

Workflow Patterns



Workflow Resource Patterns

Download of the resource patterns paper:

N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst.

[Workflow Resource Patterns](#). (PDF, 418 Kb).

BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.

Introduction

Workflow systems seek to provide an implementation vehicle for complex, recurring business processes. Notwithstanding this common objective, there are a variety of distinct features offered by commercial workflow management systems. These differences result in significant variations in the ability of distinct tools to represent and implement the plethora of requirements that may arise in contemporary business processes. Many of these requirements recur quite frequently during the requirements analysis activity for workflow systems and abstractions of these requirements serve as a useful means of identifying the key components of workflow languages. Previous work has identified a number of Workflow Control-flow Patterns and Workflow Data Patterns, which characterize the range of control flow and data constructs that might be encountered when modelling and analysing workflows. In this body of work, we describe a series of Workflow Resource Patterns that aim to capture the various ways in which resources are represented and utilized in workflows. By delineating these Patterns in a form that is independent of specific workflow technologies and modelling languages, we are able to provide a comprehensive treatment of the resource perspective and we subsequently use these Patterns as the basis for a detailed comparison of a number of commercially available workflow management systems and business process modelling languages.

Before you view the different resource patterns, you may wish to examine one of the following options to gain a better understanding of the work presented to you:

- For a general explanation of terms, go to the [resource modelling](#) web page.
- For an introduction to the concepts used in this body of work, go to the [workflow structure](#) web page.
- To read about the lifecycle of a work item, go to the [work distribution of resources](#) web page.

Creation Patterns

Creation Patterns correspond to limitations on the manner in which a work item may be executed. They are specified at design time, usually in relation to a workflow task, and serve to restrict the range of resources that can undertake work items that correspond to the task. They also influence the manner in which a work item can be matched with a resource that is capable of undertaking it. The essential rationale for Creation Patterns is that they provide a degree of clarity about how a work item should be handled after creation during the offering and allocation stages prior to it being executed. This ensures that the operation of a workflow process conforms with

its intended design principles and operates as efficiently and deterministically as possible. In terms of the work item lifecycle, Creation Patterns come into effect at the time a work item is created. This state transition occurs at the beginning of the work item lifetime and is illustrated by the bold arrow in Figure 6.

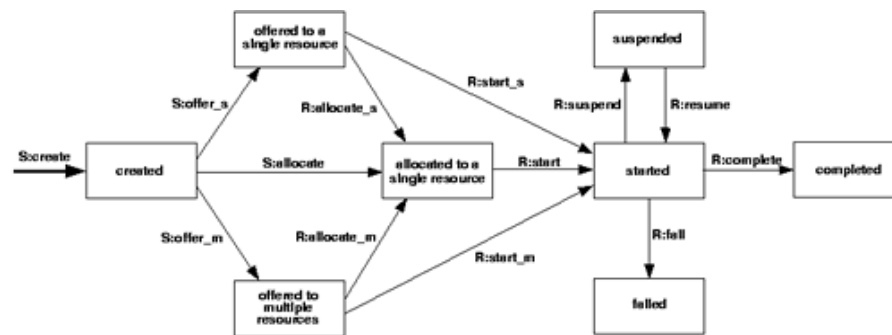


Figure 6: Creation Patterns

As Creation Patterns are specified at design time, they usually form part of the process model that defines a workflow process.

1. [Direct Allocation](#)
2. [Role-Based Allocation](#)
3. [Deferred Allocation](#)
4. [Authorisation](#)
5. [Separation of Duties](#)
6. [Case Handling](#)
7. [Retain Familiar](#)
8. [Capability-Based Allocation](#)
9. [History-Based Allocation](#)
10. [Organisational Allocation](#)
11. [Automatic Execution](#)

Push Patterns

Push Patterns characterise situations where newly created work items are proactively offered or allocated to resources by the workflow system. These may occur indirectly by advertising work items to selected resources via a shared work list or directly with work items being allocated to specific resources. In both situations however, it is the workflow engine that takes the initiative and causes the distribution process to occur. Figure 8 illustrates (as bold arcs) the potential state transitions associated with push-based distribution:

- S:offer_s corresponds to a work item being offered to a single resource
- S:offer_m corresponds to a work item being offered to multiple resources (one of which will ultimately execute it)
- S:allocate corresponds to a work item being directly allocated to a resource immediately after it has been created

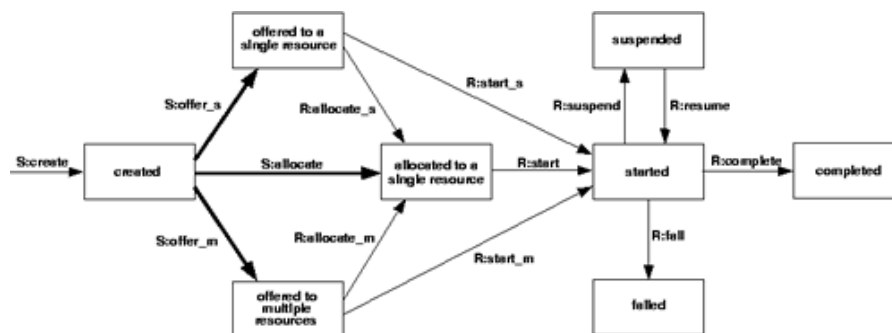


Figure 8: Push Patterns

Nine push Patterns have been identified. These divide into three distinct groups. The first three Patterns identify the actual manner of work distribution - whether the workflow system offers the work item to a single resource, to multiple resources or whether it allocates it directly to a single resource (footnote 1). These Patterns correspond directly to the bold arcs in Figure 8. The second group of Patterns relate to the means by which a resource is selected to undertake a work item where there are multiple possible resources identified. Three possible strategies are described - random allocation, round robin allocation and shortest queue. These Patterns correspond to alternate ways in which the S:offer_s and S:allocate transitions may occur. The final three Patterns identify the timing of the distribution process and in particular the relationship between the availability of a work item for offering/allocation to resources and the time at which it commences execution. Three variants are possible - work items are offered/allocation before they have commenced (early distribution), after they have commence (late distribution) or the two events are simultaneous (distribution on enablement). These Patterns do not have a direct analogue in Figure 8 but relate to the time at which the S:offer_s, S:offer_m and S:allocate transitions may occur with respect to the work item's readiness to be executed (i.e. already started, immediate start or subsequent start).

12. [Distribution by Offer - Single Resource](#)
13. [Distribution by Offer - Multiple Resources](#)
14. [Distribution by Allocation - Single Resource](#)
15. [Random Allocation](#)
16. [Round Robin Allocation](#)
17. [Shortest Queue](#)
18. [Early Distribution](#)
19. [Distribution on Enablement](#)
20. [Late Distribution](#)

Pull Patterns

Pull Patterns correspond to the situation where individual resources are made aware of specific work items, that require execution, either via a direct offer from the workflow system or indirectly through a shared work list. The commitment to undertake a specific task is initiated by the resource itself rather than the workflow system. Generally this results in the work item being placed on the specific work list for the individual resource for later execution although in some cases, the resource may elect to commence execution on the work item immediately. The various state transitions associated with Pull Patterns are illustrated in Figure 9:

- R:allocate_s corresponds to a work item offered to a single resource that the resource has indicated it will commit to executing at some future time
- R:allocate_m corresponds to a work item offered to multiple resources that one of the resources has indicated it will commit to executing at some future time.

the work item is deemed to be allocated to that resource and is no longer available to the other resources to which it was offered

- R:start_s corresponds to a work item which has been offered to a single resource being started by that resource
- R:start_m corresponds to a work item which has been offered to multiple resources being started by one of those resources
- R:start corresponds to a work item which has been allocated to a single resource being started by that resource.

Six Pull Patterns have been identified. These divide into two distinct groups. The first three Patterns identify the specifics of the actual "pull" action initiated by the resource, with a particular focus on the work item state before and after the interaction. These Patterns correspond directly to the bold arcs in Figure 9. The second group of Patterns focus on the sequence in which the work items are presented to the resource and the ability of the workflow system and the individual resource to influence the sequence and manner in which they are displayed. The final Pattern in this group illustrates the degree of freedom that the resource has in selecting the next work item to execute. These Patterns do not have a direct analogue in Figure 9 but apply to all of the "pull" transitions illustrated as bold arcs.

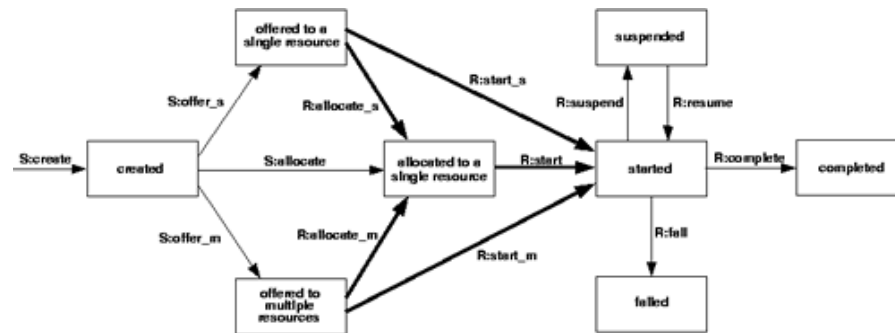


Figure 9: Pull Patterns

Note that the distinction between Push and Pull Patterns is identified by the initiator of the various transitions. For the Push Patterns in Figure 8, the state transitions for work items are all triggered by the workflow system, whereas in Figure 9 which denotes Pull Patterns, the transitions are initiated by individual resources. Other characteristics of interest which ultimately lead to additional Pull Patterns, relate to whether the resource has the ability to reorder the work sequence or it is determined by the workflow system, and whether a resource can select which work item they wish to commence next from those on its work queue.

21. [Resource-Initiated Allocation](#)
22. [Resource-Initiated Execution - Allocated Work Item](#)
23. [Resource-Initiated Execution - Offered Work Item](#)
24. [System-Determined Work Queue Content](#)
25. [Resource-Determined Work Queue Content](#)
26. [Selection Autonomy](#)

Detour Patterns

Detour Patterns refer to situations where work allocations that have been made for resources are interrupted either by the workflow system or at the instigation of the resource. As a consequence of this event, the normal sequence of state transitions for a workflow item is varied. The range of possible scenarios for Detour Patterns are illustrated in Figure 10.

