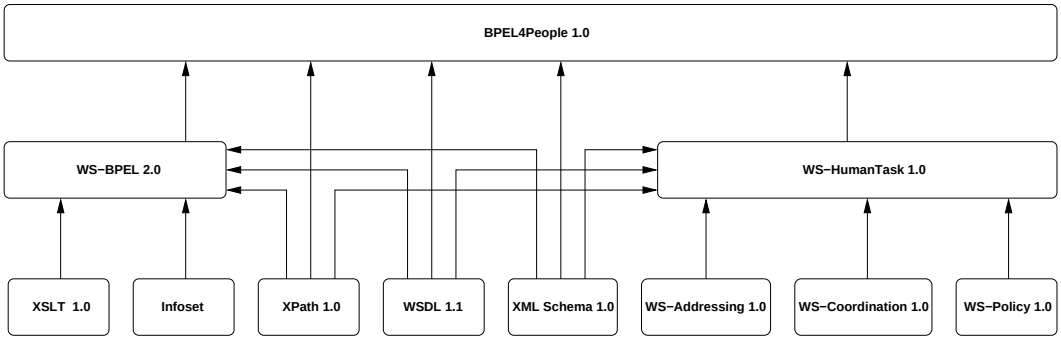

Figure 1: Web services standards hierarchy

Its interesting to note that whilst the BPEL4People proposal 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 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.

2.3 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 of work distribution directives;
- 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;
- support the notion of extensible resource descriptions (via capabilities) which can be used when making decisions about distributing work items; and
- provides a comprehensive authorisation framework which strictly defines the work item privileges available to each resource at runtime.