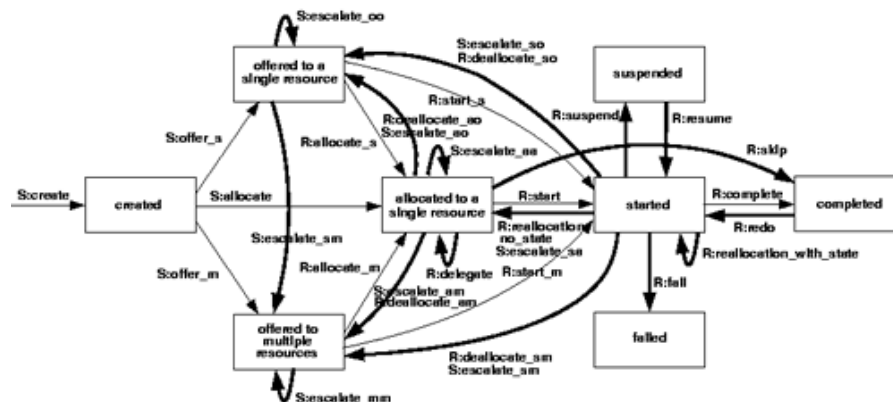


Figure 10: Detour Patterns

There are a number of possible impacts on a work item, depending on its current state of progression and whether the detour was initiated by the resource with which the work item was associated or by the workflow system. These include:

- *delegation* - where a resource allocates a work item previously allocated to it to another resource
- *escalation* - where the workflow system attempts to progress a work item that has stalled by offering or allocating it to another resource
- *de-allocation* - where the system makes a previously allocated or started work item available for offer and subsequent allocation
- *reallocation* - where a resource allocates a work item that it has started to another resource
- *suspension/resumption* - where a resource temporarily suspends execution of a work item or recommences execution of a previously suspended work item
- *skipping* - where a resource elects to skip the execution of a work item allocated to it
- *redo* - where a resource repeats execution of a work item completed earlier
- *pre-do* - where a resource executes a work item that is ahead of the current execution point of a workflow case

Each of these actions relate to one or more transitions in Figure 10 and corresponds to specific Patterns below.

27. [Delegation](#)
28. [Escalation](#)
29. [Deallocation](#)
30. [Stateful Reallocation](#)
31. [Stateless Reallocation](#)
32. [Suspension-Resumption](#)
33. [Skip](#)
34. [Redo](#)
35. [Pre-Do](#)

Auto-Start Patterns

Auto-start Patterns relate to situations where execution of work items is triggered by specific events in the lifecycle of the work item or the related process definition. Such events may include the creation or allocation of the work item, completion of another instance of the same work item or a work item that immediately precedes the one in question. The state transitions associated with these Patterns are illustrated by bold arcs in Figure 11.

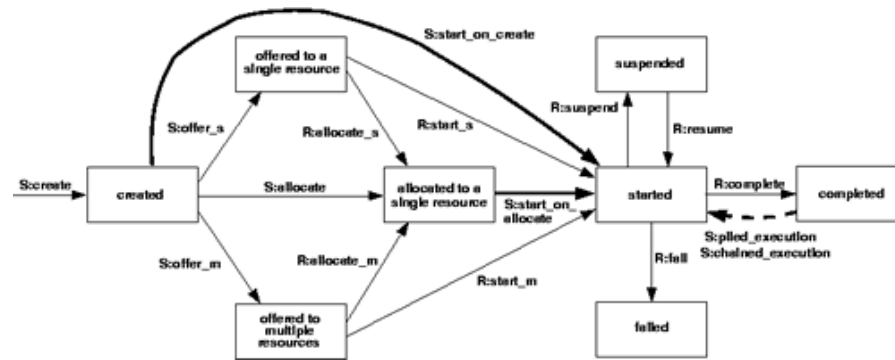


Figure 11: Auto-start Patterns

- 36. [Commencement on Creation](#)
- 37. [Commencement on Allocation](#)
- 38. [Piled Execution](#)
- 39. [Chained Execution](#)

Visibility Patterns

Visibility Patterns classify the various scopes in which work item availability and commitment are able to be viewed by workflow resources. They give a indication of how open to scrutiny the operation of a workflow system is.

- 40. [Configurable Unallocated Work Item Visibility](#)
- 41. [Configurable Allocated Work Item Visibility](#)

Multiple Resource Patterns

The focus of this body of work is on the situations where there is a one-to-one correspondence between the resources and work items in a given allocation or execution. In other words, resources cannot work on different work items simultaneously and it is not possible that multiple resources work on the same work item. In situations where people are not restricted by information technology, there is often a many-to-many correspondence between the resources and work items in a given allocation or execution. Therefore, it may be desirable to support this using workflow technology. Here we discuss Patterns relaxing the one-to-one correspondence between resources and work items that we have assumed previously. Let us first consider the one-to-many situation, i.e., resources can work on different work items simultaneously. This is a fairly simple requirement, supported by most systems.

- 42. [Simultaneous Execution](#)
- 43. [Additional Resources](#)

Pattern R-SE (Simultaneous Execution) is easy to support and contemporary systems support this one-to-many correspondence between the resources and work items in a given allocation or execution. Unfortunately, it is more difficult to support a many-to-one correspondence, i.e., multiple resources working on the same work item. This is a pity since for more complicated activities people tend to work in teams and collaborate to jointly executed work items. Given the limited support of today's workflow systems, we provide only one pattern implying a many-to-one correspondence.

Disclaimer

We, the authors and the associated institutions, assume no legal liability or responsibility for the accuracy and completeness of any product-specific information contained in this body of work. All reasonable efforts have been made to ensure that the results presented are, to the best of our knowledge, up to date and correct.

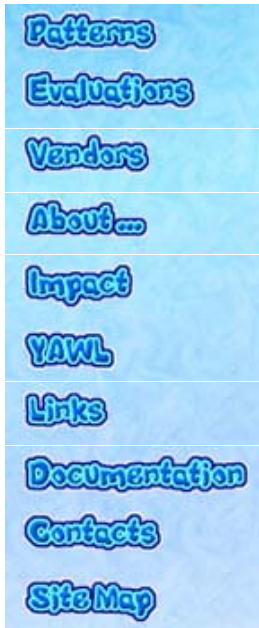
© 2007 *Workflow Patterns Initiative*

0025926

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 1 (Direct Allocation)

[FLASH animation of Direct Allocation pattern](#)

Description

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

Example

the *Fix Bentley* task must only be undertaken by *Fred*

Related Patterns

[R-D \(Delegation\)](#)

Motivation

Direct allocation offers the ability for a workflow designer to precisely specify the identity of the resource to which instances of each task will be allocated at runtime. This is particularly useful where it is known that a task can only be effectively undertaken by a specific resource as it prevents the problem of unexpected or non-suitable resource allocations arising at runtime by ensuring work items are routed to specific resources, a feature that is particularly desirable for critical tasks.

Implementation

Most workflow engines offer some form of support for direct allocation of tasks to specific resources. In most cases, the allocation is to a single resource, however Staffware allows a work item to be allocated to a series of specific resources (achieved by specifying the names of multiple resources for potential allocation) and at runtime, the work item is routed to all of these resources and each of them is required to release it before the work item can be deemed to have finished and the case can progress.

Issues

One of the main drawbacks of this approach to resource allocation is that it effectively defines a static binding of all work items associated with a task to a single resource. This removes much of the advantage associated with the use of workflow technology for managing work distribution as the workflow engine is offered little latitude for optimising the allocation of work items in this situation.

Solutions

This is no real solution to this problem although the use of deadline and escalation mechanisms offer ways of ensuring that situations are detected where a specific resource becomes overloaded and cannot deal with its assigned workload in a reasonable timeframe.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported
Websphere MQ Workflow	3.4	+	Directly supported
FLOWer	3.0	+	Where there is a 1-1 correspondence between a specific role and an actual resource
COSA	4	+	Directly supported
iPlanet	3.1	+	Directly supported
BPMN	1.0	+	Through the notion of Pool, as it can be used to denote a specific business entity, e.g. a specific actor. The notion of Lane could be used for this purpose as well
UML	2.0	+	Directly supported by basing a partition on a single object (resource) instance
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. The resources can be specified statically or dynamically. The organization structure, together with resources, their identities, and other characteristics can be specified via JAZN admintool. When adding a task to the process model, the workflow wizard allows specifying a single user, a set of the users, or a group. Each of the task parameters specified by means of the wizard can be modified later in the designer view or in the corresponding BPEL code.

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. Workflow has an integrated organisational model 2. Work items can be allocated to a specific resource identified in the organisational model 	<ol style="list-style-type: none"> 1. Work items can be routed to a specific resource by some means (e.g. queue, work list etc.)

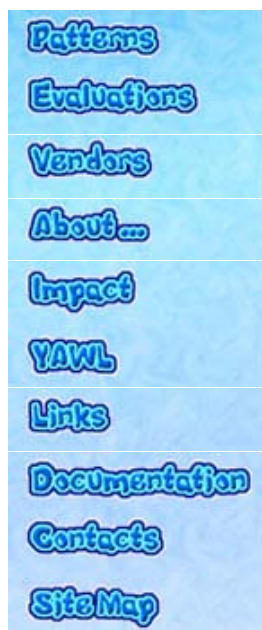
© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 2 (Role-Based Allocation)

[FLASH animation of Role-Based Allocation pattern](#)

Description

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

Example

Instances of the *Approve Travel Requisition* task must be executed by a *Manager*.

Related Patterns

[R-RA \(Authorisation\)](#), [R-OR \(Organisational Allocation\)](#).

Motivation

Perhaps the most common approach to work item allocation within workflow systems, role-based allocation offers the means for the workflow engine to route work items to suitably qualified resources at run-time. The decision as to which resource actually receives a given work item is deferred until the moment at which it becomes "runnable" and requires a resource allocation in order for it to proceed. The advantage offered by role-based allocation (over other work item allocation schemes) is that roles can be defined for a given workflow process that define the various classes of available resources to undertake work items. Task definitions within the process model can nominate the specific role to which they should be routed, however the actual population of individual roles does not need to occur until run-time.

Implementation

All of the workflow systems examined support role-based allocation in some form. Generally roles serve as groupings of resources with similar characteristics or authority and provides a means of decoupling the routing of a work item from that of resource management. The most restrictive approach to role definition occurs in Staffware where only one resource can be identified for each role although it is possible to specify multiple roles when defining the routing of work item. WebSphere MQ allows multiple resources to be specified for each role and also multiple roles to be used when routing a work item. iPlanet supports roles in a similar way although the actual mechanism used for work item distribution takes the form of an expression which includes the various roles rather than simply listing the roles to which the work item will be forwarded. COSA also uses roles as a grouping mechanism for resources and allows them to be used as a routing mechanism for work items, however where a work item is routed to multiple resources, it appears on a shared (group) work queue rather than being replicated on the work lists of individual resources. COSA provides support for explicitly representing quite complex organisational structures and work

allocation mechanisms by allowing role, organisational and authorisation hierarchies to be distinctly modelled and drawn together where required in the distribution functions for individual work items. FLOWer supports multiple users per role and allows a user to play different roles in distinct cases. Roles serve as the main basis of work item distribution although resources have a reasonable degree of autonomy in selecting the work items (and cases) that they will undertake rather than having work items directly assigned to them.

Issues

In some workflow systems, the concepts of roles and groups are relatively synonymous. Roles serve as an abstract grouping mechanism (i.e. not just for resources with similar characteristics or authority, but also for identification of organisational units e.g. teams, departments etc.) and provide a means of distributing work across a number of resources simultaneously. One difficulty that arises with this use of roles occurs where the intention is to offer a work item to several resources with the expectation that they will all work on it.

Solutions

Staffware provides support for this style of work allocation. It operates in much the same way as role-based allocation with groups being identified within the workflow system consisting of several resources. Individual resources may belong to more than one group (unlike the situation with roles) and a task within the process model can be specified as requiring routing to a specific group at run-time. However the operation of group-based allocation differs from role-based allocation at run-time with a work item that is allocated to a group being visible to all of the resources in the group and not specifically (and privately) assigned to one of them during the allocation process. Group-based allocation is non-deterministic with respect to resources and the work item is ultimately allocated to the first resource in the group that commences work on it. From his point, none of the other resources in the group can execute it although it remains in the work queue of all the resources until it has been complete. None of the other workflow engines examined provide support for this approach to work allocation.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported
Websphere MQ Workflow	3.4	+	Directly supported
FLOWer	3.0	+	Organisational hierarchy is modelled in terms of roles
COSA	4	+	Directly supported
iPlanet	3.1	+	Directly supported
BPMN	1.0	+	Through the notion of Pool, as it can also be used to denote a more general business role. The notion of Lane could be used as for this

			purpose as well
UML	2.0	+	Directly supported via classifiers on partitions.
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. The organizational structure, including the roles assigned to the users, can be specified in the jazn-data.xml and user-properties.xml, the content of which is visible for the designer creating a process model. Specifying one of the existing roles as a task assignee would make a work item visible in the worklists of users with the corresponding role. Note that Oracle BPEL PM makes not distinction between groups and roles.

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none">1. Workflow has an integrated organisational model that directly supports roles.2. Routing a work item via role-based allocation involves directing the work item to one specific resource that participates in the nominated role.	<ol style="list-style-type: none">1. Roles are not directly supported by the workflow engine but the same effect can be achieved via other mechanisms.

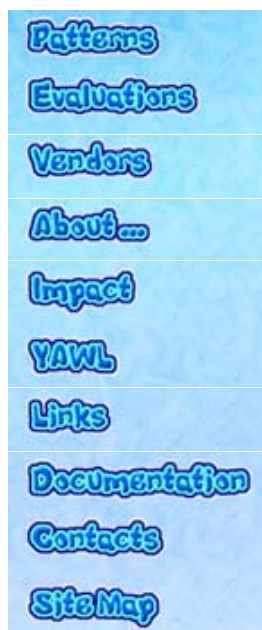
© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 3 (Deferred Allocation)

[FLASH animation of Deferred Allocation pattern](#)

Description

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

Example

During execution of a case, instances of the *Access Damage* task will be executed by the resource named in the *next_resource* field.

Related Patterns

[R-DA \(Direct Allocation\)](#), [R-RBA \(Role-based Allocation\)](#)

Motivation

Deferred allocation takes the notion of indirect resource allocation one step further and allows the workflow designer to defer the need to identify the resource for a specific task (or work items corresponding to the task) until run-time. One means of achieving this is to nominate a data field from which the identity of the resource to which a work item should be routed can be determined at runtime. The resource identity can be changed dynamically during workflow execution by updating the value of the data field, thus varying the resource allocation of future work items which are contingent on it.

Implementation

This approach to resource allocation is generally achieved by associating the field name which contain the resource identity with the task at design time. In order to facilitate this, the field needs to be a data element within the scope of the task at runtime - usually a case level data element. It is possible that more than one data element (and hence more than one resource) could be taken into account when deciding on the allocation at runtime. Both Staffware and WebSphere MQ directly support this Pattern.

Issues

Two significant issues arise in implementing this Pattern:

- Determining whether the value in the data field relates to a specific resource, group or role name. This determination is important as it varies the approach taken to resource allocation.
- Ensuring that the data field contains a valid resource name.

Solutions

The first of these issues is usually addressed by ensuring that the names used for specific resources, groups and roles are disjoint. This means that a name cannot be used in more than one context and hence there is no potential for ambiguity at runtime. The second issue is more problematic as a data element can potentially contain any value and there is no means of ensuring that it corresponds to an actual resource in the workflow or to specify the action to take when the resource name is invalid.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported
Websphere MQ Workflow	3.4	+	Directly supported
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. It allows specifying a task assignee by means of an XPath expression which is evaluated at run time

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. Workflow engine provides support for nominating the allocation strategy for a work item at run-time. 2. Identification for the resource allocation is derived from a data fields. 3. Direct, group or role-based allocation is supported. 	<ol style="list-style-type: none"> 1. Similar effect can be achieved through programmatic extensions

Workflow Patterns



Pattern 4 (Authorisation)

[FLASH animation of Authorisation pattern](#)

Description

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

Example

only the *Finance Director*, *Senior Loans Manager* and *Financial Accountant* are authorised to execute instances of the *Finalise Loan* task.

Related Patterns

[R-CBA \(Capability-based Allocation\)](#), [R-SOD \(Separation of Duties\)](#)

Motivation

Through the specification of authorisations on task definitions, it is possible to define a security framework over a workflow implementation that is independent of the way in which work items are actually routed at runtime. By defining authorisations on individual tasks, the range of resources that can access details of a work item or execute it can be restricted. This ensures that unexpected events that may arise during workflow execution (e.g. work item delegation by a resource or reallocation to another resource outside of the usual workflow operation) do not lead to unexpected resources being able to undertake work items.

Implementation

COSA is the only workflow engine observed that implements the notion of task authorisation as a distinct concept to that of task distribution. It treats authorisation and distribution of tasks in the same way in the design time model and provides facilities for defining the resources, groups and roles that are authorised to execute a task and also those to which it can be allocated. FLOWer uses roles as the main basis for case and work item distribution. Roles are organised as hierarchies and only resources that directly (or indirectly) possess a required role are able to view and execute a specific work item.

Issues

The range of resources that are authorised to undertake a task may not correspond to those to which it could be assigned based on the current resource pool within the workflow system.

Solutions

COSA provides a solution to this scenerio as follows:

- Where a resource is allocated a work item that is not authorised to execute, the work item will appear in its work list, but the resource cannot execute it. The resource can however reassign it to another resource that may be able to execute it
- Where a resource is authorised to undertake a given task, but the task is not able to be distributed to the resource (i.e. the distribution rules for the task preclude it from being allocated to the resource), work items corresponding to the task will never appear in the work list for the resource but it is able to execute them if they are directly allocated to it by other resources.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Use of execute roles allows authentication to be enforced
COSA	4	+	Distinct authorisation and distribution mechanisms are supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	This pattern is not supported by Oracle BPEL. It is possible to assign a work item to a specified role, and this work item can be reassigned to any other user/role. However, it is not possible to (re-)assign a task based on the condition, i.e. to a user having a certain authority

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. Workflow engine provides a distinct security framework defining which resources can execute which work items. 2. Authorisation to execute a work item is independent of the routing mechanism. 	<ol style="list-style-type: none"> 1. Authorisation is incorporated in the work distribution mechanisms. 2. Achievable via precondition specification or programmatic extensions for each task.

© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 5 (Separation of Duties)

[FLASH animation of Separation of Duties pattern](#)

Description

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

Example

Instances of the *Countersign cheque* task must be allocated to a different resource to that which executed the *Prepare cheque* task in a given workflow case.

Related Patterns

[R-CBA \(Capability-based Allocation\)](#), [R-RF \(Retain Familiar\)](#), [R-CH \(Case Handling\)](#)

Motivation

Separation of duties allows for the enforcement of audit controls within the execution of a workflow case. This ensures that a work item cannot be executed by the same resource that executed another work item within the same case. Another use of this Pattern arises with workflow engines that support multiple task instances. In this situation, the degree of parallelism that can be achieved when a multiple instance task is executed can be maximised by specifying that as far as possible no two task instances can be executed by the same resource.

Implementation

This Pattern can be implemented in a number of distinct ways:

- WebSphere MQ and FLOWer provide the ability to specify at task level, a link with another (preceding) task. At runtime, the work item corresponding to the task cannot be allocated to the same resource as that which undertook the previous instance of the work item corresponding to the linked task.
- iPlanet utilises the concepts of *linked activities* which allow the data elements of two distinct tasks to be shared and *evaluate methods* which define how the work items for a given task will be allocated to the various resources within the workflow system. For a given task, a custom *evaluate method* is constructed which ensures it cannot be allocated to the same resource that undertook the (preceding) instance of the task to which it was linked.
- COSA allows the effect of separation of duties to be achieved through the use of access rights which restrict the resource which undertook the preceding work item from executing the latter.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	+	Directly supported via task linking between activities in the process model that cannot have the same resource allocation at runtime within a case
FLOWer	3.0	+	Achieved by specifying additional execution constraints (not same user) on plan elements
COSA	4	+/-	Indirectly achievable through user access rights
iPlanet	3.1	+	Directly supported through linked activities
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL does not allow specifying the separation of duties in terms of relationships between tasks, nor it allows the separation of duties based on security mechanisms. Thus this pattern is not supported by Oracle BPEL PM.

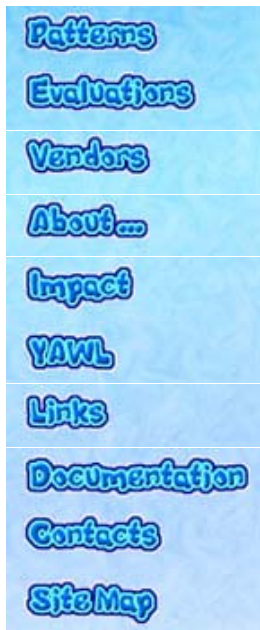
Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. Workflow engine supports the ability to prevent nominated work items within a case being allocated to the same resource as other specified work items in the same case. 2. The required separation of duties can be specified in the process model in terms of relationships between tasks. 	<ol style="list-style-type: none"> 1. Separation of duties is based on security mechanisms rather than the work distribution framework. 2. Programmatic extensions required to the work distribution process or individual task definitions.