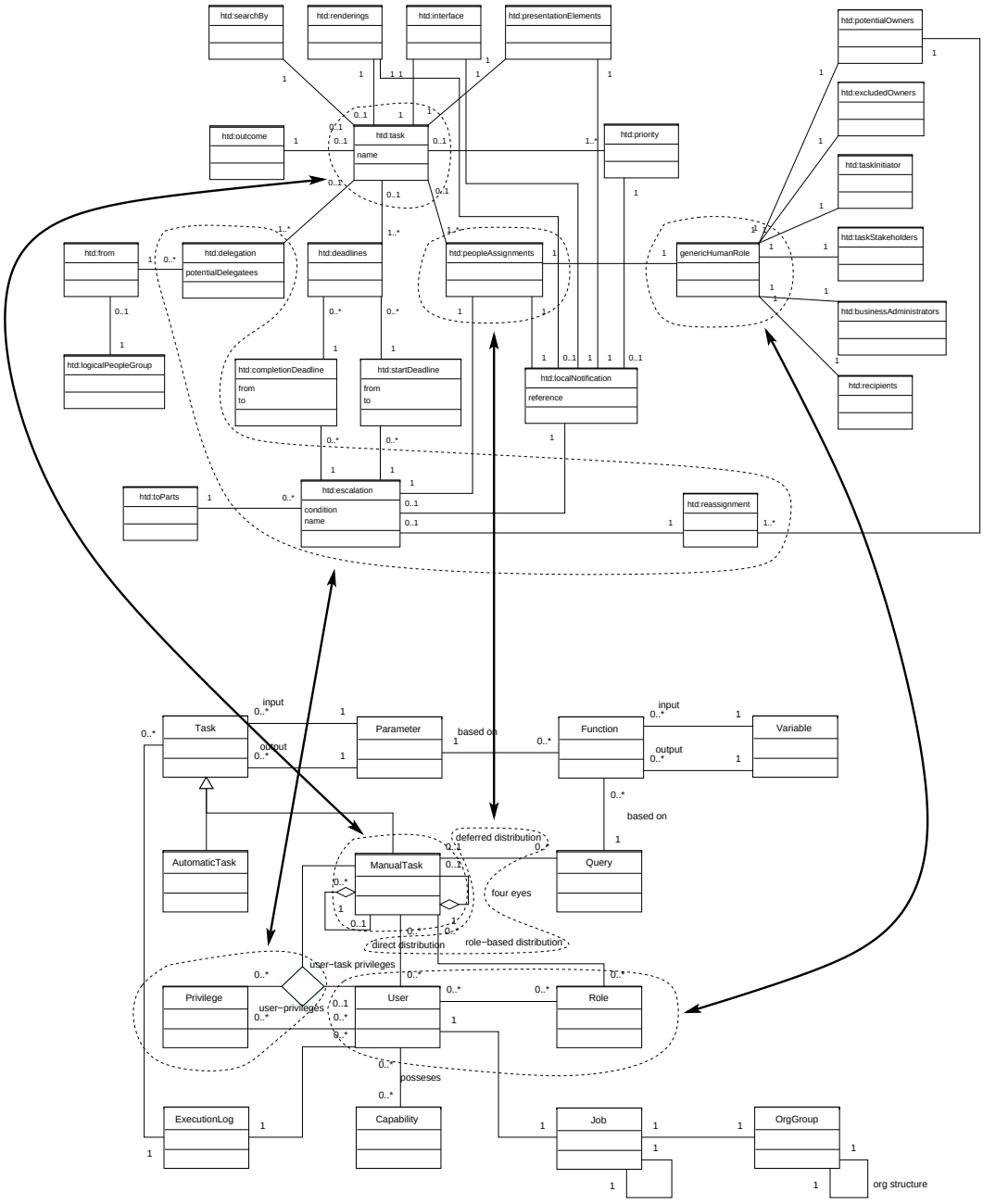

Figure 2: Comparison of information coverage in WS-HumanTask and workflow resource patterns

The BPEL4People/WS-HumanTask proposals:

- distinguish between a series of distinct task implementation strategies (local, remote etc.);
- 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;
- support a series of notification capabilities to advise resources of adverse work item execution circumstances;
- include a series of designated roles for each task that describe specific privileges. These include task initiator and task stakeholder;
- incorporate the identification of rendering facilities for each task which describe the potential user interfaces that will be presented to resources undertaking the task;
- support the notion of ad-hoc data attachments to tasks;
- support the notion of user comments being attached to tasks; 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 a specific task instances in a specific case).

2.4 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 each of them. Major differences between them are that WS-HumanTask also includes broader consideration of error states and allows tasks that haven't yet started to be suspended. In contrast, the resource patterns support a slight wider range of detour actions (as illustrated by the bold arcs).

3 Resource pattern support

In the following section, we provide an evaluation of the capabilities of BPEL4People amd 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:

- *creation* patterns – which describe 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 workflow system;
- *pull* patterns – which correspond to situations where individual resources take the initiative in committing to and undertaking available work items;