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 6 (Case Handling)

[FLASH animation of Case Handling pattern](#)

Description

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

Example

All tasks in a given case of the *Prepare defence* process are allocated to the same *Legal Advisor*.

Related Patterns

[R-RF \(Retain Familiar\)](#)

Motivation

Case handling is a specific approach to work distribution that is based on the premise that all work items in a given case are so closely related that they should all be undertaken by the same resource. The identification of the specific resource occurs when a case (or the first work item in a case) requires allocation. Case handling may occur on either a "hard" or "soft" basis i.e. work items within a given case can be allocated exclusively to the same resource which must complete them all or alternatively it can serve as a guide to how work items within a given case should be routed with an initial resource being identified as having responsibility for all work items and subsequently delegating them to other resources or allowing them to nominate work items they would like to complete.

Implementation

This approach to work distribution is not generally supported by workflow systems. Of those examined only FLOWer (which describes itself as a case handling system) provides direct support.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Achieved by specifying that all plan elements should have the same user
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. The feature of dynamic assignment using an XPath expression allows specifying that a next task must be assigned to the resource who executed the previous (first) task. In particular, the function <code>ora:getPreviousTaskApprover()</code> can be used for this purposes

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. Workflow engine supports the ability to allocate all work items in a given case to the same resource. 2. Allocations are implied/specified at process level rather than for individual task definitions. 	1. N/A

© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 7 (Retain Familiar)

[FLASH animation of Retain Familiar pattern](#)

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.

Example

- If there are several suitable resources available to undertake the *Prepare Match Report* work item, it should be allocated to the same resource that undertook the *Umpire Match* task in a given workflow case.

Related Patterns

[R-CH \(Case Handling\)](#), [R-CE \(Chained Execution\)](#)

Motivation

Allocating a work item to the same resource that undertook a previous work item is a common means of expediting a workflow case. As the resource is already aware of the details of the case, it saves familiarisation time at the commencement of the work item. Where the two work items are sequential, it also offers the opportunity for minimising switching time as the resource can commence the latter work item immediately on completion of the former. This Pattern is a more flexible version of the Case Handling (R-CH) Pattern discussed earlier. It only comes into effect when there are multiple resources available to undertake a given work item and where this occurs, it favours the allocation of the work item to the resource that undertook a previous work item in the case. Unlike the Case Handling Pattern (which operates at case level), this Pattern applies at the work item level and comes into play when a work item is being allocated to a resource. The Chained Execution Pattern is a specialised form of this Pattern designed to expedite the completion of a case by automatically starting subsequent work items once the preceding work item is complete.

Implementation

Not surprisingly, this Pattern enjoys wider support than the Case Handling Pattern. WebSphere MQ allows individual work items to be allocated to the same resource that started another work item in a case or to the resource that started the case itself. FLOWer provides a facility in the design time workflow model to enforce that a task must be executed by the same resource as another specified task in the case. COSA does the same thing using a customised distribution algorithm for a specific work item that requires it to have the same executor as another work item in the case. Similarly

iPlanet achieves the same result using the linked user concept which requires two work items to be executed by the same resource.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	+	Common resource allocation can be specified for specific tasks in the process model requiring the same resource allocation at runtime within a case
FLOWer	3.0	+	Directly supported at plan element level
COSA	4	+	Supported via a customised distribution algorithm
iPlanet	3.1	+	Directly supported through linked tasks with common resources
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern by means of the ora:getPreviousTaskApprover() function during the dynamic assignment an assignee to a task

Summary of Evaluation

+ Rating	+/- Rating
1. The process model provides support for specifying that the resource to which a work item is allocated should be the same as that to which a preceding work item was allocated within a given case	1. N/A

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 8 (Capability-Based Allocation)

[FLASH animation of Capability-based Allocation pattern](#)

Description

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

Example

Instances of the *Airframe Examination* task should be allocated to an *Engineer* with an aeronautics degree, an Airbus in-service accreditation and more than 10 years experience in Airbus servicing.

Related Pattern

[R-RA \(Authorisation\)](#)

Motivation

Capability-based allocation provides a mechanism for offering or allocating work items to resources through the matching of specific requirements of work items with the capabilities of the potential range of resources that are available to undertake it. Capabilities are evaluated at run-time during work item offering or allocation. Depending on whether the work allocation strategy is push or pull-based, the actual allocation process can be initiated by the workflow system or the resource. In the former situation, the workflow system determined the most appropriate resource(s) to which a work item should be routed. In the latter, a resource initiates a search for an unallocated work item(s) which it is capable of undertaking.

Implementation

Capability-based allocation is based on the specification of capabilities for individual resources. Capabilities generally take the form of attribute - value pairs. A dictionary of capabilities can be defined in which individual capabilities have a distinct name and the type and potential range of values that each capability may take can also be specified. Classes of resources then indicate which capabilities are relevant to (and recorded for) individual resource instances. Similarly, tasks can also have capabilities recorded for them. The actual allocation process is often based on the specification of competency functions which are executed at runtime and determine how individual work items can be matched with suitable resources. These may be arbitrarily complex in nature depending on the range of capabilities that require matching between resources and work items and the approach that is taken to ranking the matches that are achieved in order to select the most appropriate resource to undertake a given work item. Both COSA and iPlanet implement capability-based allocation through the use of user-specified competency functions that form part of the process model. In both cases, the strategy is push-based. The example shown below in Figure 7 illustrates capability-based allocation with the capability function matching a work item to a resource on the basis of both resource capabilities and work item attributes. FLOWer uses case queries to determine which cases can be allocated to a specific resource. These can include data elements relating to both the case and the individual resource.

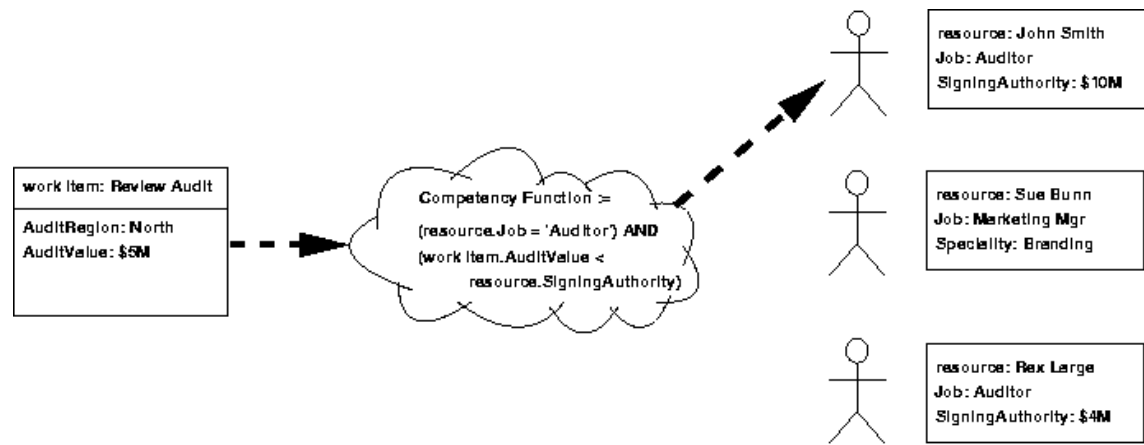


Figure 7. Capability Based Allocation

Issues

1. One consideration with push-based capability allocation is that it is possible for capability functions to identify more than one possible resource to which a work item may be assigned. Where this occurs, the work item may either be offered to multiple resources or assigned to one of the identified resources on a random basis. It is also possible for the capability function to return an empty set of possible resources.
2. In the case of pull-based capability allocation, it is possible for a resource to identify more than one work item that it is capable of undertaking.

Solutions

1. The first of these issues is not necessarily a problem although it may result in sub-optimal resource allocations. It can be avoided through more precise definition of capability functions. As an example, if the intention of the competence function in Figure 7 was to allocate the task to a single auditor, then a ranking function (e.g. minimum) should be included in the competence function to ensure only a single resource is returned. The second issue can be avoided by testing whether the competence function returns an empty set and if so, assigning a default value for the resource. COSA provides the ifnull operator for this purpose. iPlanet allows its evaluate methods to be arbitrarily complex to cater for situations such as this.
2. This should not generally result in difficulties. Under a pull-based allocation strategy, resources should anticipate the possible return of multiple work items. In some systems, it is possible for a resource to query matching work items without committing to executing them.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

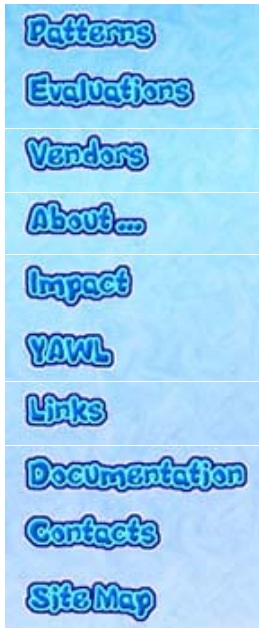
Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Cases can be assigned based on data or process elements

COSA	4	+	Supported via competency definitions for users, groups and activity definitions
iPlanet	3.1	+	Directly supported through customised distribution algorithms
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. It allows defining user properties and store them in user-properties.xml file, which become accessible via the function ora:getUserProperty(). This function can be used in the condition associated with the dynamic assignment feature.

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none">1. Workflow engine supports definition of resource capabilities in the process model.2. Resource requirements can be specified (in terms of capabilities) for individual task in the process model.3. Work distribution strategy takes capabilities into account when allocating work items.	<ol style="list-style-type: none">1. A similar effect is achieved indirectly through a combination of rules, role or group-based allocation and programmatic extensions.2. Allocation of a work item to an optimal resource (in terms of specified capabilities) cannot be guaranteed.

Workflow Patterns



Pattern 9 (History-Based Allocation)

[FLASH animation of History-based Allocation pattern](#)

Description

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

Example

- Allocate the *Finalise heart bypass* task to the *Surgeon* who has successfully completed the most of these tasks.
- Allocate the *Core extraction* task to the *drill* that has the lowest utilisation over the past 3 months.

Related Pattern

[R-RF \(Retain Familiar\)](#)

Motivation

History-based allocation involves the use of information on the previous execution history of resources when determining which of them a work item should be offered or allocated to. This is an analogue to common human experience when determining who to distribute a specific work to which considers factors such as who has the most experience with this type of work item or who has had the least numbers of failures when tackling similar tasks.

Implementation

None of the workflow engines examined provide direct support for history-based allocation, however for some of them it is possible to achieve some of the benefits of this approach by extending specific workflow models. There are essentially two methods of facilitating this:

- Extend the details maintained by individual resources on their work history and utilise this information when allocating work items.
- Extract details of work performance from the workflow log and incorporate this into the allocation process.