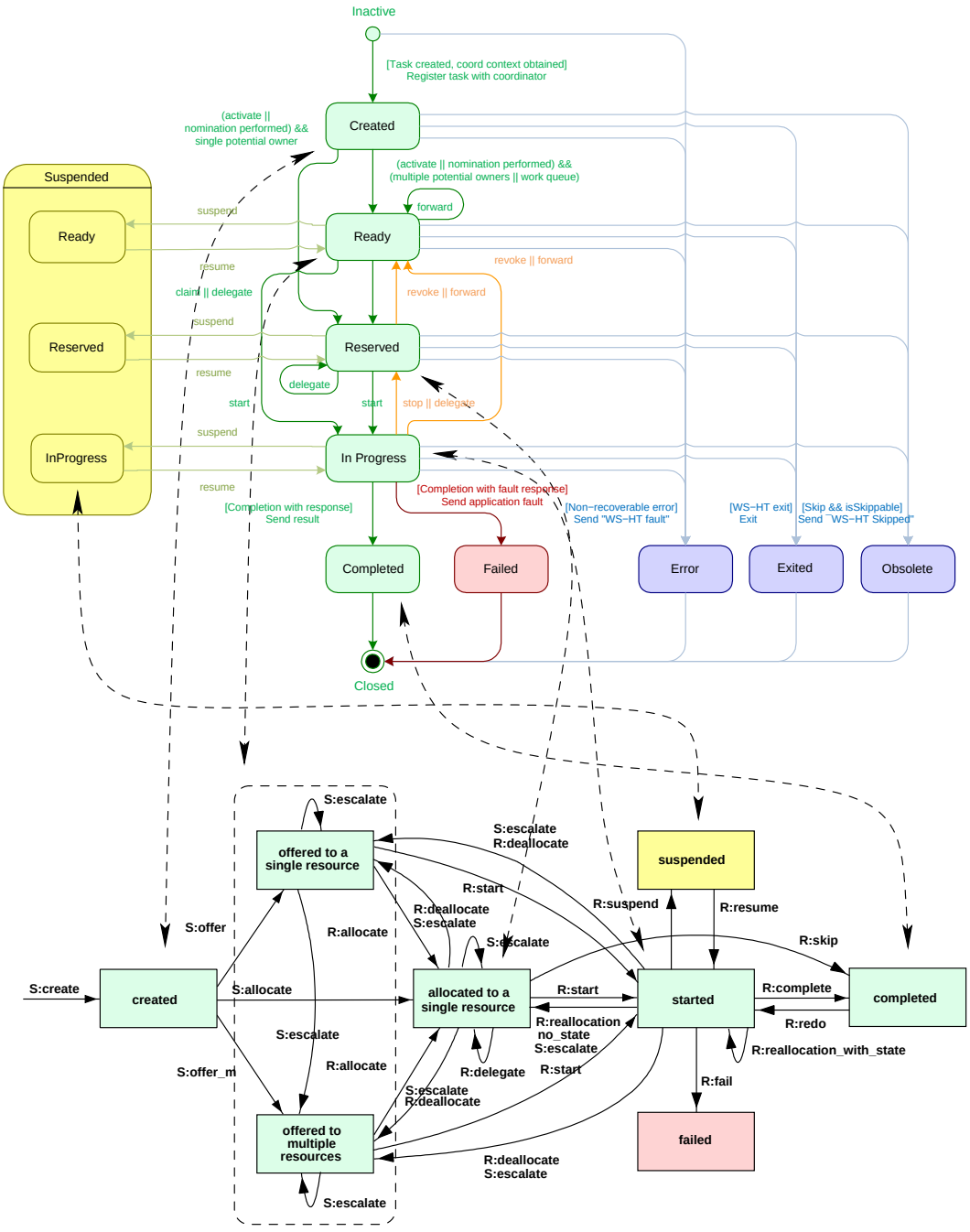

Figure 3: Comparison of supported states in WS-HumanTask and the workflow resource patterns

- *detour* patterns – which refer to situations where work allocations that have been made for resources are interrupted either by the workflow 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 workflow 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 and WS-HumanTask proposals in detail.

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 grouping 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-HumanTask to impose an authorisation framework 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).

Nr	Pattern	Rating	Rationale
1	Direct Distribution	+	Supported via literal assignment of potential or actual task owners
2	Role-Based Distribution	+	Supported via logical people group assignment of potential or actual task owners
3	Deferred Distribution	+	Supported via assignment of potential or actual owners based on expressions
4	Authorisation	+/−	Limited support for nominating delegation and skip on a per task basis but no general support for user privileges
5	Separation of Duties	+	Supported via excluded owners attribute for <peopleAssignment> elements
6	Case Handling	−	No support for case handling
7	Retain Familiar	+	Supported by assigning actual owner to the same value as actual owner of another task
8	Capability-Based Distribution	−	No support for people to have additional capability attributes
9	History-Based Distribution	+/−	Expressions can take into account the 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
10	Organisational Distribution	+/−	The organisational model only identifies group membership and role participation for individual people
11	Automatic Execution	+	Directly supported by BPEL

Table 1: Creation patterns support

3.2 Push patterns

The work distribution model in WS-HumanTask is based on work being advertised to individual resources via their worklists 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 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.

Nr	Pattern	Rating	Rationale
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
14	Distribution by Allocation - Single Resource	+	Supported by setting a single potential owner for a task instance in the Created or Ready state
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
18	Early Distribution	−	Not supported
19	Distribution on Enablement	+	Potential owners are notified of tasks when they are created
20	Late Distribution	−	Not supported

Table 2: Push patterns support

3.3 Pull patterns

As indicated previously, under the WS-HumanTask proposal, work 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 3. 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.

3.4 Detour 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 4, although there is no ability to undertake work items outside of the normal execution sequence (i.e. redo/pre-do) or to rollback their execution state (i.e. stateless reallocation).

Nr	Pattern	Rating	Rationale
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 user.
22	Resource-Initiated Execution - Allocated Work Item	+	Supported via the start function
23	Resource-Initiated Execution - Offered Work Item	+	Supported via the start function
24	System-Determined Work Queue Content	−	No ability to limit or order the work queue for a person
25	Resource-Determined Work Queue Content	+	The simple and advanced query functions provide the ability for users to restrict and format the content of their worklists
26	Selection Autonomy	+	People can choose to start any task instance available available to them

Table 3: Pull patterns support

3.5 Auto-start patterns

Auto-start patterns correspond to mechanisms which attempt to speed up the overall throughput of work in various ways. As indicated in Table 5, BPEL4People and WS-HumanTask do not provide any capabilities in this area.

3.6 Visibility patterns

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 5, WS-HumanTask potentially provides support in this area, however it is unclear how the query function operates in the context of multiple concurrent processes.

3.7 Multiple resource patterns

Multiple resource patterns characterise situations where the work item - resource relationship is not 1-1. As indicated in Table 5, 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.

Nr	Pattern	Rating	Rationale
27	Delegation	+	Supported via the delegate function
28	Escalation	+	Escalations can be specified for tasks. Both commencement and completion deadlines are supported together with logical conditions that restrict their application
29	Deallocation	+	Supported via the release function
30	Stateful Reallocation	+	Supported via the the forward function
31	Stateless Reallocation	−	Not supported
32	Suspension/Resumption	+	Supported via the suspend and resume functions
33	Skip	+	Supported via the skip function
34	Redo	−	Not supported
35	Pre-Do	−	Not supported

Table 4: Detour patterns support

Nr	Pattern	Rating	Rationale
Auto-start patterns			
36	Commencement on Creation	−	Not supported. Task instances must be explicitly started by an owner
37	Commencement on Allocation	−	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
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
Multiple resources patterns			
42	Simultaneous Execution	+	Directly supported
43	Additional Resources	−	Not supported. There can only be one resource for a task instance

Table 5: Auto-start, visibility and multiple resource patterns support

4 Observations

Significant observations that can be drawn about the BPEL4People and WS-HumanTask proposals at this stage are as follows:

- 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;
- There are a number of distinct ways in which manual tasks (i.e. those undertaken by human resources) 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 in a distinct process definition) which are coordinated by a PeopleActivity node in a BPEL process;
- 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 or updating an associated data element);
- 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;
- 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;
- 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;
- There is no provision for imposing an authorisation framework over the tasks in a process to limit the potential range of resource to whom they can be directed and that are able to ultimately execute them;
- There is minimal support for restricting the range of actions that a resource can initiate in regard to a task (e.g. delegation, reallocation etc.). Some of these options can be restricted at task level (eg. skipping, delegation) but not on a per-resource basis;
- There are no facilities for optimising work item throughput (e.g. auto-starting tasks, piled execution etc.);
- There is no ability to impose restrictions or format changes on work items in individual resource's work lists on a system-wide basis; and
- The facilities available for potentially restricting work item visibility are unclear.

Disclaimer

We, the authors and the associated institution (TU/e), assume no legal liability or reponsibility for the accuracy and completeness of any information about WS-BPEL, BPEL4People, WS-HumanTask and other related standards and proposals, contained in this paper. However, all possible efforts have been made to ensure that the results presented are, to the best of our knowledge, up-to-date and correct.