COSA provides facilities for the second method via customised distribution functions utilising the services of an external program to mine the workflow log. iPlanet is able to support both options using extended user profiles, modified task definitions to update user histories and customised distribution functions.

Issues

The main difficulty with facilitating this allocation strategy is that it places an additional processing overhead on workflow execution in order to maintain user execution details in a format that can be used when distributing work items.

Solutions

There is no immediate solution to this issue. Maintaining user execution profiles in a useful form requires additional processing to gather the required information and additional storage to maintain it. Where the allocation strategy is not directly supported by the workflow engine, modifications are required to the workflow process in order to achieve this. The only recommendation that can be made in this situation is to gather and manage the least amount of execution history for each resource that is required to facilitate the chosen work distribution strategy.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+/-	Indirectly achievable via a custom (external) distribution algorithm
iPlanet	3.1	+	Achievable through extended user profiles and customised distribution algorithms
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+/-	Oracle BPEL PM does not offer the direct support for this pattern, but it allows implementing this feature and accessing it via the properties of the dynamic assignment

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. The workflow engine provide facilities for utilising previous execution history when distributing work items. 2. The required allocation criteria can be specified in the process model. 	<ol style="list-style-type: none"> 1. The use of previous execution history requires (external) programmatic extensions to the work item allocation algorithm.

© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 10 (Organisational Allocation)

[FLASH animation of Organisational Allocation pattern](#)

Description

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

Example

- The *Review Audit* work item must be allocated to a *Partner* resource.
- The Authorise Expenditure work item must be allocated to the *Manager* of the resource that undertook the *Claim Expenditure* work item in a given case.

Related Patterns

[R-RBA \(Role-based Allocation\)](#), [R-RA \(Authorisation\)](#)

Motivation

Most workflow systems provide some degree of support for modelling the organisational context in which a given process operates. This is an important aspect of process modelling and implementation as many work allocation decisions are made in the context of the organisational structure and the relative position of individual resources both in the overall hierarchy and also in terms of their relationships with other resources. The ability to capture and emulate these types of work allocation strategies are an important requirement if workflow systems are to provide a flexible and realistic basis for managing work in an organisational setting.

Implementation

The degree of support for this Pattern varies widely. Staffware does not incorporate an organisational model and only provides support for role and group based work allocation. iPlanet is similar and only provides for role-based allocation, however it lacks any form of integrated organisational model. FLOWer extends the notion of role-based allocation and provides limited support for organisational structure in the form of a role hierarchy. WebSphere MQ supports a hierarchical organisational model and in addition to direct and role-based allocation, it allows organisational relationships such as coordinator of role, member of organisational unit, manager of organisation and starter of activity to be for work item allocation. COSA also incorporates a hierarchical organisational model and supports work allocation based either on roles or characteristics of the organisational model (e.g. supervisor, group membership).

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+/-	Partial support for roles and groups
Websphere MQ Workflow	3.4	+	Directly supported
FLOWer	3.0	+/-	Role hierarchy provides limited support for specifying an organisational structure for use in work distribution
COSA	4	+	Directly supported through user/group and user/group script languages
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+/-	Oracle BPEL PM offers an indirect support for this pattern. The organisational structure is stored in the xml format in the jasn-data.xml file, and it can be modified and extended. The relationships between roles are specified via the role-hierarchy tree. The roles defined become accessible via the look-up wizard.

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. The workflow engine provides support for specifying the relationships between workflow users (i.e. resources) in the form of an organisational model. 2. These relationships can be used as the basis of work item offering and allocation. 	<ol style="list-style-type: none"> 1. A similar effect can be achieved through programmatic extensions. 2. There is partial support for an organisational model.

© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 11 (Automatic Execution)

[FLASH animation of Automatic Execution pattern](#)

Description

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

Example

The *End of Day* work item executes without needing to be allocated to a resource.

Related Pattern

None

Motivation

Not all tasks within a workflow need to be executed under the auspices of an actual human resource, some are able to execute independently once the specified enabling criteria are met.

Implementation

Staffware, FLOWer, COSA and iPlanet all provide facilities for defining tasks which can run automatically within the context of the workflow without requiring allocation to a resource.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product	Version	Score	Motivation
Staffware	9	+	Directly supported via automatic steps
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Facilitated through the use of automatic actions
COSA	4	+	Achievable by assigning work items to internal system users
iPlanet	3.1	+	Directly supported
BPMN	1.0	+	A partitioning of a process into Pools and Lanes is not required, i.e., any resource allocation of a task is not necessarily done during design time.
UML	2.0	+	Directly supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM allows specifying tasks which involving the user, but also the tasks which are to be performed automatically. Any of the basic or structured activities offered by Oracle BPEL PM in the BPEL palette are executed automatically. Thus this pattern is directly supported.

Summary of Evaluation

+ Rating	+/- Rating
1. Provision exists within the process model to nominate tasks that can execute without requiring their corresponding work items to be allocated to a resource.	1. Work arounds exist to execute a task automatically by allocating it to an internal resource or by executing a linked external program.

© 2007 *Workflow Patterns Initiative*

0025927

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 12 (Distribution by Offer - Single Resource)

[FLASH animation of Distribution by Offer - Single Resource pattern](#)

Description

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

Example

The *Prepare defense* work item is offered to a selected *Barrister*.

Related Patterns

[R-DA \(Direct Allocation\)](#), [R-DBOM \(Distribution by Offer - Multiple Resources\)](#), [R-DBAS \(Distribution by Allocation - Single Resource\)](#)

Motivation

Once a work item has been created and it has been determined that the work item should be offered to a single distinct resource for potential execution, a means of actually informing the resource of the pending work item is required. The mechanism should notify the resource that a work item exists that it may wish to undertake, however it should not commit the resource to its execution and it should not advise any other resources of the potential work item. This Pattern directly corresponds to the state transition denoted by arc S:offer_s in Figure [8](#).

Implementation

Of the workflow engines examined, only iPlanet directly supports the ability to offer a work item to a single resource without the resource being committed to executing the work item. COSA provides a close analogy to this concept in that it allows a resource to reject a work item that has been allocated to it and placed on its work queue. When this occurs, the work item goes through a subsequent reallocation process, ultimately resulting in it being assigned to a different resource.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+/-	Non-binding offers not supported but allocated work items can be rejected for reallocation
iPlanet	3.1	+	Directly supported for offered work items
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern by offering work item to members of a group. A group containing one user allows this user to "acquire" the offered work item

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine provides a mechanism to advertise a work item to a specific resource. The resource is not obliged to commit to executing the work item.	1. The workflow engine indirectly supports the concept by allowing resources to reject work items assigned to them.

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 13 (Distribution by Offer - Multiple Resources)

[FLASH animation of Distribution by Offer - Multiple Resources pattern](#)

Description

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

Example

The *Sell portfolio* work item is offered to multiple *Stockbrokers*.

Related Patterns

[R-RBA \(Role-based Allocation\)](#), [R-CBA \(Capability-based Allocation\)](#), [R-DBAS \(Distribution by Allocation - Single Resource\)](#), [R-DBAM \(Distribution by Allocation - Multiple Resources\)](#)

Motivation

Offering a work item to multiple resources is the workflow analogy to the act of "calling for a volunteer" in real life. It provides a means of advising a suitably qualified group of resources that a work item exists but leaves the onus with them as to who actually commits to undertaking the activity. This Pattern directly corresponds to the state transition denoted by arc S:offer_m in Figure [8](#).

Implementation

Several workflow engines support the notion of work groups and allow work items to be allocated to them. A work group is a group of resources with a common organisational focus. When a work item is allocated to the group, each of the members of the group is advised of its existence, but until one of them commits to starting it and advises the workflow engine of this fact, it remains on the work queue for each of the resources. There are several possibilities for resources being advised of group work items - they may appear on each of the individual resource's work queues, each resource may have a distinct work queue for group items on which they may appear or all resources in a work group may have the ability to view a shared group work queue in addition to their own dedicated work queue (footnote [2](#)). Different workflow engines handle the offering of a work item to multiple resources in different ways:

- WebSphere MQ treats work items offered to multiple resources in the same way as work items allocated to a specific resource and they appear on the work list of resources to whom they are offered. When a multiply-offered work item is accepted by one of the resources to which it is offered, it is removed from the work lists of all other resources.

- Staffware and COSA support the concept of distinct user specific work queues and group work queues. Where a multiply-offered work item is accepted by a resource, it remains on the group work list but is not able to be selected for execution by other resources.
- iPlanet supports distinct work queues for offered and queued (i.e. allocated) work items. Once a multiply-offered work item has been accepted by a resource, it is removed from all offered work queues and only appears on the queued list for the resource which has accepted it.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported for group queues
Websphere MQ Workflow	3.4	+	Work queues combine work item items specifically offered to this resource and those offered to multiple resources
FLOWer	3.0	+	Default work allocation mechanism is to offer a work item to all participants in a role
COSA	4	+	Directly supported for multiple resources via group queues
iPlanet	3.1	+	Directly supported for offered and queued work items
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly by specifying the name of a group as an assignee of the task. As a result, the task will be offered to all members of a group, and any of the members may acquire it. After the work item has been acquired, not other users may acquire this work item any more.

Summary of Evaluation

+ Rating	+/- Rating
-----------------	-------------------

<ol style="list-style-type: none">1. The workflow engine provides a mechanism to advertise a work item to several resources simultaneously2. None of the resources are obliged to commit to executing the work item.	1. N/A
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 14 (Distribution by Allocation - Single Resource)

[FLASH animation of Distribution by Allocation - Single Resource pattern](#)

Description

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

Example

The *Cover Comalco AGM* work item should be allocated to the *Finance Sub-editor*.

Related Patterns

[R-DA \(Direct Allocation\)](#), [R-DBOS \(Distribution by Offer - Single Resource\)](#),
[R-DBOM \(Distribution by Offer - Multiple Resources\)](#)

Motivation

Distribution by allocation to a single resource corresponds to the notion of the workflow engine directly assigning a work item to a resource without first offering it to other resources or querying whether the resource will undertake it. This approach to work distribution is also known as "heads down" processing as it offers the resource little or no input in the work that they are allocated and the main focus is on maximising work throughput by keeping the resource busy. In many implementations, resources are simply allocated a new work item once the old one is completed and they are not offered any insight into what work items might lay ahead for them. This Pattern directly corresponds to the state transition denoted by arc S:allocate in Figure [8](#).