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] 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`.

[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. 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`.

[4] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services version 1.1. Technical report, 2003. `http://xml.coverpages.org/BPELv11-May052003Final.pdf`.

[5] 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`.

[6] 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`.

[7] 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.

[8] 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`.

[9] N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow data patterns (revised version). Technical report, Queensland University of Technology, Brisbane, Australia, 2004. `http://www.bpmcenter.org`.

[10] N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow resource patterns. Technical report, Eindhoven University of Technology, Eindhoven, Netherlands, 2004. http://is.tm.tue.nl/research/patterns/download/Resource%20Patterns%20BETA%20TR.pdf.

[11] 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.

A Detailed Evaluation Results

WRP-1: Direct Distribution

Description: The ability to specify at design time the identity of the resource that will execute a task.

Rating: +

Rationale: Support for this pattern is exemplified by the following code fragment. In this instance the GoUpTheHill task is offered to Jack and Jill.

```
<htd:task name="GoUpTheHill">
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:users>
              <htd:user>Jack</htd:user>
              <htd:user>Jill</htd:user>
            </htd:users>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-2: Role-Based Distribution

Description: The ability to specify at design time that a task can only be executed by resources which correspond to a given role.

Rating: +

Rationale: Support for this pattern is exemplified by the following code fragment. It illustrates the distribution of the GoRoundTheMulberryBush task to users who are part of the NurseryCharacters role. If the NurseryCharacters role has multiple users, then the task is offered to each of these users. If it only contains one user, then it is allocated to that user.

```
<htd:task name="GoRoundTheMulberryBush">
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:groups>
```

```
            <htd:group>NurseryCharacters</htd:group>
          </htd:groups>
        </htd:organizationalEntity>
      </htd:literal>
    </htd:from>
  </htd:potentialOwners>
</htd:peopleAssignments>
</htd:task>
```

WRP-3: Deferred Distribution

Description: The ability to defer specifying the identity of the resource that will execute a task until runtime.

Rating: +

Rationale: Support for this pattern is exemplified by the following code fragment. The RunMarathon task is distributed to those users who are identified as runners in the atheletelist parameter to the task. This allows the identification of potential users to whom the task will be distributed to be deferred until runtime.

```
<htd:task name=''RunMarathon''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        <htd:getInput(‘‘athletelist")/runners>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-4: Authorisation

Description: The ability to specify the range of resources that are authorised to execute a task.

Rating: +/−

Rationale: There are basic capabilities for defining the set of resources to whom a task may be delegated as indicated by the code fragment below, which restricts work items for the GoRoundTheMulberryBush task being delegated to users other than those originally intended to execute it but execution restrictions cannot be imposed on a more general basis e.g. limit the users who can reallocate this task.

```
<htd:task name=''GoRoundTheMulberryBush''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
```

```
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:groups>
              <htd:group>NurseryCharacters</htd:group>
            </htd:groups>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
    <htd:delegation potentialDelegatees=''potentialOwners''>
  </htd:delegation>
</htd:task>
```

WRP-5: Separation of Duties

Description: The ability to specify that two tasks must be allocated to different resources in a given workflow case.

Rating: +

Rationale: This pattern is directly supported in WS-HumanTask via the `<excludedOwners>` element which forms part of the task assignment specification. In the code fragment below the task GoRoundTheMulberryBush can be distributed to any of the NurseryCharacters except for Jack and Jill.

```
<htd:task name=''GoRoundTheMulberryBush''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:groups>
              <htd:group>NurseryCharacters</htd:group>
            </htd:groups>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
    <htd:excludedOwners>
       <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:users>
              <htd:user>Jack</htd:user>
              <htd:user>Jill</htd:user>
            </htd:users>
```

```
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-6: Case Handling

Description: The ability to allocate the work items within a given workflow case to the same resource.

Rating: –

Rationale: Not supported. There is no mechanism for distributing all of the work items in a process instance at the commencement of the process instance.

WRP-7: Retain Familiar

Description: Where several resources are available to undertake a work item, the ability to allocate a work item within a given workflow case to the same resource that undertook a preceding work item.

Rating: +

Rationale: This pattern is supported by setting the `<potentialOwners>` element to the identity of the user that undertook a previous task in the process. The following code fragment sets the (single) owner of the AllFallDown task to be the same as the user that undertook the (preceding) FormCircle task.

```
<htd:task name=''AllFallDown''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        htd:getActualOwner(‘‘FormCircle")
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-8: Capability-based Distribution

Description: The ability to offer or allocate instances of a task to resources based on specific capabilities that they possess.

Rating: –

Rationale: There are no mechanisms for specifying additional information about users in a BPEL process.

WRP-9: History-based Distribution

Description: The ability to offer or allocate work items to resources on the basis of their previous execution history.

Rating: +/–

Rationale: The ability exists to query the initiator and actual owner of previous tasks in the same process instance and to use this information when distributing tasks, however it is unclear what the semantics of this are where (1) the (preceding) task instance has not yet completed, (2) the task that is being queried has run multiple times in the same process instance (e.g. as part of a loop or multiple instance task) or (3) the task instance(s) that is of interest ran in a distinct process instance. For purposes of illustration, a code fragment (similar to that for the Retain Familiar pattern) illustrating the simple activity of distributing a task to the same user that initiated a preceding task is shown below.

```
<htd:task name=''AllFallDown''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        htd:getTaskInitiator(‘‘FormCircle")
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-10: Organisational Distribution

Description: The ability to offer or allocate instances of a task to resources based on their position within the organisation and their relationship with other resources.

Rating: +/–

Rationale: BPEL4People only establishes a minimalistic organisational model denoting roles and group membership although both of these criteria can be used as the basis of work distribution directives. More complex organisational concepts e.g. organisational hierarchy, reporting lines are not supported. An example of distributing a task to members of a specific organisational group (EarlyRisers) is illustrated below.

```
<htd:task name=''RingaRoses''>
  <htd:peopleAssignments>
    <htd:potentialOwners>
```

```
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:groups>
              <htd:group>EarlyRisers</htd:group>
            </htd:groups>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-11: Automatic Execution

Description: The ability for an instance of a task to execute without needing to utilise the services of a resource.

Rating: +

Rationale: This is the default means of executing a task in BPEL where the BPEL4People/WS-HumanTask extensions are not employed.

WRP-12: Distribution by Offer – Single Resource

Description: The ability to offer a work item to a selected individual resource.

Rating: –

Rationale: There is no mechanism for offering a task instance to a single user on a non-binding basis. Where the `<potentialOwners>` attribute for a task instance corresponds to a single user, then it is assumed to be allocated to that user.

WRP-13: Distribution by Offer – Multiple Resources

Description: The ability to offer a work item to a group of selected resources.

Rating: +

Rationale: Where the `<potentialOwners>` attribute for a task instance corresponds to multiple users, it is offered to all of those users on a non-binding basis. An example of this is illustrated below. The CatchTheBus task is offered to the users Mary, Mungo and Midge.

```
<htd:task name="CatchTheBus">
  <htd:peopleAssignments>
    <htd:potentialOwners>
```

```
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:users>
              <htd:user>Mary</htd:user>
              <htd:user>Mungo</htd:user>
              <htd:user>Midge</htd:user>
            </htd:users>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-14: Distribution by Allocation – Single Resource

Description: The ability to directly allocate a work item to a specific resource for execution.

Rating: +

Rationale: Where the `<potentialOwners>` attribute for a task instance corresponds to a single user, the it is assumed to be allocated to that user. An example of this is shown below. The SitOnTheWall task is allocated to the user Humpty.