Implementation

Where a specific resource has been identified during the course of work item distribution, this is the standard means of allocating a work item to a resource. It is done pre-emptively by the workflow engine and necessitates that the resource actually execute the work item unless it has recourse to a means of rejecting it. All the tools evaluated support direct allocation of work items to resources.

Issues

None observed.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported where a single resource is identified during resource selection for a work item
Websphere MQ Workflow	3.4	+	Directly supported for work items allocated to a single resource
FLOWer	3.0	+	Directly supported where a role corresponds to a specific resource
COSA	4	+	Directly supported
iPlanet	3.1	+	Directly supported where a work item is allocated to a single resource in "heads down" mode
BPMN	1.0	+	We assume that the resource associated with a Swimlane is immediately allocated a Task/Sub-Processes once it is triggered
UML	2.0	+	Resource allocation is immediate once an action is triggered.
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. Assigning a user with a given identity statically or dynamically, automatically allocated the work item to this user. This differs from the pattern RP12 where the work item is offered to the user.

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine provides a means of directly allocating a work item to a single resource.	1. N/A

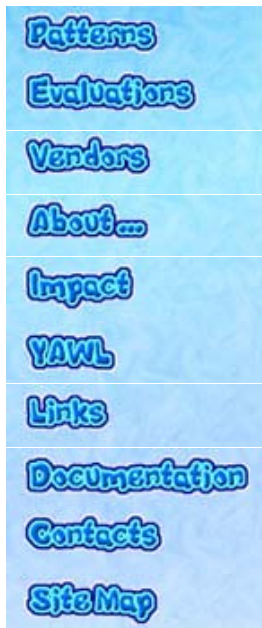
© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 15 (Random Allocation)

[FLASH animation of Random Allocation pattern](#)

Description

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

Example

The *Judge case* work item is allocated to a *Magistrate* on a random basis.

Related Patterns

None.

Motivation

Random allocation provides a non-deterministic mechanism for allocating work items to resources in workflow systems. Once the possible range of resources that a work item can be allocated to have been identified at runtime, one of these is selected at random to execute the work item.

Implementation

Of the systems examined, only COSA provides direct support for work allocation on a random basis using the random operator which forms part of the user/group language. This is a scripting language which allows relatively complex work allocation rules to be specified. Similarly, iPlanet allows the work distribution algorithm to be extended programmatically although there is no direct support for random allocation within the product.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.

- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+	Directly supported via random function in user/group language
iPlanet	3.1	+/-	Indirectly achievable via customised (external) distribution algorithm
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+/-	Oracle BPEL PM offers no direct support for this pattern. However, the feature of assigning a work item dynamically can be used to support this pattern. For example, a service can be implemented to retrieve resource on the random basis

Summary of Evaluation

+ Rating	+/- Rating
1. Facilities exist (either in the process model or at run-time) to nominate that the resource to which a work item is allocated is picked at random from those that satisfy the selection criteria for potential assignment.	1. N/A

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
 (webmaster@workflowpatterns.com)

Workflow Patterns

[Patterns](#)[Evaluations](#)[Vendors](#)[About...](#)[Impact](#)[YAWL](#)[Links](#)[Documentation](#)[Contacts](#)[Site Map](#)

Pattern 16 (Round Robin Allocation)

[FLASH animation of Round Robin Allocation pattern](#)

Description

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

Example

Work items corresponding to the *Umpire Match* task are allocated to each available *Referee* on a cyclic basis.

Related Patterns

None.

Motivation

A round robin allocation strategy provides a means of ensuring that all resources are allocated the same number of work items.

Implementation

None of the workflow engines examined provide direct support for round robin allocation. However both COSA and iPlanet provide facilities for specifying custom allocation strategies for workflow tasks. In the case of COSA, a custom distribution algorithm can be specified (incorporating an external program) that implements round robin allocation. As the total available working time for each user can be specified (as a percentage between 0 and 100%), there is the opportunity to establish a relatively fair basis for round robin allocation. For iPlanet, it is possible to develop an Evaluate method that achieves a similar result.

Issues

By its nature, round robin allocation requires details of individual resource allocations to be maintained so that a decision can be made as to which resource should be used when the next allocation decision is made.

Solutions

Where a workflow engine does not directly support round robin allocation, it is left to the auspices of the workflow developer to implement a strategy for this form of allocation. For the systems described above, COSA relies on the use of an external program to manage the allocation decision and keep track of previous allocations. iPlanet utilises Evaluate methods based on the TOOL language and access to an

external SQL database for managing allocations.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+/-	Indirectly achievable via a custom (external) distribution algorithm
iPlanet	3.1	+/-	Indirectly achievable via customised (external) distribution algorithm
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+/-	Oracle BPEL PM offers no direct support for this pattern. However, the feature of assigning a work item dynamically can be used to support this pattern. For example, a service can be implemented to retrieve resource based on the round-robin algorithm

Summary of Evaluation

+ Rating	+/- Rating
1. Facilities exist (either in the process model or at run-time) to nominate that the resource to which a work item is allocated is chosen on a cyclic basis from the group of resources which satisfy the selection criteria for potential assignment.	1. N/A

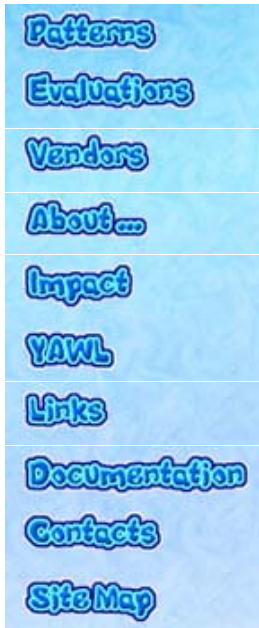
© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 17 (Shortest Queue)

[FLASH animation of Shortest Queue pattern](#)

Description

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

Example

The *Heart Bypass Procedure* is allocated to the *Surgeon* who has the least number of operations allocated to them.

Related Patterns

None

Motivation

This allocation mechanism seeks to expedite the throughput of a workflow process by ensuring that work items are allocated to the resource that is able to undertake them in the shortest possible timeframe. Typically the shortest timeframe means the resource with the shortest work list queue although other interpretations are possible.

Implementation

In order to implement this allocation method, workflow engines need to maintain information on the work items currently allocated to resources and make this information available to the work item distribution algorithm. Of the workflow engines examined, COSA provides the `fewwork()` function which allows this Pattern to be directly realised. iPlanet provide facilities for programmatically extending the work item distribution algorithm and enabling this to be achieved indirectly.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it

complies with each of the criteria specified.

- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+	Supported via a customised distribution algorithm based on the fewwork function
iPlanet	3.1	+/-	Indirectly achievable via customised (external) distribution algorithm
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+/-	Oracle BPEL PM offers no direct support for this pattern. However, the feature of assigning a work item dynamically can be used to support this pattern. For example, a service can be implemented to retrieve resource based on the shortest queue algorithm

Summary of Evaluation

+ Rating	+/- Rating
1. Facilities exist (either in the process model or at run-time) to nominate that a work item should be allocated to the resource with the shortest work queue from the group of resources that meets the selection criteria for potential assignment.	1. N/A

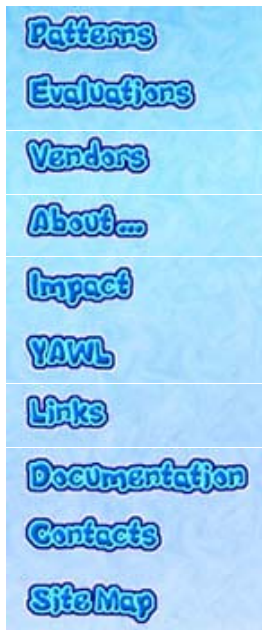
© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 18 (Early Distribution)

[FLASH animation of Early Distribution pattern](#)

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.

Example

The *Captain BA12 London - Bangkok flight* work item is offered to potential *Chief Pilots* at least two weeks ahead of the time that it will commence.

Related Patterns

[R-CA \(Commencement on Allocation\)](#)

Motivation

Early distribution provides a means of notifying resources of upcoming work items ahead of the time at which they need to be executed. This is useful where resources are able to provide some form of forward commitment (or booking) indicating they they will execute and complete a work item at some future time. It also provides a means of optimising the throughput of a workflow case by ensuring that minimal time is spent waiting for resource allocation during case execution.

Implementation

None of the workflow systems examined directly support this Pattern, suggesting that the focus of production workflow systems tends to be on the management and completion of current work rather than on planning the optimal execution strategy for future work items. FLOWer (a case handling system) provides the ability for a resource to view future work items and potentially commence work on them even though they are not the next items in the process sequence. The case handling paradigm offers a different approach to work allocation. It is not discussed in detail here and interested readers are referred to [\[AWG05\]](#) for further information.

Issues

None observed.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Resources are able to view and execute tasks ahead of the wavefront
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
1. Resources can commit to executing potential work items ahead of the time that the work items exist or become available for execution	1. N/A

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
[\(webmaster@workflowpatterns.com\)](mailto:webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 19 (Distribution on Enablement)

[FLASH animation of Distribution on Enablement pattern](#)

Description

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

Example

The *Delivery Round* work item is allocated to a *Paper boy* at the time it is required to commence.

Related Patterns

[R-CC \(Commencement on Creation\)](#), [R-CE \(Chained Execution\)](#)

Motivation

Distribution of work items at the time that they are enabled for execution is effectively the standard mechanism for work distribution in a workflow system. The enablement of a work item serves as the trigger for the workflow engine to make it available to resources for execution. This may occur indirectly by placing it on the worklists for individual resources or on the global work list or directly by allocating it to a specific resource for immediate execution.

Implementation

All of the systems examined directly support this approach to work distribution in some form.

Issues

None observed.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it

complies with each of the criteria specified.

- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported as standard means of work item distribution
Websphere MQ Workflow	3.4	+	Standard mechanism for work item distribution
FLOWer	3.0	+	In general, resources are allocated and execute work items on the wave front (i.e. one that have been enabled)
COSA	4	+	Standard means of work distribution
iPlanet	3.1	+	Standard means of work item distribution
BPMN	1.0	+	All activities in a Swimlane are associated with the resource responsible for the Swimlane when they are triggered
UML	2.0	+	Directly supported. All actions in a partition are associated with the resource responsible for the partition when they are triggered
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly: as soon as a work item becomes available, it appears in the work list of the assigned resource

Summary of Evaluation

+ Rating	+/- Rating
1. Work items become available for advertising/allocation to resources at the time that they are enabled	1. N/A

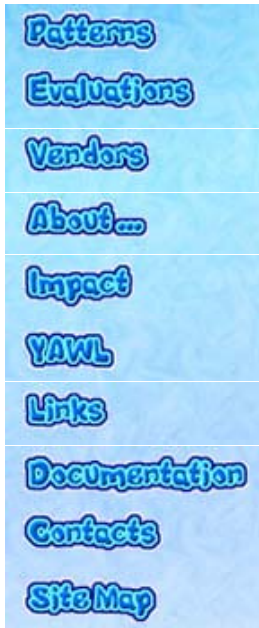
© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 20 (Late Distribution)

[FLASH animation of Late Distribution pattern](#)

Description

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

Example

The *Service Car* work item is allocated to a *Mechanic* after the car has been delivered for repair.

Related Patterns

None

Motivation

Late distribution of work items effectively provides a means of demand driving a workflow process by only advertising or allocating work items to resources when the work item has already been enabled for execution, possibly at some previous time. By adopting this approach, it is possible to reduce the current volume of work in progress within a workflow. This may involve stopping work items (or cases) from executing once the workload exceeds a certain threshold or restricting the amount of work in specific segments of the workflow. Often this strategy is undertaken with the aim of preventing resources from becoming overwhelmed by the apparent workload even though they may not be required to undertake all of it themselves.

Implementation

None of the workflow engines examined support the notion of late distribution for newly created work items. However, a similar notion is used by some workflow engines for redeploying work items that have been allocated to resources or possibly have even commenced execution. COSA supports manual rerouting of work items by workflow users. WebSphere MQ provides an API for rerouting of work items.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM does not support this pattern since by any work item requires to have an assignee. Since the work item has to be allocated or offered to a user/role from the moment of creation, the late distribution is not possible

Summary of Evaluation

+ Rating	+/- Rating
1. Work items can be advertised/allocated to resources after the time at which they are enabled for execution (i.e. general knowledge of their existence can be delayed).	1. N/A

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 21 (Resource-Initiated Allocation)

[FLASH animation of Resource-Initiated Allocation pattern](#)

Description

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

Example

The *Clerk* selects the Town Planning work items that she will undertake today although she only commence working on one of these at this point.

Related Patterns

None

Motivation

This Pattern provides a means for a resource to signal its intention to execute a given work item at some point although it may not commence working on it immediately. As a consequence of this action, the work item is considered to be allocated to the resource and it cannot be allocated to or executed by another resource. There are two variants of this Pattern as illustrated by the bold arcs in Figure 9, depending on whether the work item has been offered to a single resource (R:allocate_s) or to multiple resources (R:allocate_m).

Implementation

The implementation of this Pattern generally involves the removal of the work item from a globally accessible or shared work list and its placement on a work queue specific to the resource to which it is allocated. Surprisingly only two of the workflow engines examined supports this function. COSA allows a resource to reserve a work item that is displayed on a shared or global worklist for later execution by a user, however in doing so, the entire process instance is locked by the resource until the work item is completed or the reserve timeout is reached. In FLOWer, cases are retrieved for a given resource via a case query which specifies the distribution criteria for cases that can be allocated to the resource. Where a resource executes a case query and a matching case is identified, all of the work items in the case are effectively allocated to the resource. Each of these work items is listed in the resource's work tray but is not commenced until specifically requested by the resource.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Directly supported
COSA	4	+/-	The act of a resource reserving a work item on a shared work list has the effect of locking the process instance
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM does not support this pattern directly. If a work item is assigned to a set of users or a group, one of the users in the list can "acquire" the task. However, this corresponds to the commence working on it immediately

Summary of Evaluation

+ Rating	+/- Rating
1. Resources are able to commit to executing an offered work item, resulting in it being allocated to their work queue. They are not obliged to commence working on it immediately.	1. Resources can reserve a work item for later execution by suspending execution of the workflow case.

© 2007 *Workflow Patterns Initiative*

0025928

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 22 (Resource-Initiated Execution - Allocated Work Item)

[FLASH animation of Resource-Initiated Execution - Allocated Work Item pattern](#)

Description

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

Example

The *Courier Driver* selects the next *Delivery* work item which is allocated to it and commences work on it.

Related Patterns

[R-CC \(Commencement on Allocation\)](#)

Motivation

Where a resource has work items that it has committed to execute, but has not yet commenced, a means of signalling their commencement is required. This Pattern fulfils that requirement. It corresponds to the R:start transition illustrated in Figure [9](#).

Implementation

The general means of handling that a work item has been allocated to a resource is to place it on a resource-specific work queue. This ensures that the work item is not undertaken by another resource and that the commitment made by the resource to which it is allocated is maintained. Staffware, WebSphere MQ, FLOWer and COSA all support the concept of resource-specific work queues and provide mechanisms in the work list handlers for resources to indicate that an allocated work item has been commenced.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported as a standard consequence of starting an item on a work queue
Websphere MQ Workflow	3.4	+	Standard means for a resource to initiate a work item is to select one from those allocated to it
FLOWer	3.0	+	Directly supported
COSA	4	+	Resources can initiate allocated work items from their work queues
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	In Oracle BPEL PM resources are able to commence the execution of a work item available at their own worklists at a time of their own choosing, but before the task has expired. Therefore this pattern is supported directly.

Summary of Evaluation

+ Rating	+/- Rating
1. Resources are able to commence execution of a work item allocated to them at a time of their own choosing.	1. N/A

© 2007 *Workflow Patterns Initiative*

0025929

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 23 (Resource-Initiated Execution - Offered Work Item)

[FLASH animation of Resource-Initiated Execution - Offered Work Item pattern](#)

Description

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

Example

The *Courier Driver* selects the next *Delivery* work item from those offered and commences work on it.

Related Patterns

None

Motivation

In some cases it is preferable to view a resource as being committed to undertaking a work item only when the resource has actually indicated that it is working on it. This approach to work distribution effectively speeds throughput by eliminating the notion of work item allocation. Work items remain on offer to the widest range of appropriate resources until one of them actually indicates they can commence work on it. Only at this time is the work item removed from being on offer and allocated to a specific resource. This Pattern corresponds to the R:start_s and R:start_m transitions shown in Figure [9](#).

Implementation

This approach to work distribution is adopted by Staffware, WebSphere MQ and COSA for shared work queues (e.g. group queues). For these systems, a work item remains on the queue until a resource indicates that it has commenced it. At this point, its status changes and no other resource can execute it although it remains on the shared queue until it is completed. iPlanet adopts this approach for all work items and effectively presents each resource with a single amalgamated queue of work items allocated directly to it and also those offered to a range of resources. The resource must indicate when it wishes to commence a work item. This results in the status of the work item changing and it being removed from any other work queues on which it might have existed.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported for group queues
Websphere MQ Workflow	3.4	+	Supported for work items distributed via shared work queues
FLOWer	3.0	-	Not supported
COSA	4	+	Supported for work items on shared work queues
iPlanet	3.1	+	Standard approach to initiating work items
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. When a user is offered a work item, and the user "acquires" it, he commences work on it immediately

Summary of Evaluation

+ Rating	+/- Rating
1. Resources are able to simultaneously request allocation and commence execution on work item(s) offered to them.	1. N/A

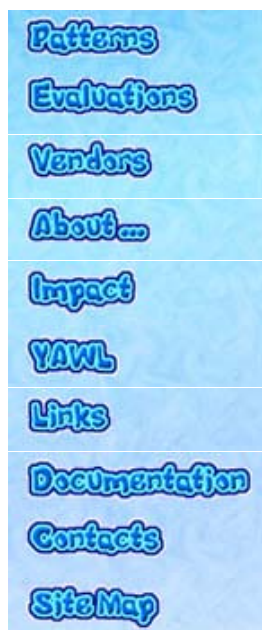
© 2007 *Workflow Patterns Initiative*

0025929

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 24 (System-Determined Work Queue Content)

[FLASH animation of System-Determined Work Queue Content Animation](http://www.workflowpatterns.com/patterns/resource/pull/wrp24.php)

Description

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

Example

The *Staffware* workflow engine presents work items to resources in order of work item priority.

Related Patterns

None

Motivation

Where a workflow engine provide facilities for specifying the default ordering in which work items are presented to resources, the opportunity exists to enforce a work ordering policy for all workflow resources or on a group-by-group or individual resource basis. Such ordering may be time-based (e.g. FIFO, LIFO, EDD) or relate to data values associated with individual work items (e.g. cost, required effort, completion time).

Implementation

Where this concept is supported by individual workflow engines, it is generally done so in terms of a single ordering sequence for all resources. Both Staffware and iPlanet support the ordering of work items on a priority basis for each resource's worklist. In both cases they also support the dynamic reordering of worklists as the priorities of individual work items change.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported with work items ordered by priority
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	The system presents work items in order of execution sequence
COSA	4	-	Not supported
iPlanet	3.1	+	Work items are ordered by priority by default
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM does not impose a default ordering of the work items in the resources work queue, thus offering no support for this pattern.

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine is able to impose a default ordering of the work items in a resource's work queue	1. N/A

© 2007 *Workflow Patterns Initiative*

0025929

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 25 (Resource-Determined Work Queue Content)

[FLASH animation of Resource-Determined Work Queue Content pattern](#)

Description

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

Example

The *Coordinator* resource has a work list ordered by time of receipt.

Motivation

Enabling resources to specify the format, content and ordering of their work queue provides them with a greater degree of flexibility in how they go about tackling the work items to which they have committed.

Implementation

For those workflow engines which provide a client application for resources to interact with the workflow engine, the ability to be able to sort and filter work items is relatively commonplace. Staffware and WebSphere MQ allow any work item attribute to be used as the basis of the sort criterion or for filtering the work items that are displayed. FLOWer goes a step further and allows the user to specify "case queries" which define the type of cases that are retrieved into their work tray. COSA allows multiple views of available work to be defined and used at the resource level and includes support for the filtering of work items and specification of worklist queries.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.

- Otherwise a - rating is recorded.

Product	Version	Score	Motivation
Staffware	9	+	Directly supported through resource-specific work queue customisation and filtering
Websphere MQ Workflow	3.4	+	Work queues can be sorted or filtered on any work item attribute at the discretion of individual resources
FLOWer	3.0	+	Distinct work trays can be configured for users
COSA	4	+	A series of options are provided for configuring views in the worklist handler
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. The user is able to format the work items listed in his worklist based on the task id, priority and other parameters. In addition, Oracle ships a JSP based sample worklist application that can be customized to list specific content on the worklist application

Summary of Evaluation

+ Rating	+/- Rating
1. Each resource is able to impose a filtering and/or an ordering sequence on the work items appearing in its work queue.	1. The extent of the ordering sequence is limited 2. Filtering of work items is supported by not reordering.