```
<htd:task name="SitOnTheWall">
  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from>
        <htd:literal>
          <htd:organizationalEntity>
            <htd:users>
              <htd:user>Humpty</htd:user>
            </htd:users>
          </htd:organizationalEntity>
        </htd:literal>
      </htd:from>
    </htd:potentialOwners>
  </htd:peopleAssignments>
</htd:task>
```

WRP-15: Random Allocation

Description: The ability to offer or allocate work items to suitable resources on a random basis.

Rating: –

Rationale: There is no direct means of randomly selecting a user from a list of user to whom a work item should be allocated. Workarounds based on extensions to the XPath 2.0 functions can be conceived but these are not part of the BPEL4People/WS-HumanTask functionality at present.

WRP-16: Round Robin Allocation

Description: The ability to allocate a work item to available resources on a cyclic basis.

Rating: –

Rationale: There is no mechanism for distributing tasks to a group of users on a cyclic basis.

WRP-17: Shortest Queue

Description: The ability to allocate a work item to the resource that has the least number of work items allocated to it.

Rating: +/–

Rationale: Conceptually this should be possible by summing the size of the work lists for each of the potentialOwners for a task and then selecting the shortest. However its unclear if the query function will allow the task lists of multiple users to be accessed.

WRP-18: Early Distribution

Description: The ability to advertise and potentially allocate work items to resources ahead of the moment at which the work item is actually enabled for execution.

Rating: –

Rationale: There is no means of distributing work items ahead of the time that they are enabled.

WRP-19: Distribution on Enablement

Description: The ability to advertise and allocate work items to resources at the moment they are enabled for execution.

Rating: +

Rationale: Work items are distributed as soon as they are enabled.

WRP-20: Late Distribution

Description: The ability to advertise and allocate work items to resources after the work item has been enabled.

Rating: −

Rationale: There is no means of delaying the distribution of a task to users once it has been enabled.

WRP-21: Resource-Initiated Allocation

Description: The ability for a resource to commit to undertake a work item without needing to commence working on it immediately.

Rating: +/−

Rationale: Supported via the claim function providing the work item is offered to more than one user. It is automatically started if it is only offered to one user.

WRP-22: Resource-Initiated Execution − Allocated Work Item

Description: The ability for a resource to commence work on a work item that is allocated to it.

Rating: +

Rationale: A user can initiate a work item that is on their work list in the *Reserved* state via the start operation.

WRP-23: Resource-Initiated Execution − Offered Work Item

Description: The ability for a resource to select a work item offered to it and commence work on it immediately.

Rating: +

Rationale: A user can initiate a work item that is on their work list in the *Ready* state via the start operation.

WRP-24: System-Determined Work Queue Content

Description: The ability of the workflow engine to order the content and sequence in which work items are presented to a resource for execution.

Rating: –

Rationale: There are no mechanisms for imposing an ordering or specific content requirements on a user's task list by the system.

WRP-25: Resource-Determined Work Queue Content

Description: The ability for resources to specify the format and content of work items listed in the work queue for execution.

Rating: +

Rationale: The simple and advanced query functions provide the ability for users to restrict and format the content of their worklists.

WRP-26: Selection Autonomy

Description: The ability for resources to select a work item for execution based on its characteristics and their own preferences.

Rating: +

Rationale: There are no limitations placed on the work items that can be started by users.

WRP-27: Delegation

Description: The ability for a resource to allocate a work item previously allocated to it to another resource.

Rating: +

Rationale: Directly supported via the delegate function.

WRP-28: Escalation

Description: The ability of the workflow system to offer or allocate a work item to a resource or group of resources other than those it has previously been offered or allocated to in an attempt to expedite the completion of the work item.

Rating: +

Rationale: Escalations can be specified for tasks as illustrated by the following code fragment. If the SitOnTheWall task is not completed within one day of commencement, it is reassigned to the user theManager.

```
<htd:task name=''SitOnTheWall''>
  <htd:deadlines>
    <htd:completionDeadline>
      <htd:for>P1D</htd:for>
      <htd:escalation name="deadlineExpired">
      <htd:peopleAssignments>
        <htd:potentialOwners>
          <htd:from>
            <htd:literal>
              <htd:organizationalEntity>
                <htd:users>
                  <htd:user>theManager</htd:user>
                </htd:users>
              </htd:organizationalEntity>
            </htd:literal>
          </htd:from>
        </htd:potentialOwners>
      </htd:peopleAssignments>
    </htd:completionDeadline>
  </htd:deadlines>
</htd:task>
```

WRP-29: Deallocation

Description: The ability of a resource (or group of resources) to relinquish a work item which is allocated to it and make it available for allocation to another resource or group of resources.

Rating: +

Rationale: Directly supported via the release function.

WRP-30: Stateful Reallocation

Description: The ability of a resource to allocate a work item to another resource without loss of state data.

Rating: +

Rationale: A work item can be reallocated to another set of users, groups or the entities specified by the results of a query via the forward function.

WRP-31: Stateless Reallocation

Description: The ability for a resource to reallocate a work item currently being executed to another resource without retention of state.

Rating: –

Rationale: There is no means of reallocating a work item and resetting its state when doing so.

WRP-32: Suspension/Resumption

Description: The ability for a resource to suspend and resume execution of a work item.

Rating: +

Rationale: Supported via the suspend and resume functions.

WRP-33: Skip

Description: The ability for a resource to skip a work item allocated to it and mark the work item as complete.

Rating: +

Rationale: Directly supported via the skip operation.

WRP-34: Redo

Description: The ability for a resource to redo a work item that has previously been completed in a case.

Rating: –

Rationale: There is no mechanism for distributing and/or executing a work item which has already been completed.

WRP-35: Pre-Do

Description: The ability for a resource to execute a work item ahead of the time that it has been offered or allocated to resources working on a given case.

Rating: –

Rationale: There is no mechanism for distributing and/or executing a work item which has not yet been enabled.

WRP-36: Commencement on Creation

Description: The ability for a resource to commence execution on a work item as soon as it is created.

Rating: –

Rationale: There is no support for automatically starting work items during distribution.

WRP-37: Commencement on Allocation

Description: The ability to commence execution on a work item as soon as it is allocated to a resource.

Rating: –

Rationale: There is no support for automatically starting work items during distribution.

WRP-38: Piled Execution

Description: The ability of the workflow system to initiate the next instance of a workflow task (perhaps in a different case) once the previous one has completed.

Rating: –

Rationale: There is no support for automatically starting work items during distribution.

WRP-39: Chained Execution

Description: The ability of the workflow engine to automatically start the next work item in a case once the previous one has completed.

Rating: –

Rationale: There is no support for automatically starting subsequent work items once a preceding work item completes.

WRP-40: Configurable Unallocated Work Item Visibility

Description: The ability to configure the visibility of unallocated work items by workflow participants.

Rating: +/−

Rationale: This capability does not seem to be possible using the simple query function which only returns tasks for the requesting user (and not all users). The advanced query function provides a more general mechanism for retrieving task data however it is not clear how this operates across process instances nor how user authorizations are defined. Also support for the function is not a mandatory part of the standard.

WRP-41: Configurable Allocated Work Item Visibility

Description: The ability to configure the visibility of allocated work items by workflow participants.

Rating: +/−

Rationale: This capability does not seem to be possible using the simple query function which only returns tasks for the requesting user (and not all users). The advanced query function provides a more general mechanism for retrieving task data however it is not clear how this operates across process instances nor how user authorizations are defined. Also support for the function is not a mandatory part of the standard.

WRP-42: Simultaneous Execution

Description: The ability for a resource to execute more than one work item simultaneously.

Rating: +

Rationale: There is nothing that precludes a user have more than one work item in their work list with an *InProgress* status.

WRP-43: Additional Resources

Description: The ability for a given resource to request additional resources to assist in the execution of a work item that they are currently undertaking.

Rating: −

Rationale: The distribution of work items is based on the premise that they are executed by a single user.