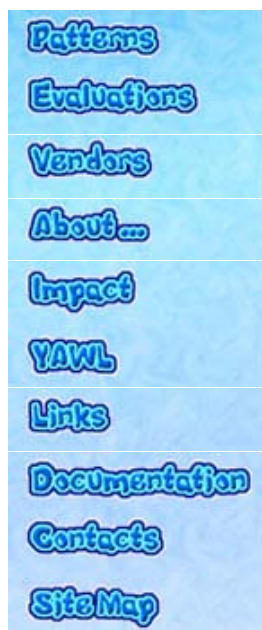
© 2007 *Workflow Patterns Initiative*

0025929

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 26 (Selection Autonomy)

[FLASH animation of Selection Autonomy pattern](#)

Description

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

Example

Of the outstanding Pruning work items, the *Head Gardener* chooses the one for execution they feel they are best suited to.

Related Patterns

None

Motivation

The ability for a resource to select the work item that they will commence next is a key aspect of the "heads up" approach to workflow execution. It aims to empower resources and let them have the flexibility to prioritise and organise their own individual work sequence.

Implementation

All of the workflow engines examined provide support for this Pattern.

Issues

One consideration with this Pattern is whether resources are still offered complete flexibility to choose which work item they will undertake next when there are urgent work items allocated to them or whether the workflow engine can guide their choice or dictate that a specific work item will be undertaken next.

Solutions

Where autonomy is offered to resources in terms of the work items that they choose to execute, it is typically not revoked even in the face of pressing work items. Staffware and WebSphere MQ provide a means of highlighting urgent work items but do not mandate that these should be executed. Other workflow engines examined do not provide any facilities in this regard.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported as the standard means for a resource to select the next work item
Websphere MQ Workflow	3.4	+	Resources can select the next item for execution from those on their work queue
FLOWer	3.0	+	Users can execute any work item at, ahead or behind the wavefront
COSA	4	+	Resources can select any of the work items on their queue to initiate next
iPlanet	3.1	+	Directly supported for offered work items
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. The user can select and act on any of the task displayed in his work list.

Summary of Evaluation

+ Rating	+/- Rating
1. A resource is able to select the next work item that it will commence from any of those currently in its work queue	1. N/A

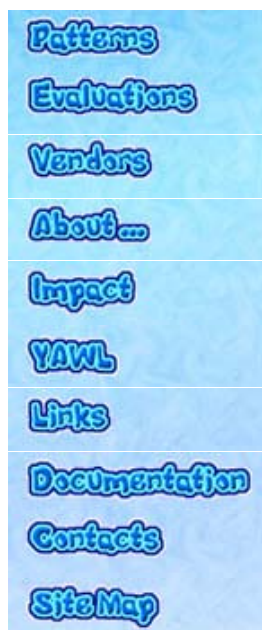
© 2007 *Workflow Patterns Initiative*

0025930

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 27 (Delegation)

[FLASH animation of Delegation pattern](#)

Description

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

Example

Before going on leave, the *Chief Accountant* passed all of their outstanding work items onto the *Assistant Accountant*.

Related Patterns

None

Motivation

Delegation provides a resource with a means of re-routing work items that it is unable to execute. This may be because the resource is unavailable (e.g. on vacation) or because they do not wish to take on any more work. It is illustrated by the R:delegate transition in Figure [10](#).

Implementation

Generally the ability to delegate work items is included in the client work list handler for a workflow engine. Staffware, WebSphere MQ and COSA all provide the ability to manually redirect queued work items to a nominated resource. COSA supports an enhanced notion of delegation in that it redirects all work items corresponding to a specific task definition to a specified resource.

Issues

One consideration associated with delegation is what happens where a work item is delegated to a user who is not authorised to execute it.

Solutions

This scenario is only a problem for workflow engines that support distinct task routing and authorisation mechanisms. Both Staffware and WebSphere MQ allow a resource to execute any work item that is routed to them. However COSA provides an authorisation framework for work items that operates alongside the distribution mechanism. In COSA, a work item could be distributed to a resource that does not have authorisation rights for it. Where this occurs, the resource can view the work

item in their work list but cannot execute it. The only resolution is for them to delegate the work item to another resource that does have the required authorisation rights, or else acquire those rights themselves.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported through task forwarding
Websphere MQ Workflow	3.4	+	Work items can be manually redirected by resources
FLOWer	3.0	-	Not supported
COSA	4	+	Resources can manually reroute work items from their work queues
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly by means of "reassign" action. As such, a manager can delegate a task to reportees. Similarly, the process owner or a user with BPMWorkflowReassign privileges can delegate a specific task to any other person in the organisation.

Summary of Evaluation

+ Rating	+/- Rating
1. Resources can redirect a work item allocated to them (but not initiated) to another resource either manually or automatically.	1. N/A

© 2007 *Workflow Patterns Initiative*

0025930

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
[\(webmaster@workflowpatterns.com\)](mailto:webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 28 (Escalation)

[FLASH animation of Escalation pattern](#)

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.

Example

- The *review earnings* work item was reallocated to the *CFO*. It had previously been allocated to the *Financial Accountant* but the deadline for completion had been exceeded.

Related Patterns

None

Motivation

Escalation provides the ability for the workflow system to intervene in the conduct of a work item and assign it to alternative resources. Generally this occurs as a result of a specified deadline being exceeded, but it may also be a consequence of pre-emptive load balancing of work allocations undertaken by the workflow system or administrator in an attempt to optimise workflow throughput. There are various ways in which a work item may be escalated depending on its current state of progression and the approach that is taken to identifying a suitable party to which it should be reassigned. The possible range of alternatives are illustrated by the R:escalate_sm, R:escalate_am, R:escalate_mm, R:escalate_ss, R:escalate_as and R:escalate_aa transitions in Figure [10](#).

Implementation

Staffware, COSA and iPlanet provide direct support for deadlines on work items and allow alternate work items to be triggered (with distinct routing options) in the event that a work item fails to be completed in the required timeframe. WebSphere MQ provides reminders that notify a nominated resource that a given work item has exceeded a specified deadline.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported through withdraw actions and re-allocating another instance of the same task
Websphere MQ Workflow	3.4	+	Directly supported via reminders
FLOWer	3.0	-	Not supported
COSA	4	+	Supported via the Target Dates option for processes together with a trigger to reroute the work item
iPlanet	3.1	+/-	Indirectly supported via timers but the cleanup action for the escalated work item must be specified programmatically
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly by allowing escalation of a task to the manager for further action. The escalation continues until a certain user, a certain level (number of escalations to a 'manager'), or a certain title is reached. The escalation feature works correctly if a task has been assigned to a specific user, however if a task has been assigned to a role or a group, Oracle BPEL PM does not seem to know an upper level where a task should be escalated to.

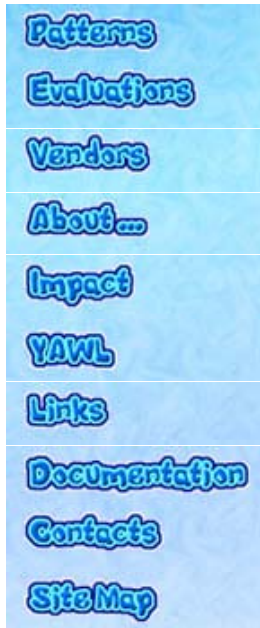
Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. The workflow system can redirect a work item to another resource (or group of resources) either manually or automatically. 2. Work items in any state (offered, allocated or commenced) can be redirected. 	<ol style="list-style-type: none"> 1. Only work items in certain states can be redirected.

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 29 (Deallocation)

[FLASH animation of Deallocation pattern](#)

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.

Example

As progress on the *Conduct initial investigation* work item is not sufficient, the *Level 1 support officer* resource has made it available for reallocation to another *Support Consultant*.

Related Patterns

[R-E \(Escalation\)](#)

Motivation

Deallocation provides resources with a means of relinquishing work items allocated to them and making them available for re-allocation to other resources. This may occur for a variety of reasons including insufficient progress, availability of a better resource or a general need to unload work from a resource. There are two possible variations to deallocation - either the work item can be offered to a single resource or to multiple resources. These transitions are illustrated by the R:deallocate_s and R:deallocate_m arcs in Figure [10](#).

Implementation

Despite the potential that this Pattern offers for actively managing the workload across a process, it is not widely implemented. COSA supports this Pattern through the redistribution function. iPlanet provides the ability for the workflow engine to reset the status of an active work item to ready. This has the effect of causing the work item to be reallocated using the same set of distribution criteria as were previously utilised for the work item.

Issues

One problem that can arise when deallocating a work item is that it could ultimately be re-allocated to the same resource that it was previously retrieved from.

Solutions

As the act of deallocating a work item is generally disjoint from that of reallocating it, the potential always exists for reallocation to the same resource unless active measures are taken to ensure that this does not occur. Generally there are three approaches for doing this:

- Make the resource unavailable for the period in which the reallocation will occur so that it is not considered in the work item redistribution.
- Stop the resource accepting new allocations or offers.
- Ensure that the distribution algorithm does not attempt to allocate a work item to a resource to which it has previously been allocated.

For iPlanet, the second and third options are both possible solutions where the workflow is running in "heads up" mode and resources have work items offered to them. Where it is running "heads down" and resources are directly allocated the next work item without an offer occurring, only the third option is feasible. In COSA, there is no direct solution to this problem.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+	Directly supported through redistribution
iPlanet	3.1	+	Achieved by changing the status of a work item to READY
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. If a work item has been assigned to a set of users of a group, one of the users in the list can "acquire" the task. At anytime before the task expires or before a user has updated the task, the user can "release" the task to the set of users/group the task was originally assigned to

Summary of Evaluation

+ Rating	+/- Rating
1. A resource is able to remove (allocated or commenced) items from its work queue and make them available for reallocation to other resources.	1. Only allocated work items can be redirected.

© 2007 *Workflow Patterns Initiative*

0025930

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 30 (Stateful Reallocation)

[FLASH animation of Stateful Reallocation pattern](#)

Description

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

Example

- The *Senior Partner* has suspended work on the *Building Society Audit Plan* work item and passed it to the *Junior Project Manager* for further work.

Related Patterns

[R-UR \(Stateless Reallocation\)](#), [R-D \(Delegation\)](#)

Motivation

Planned reallocation provides a resource with the ability to offload both pending and currently executing work items to other resources whilst maintaining the current state of the work item and the results of work undertaken on it to date. In the main, this centres on the ability to retain the current values of all data elements associated with the work item. This Pattern corresponds to the R:reallocation_with_state arc in Figure [10](#). It is interesting to note the similarities between this Pattern and the Delegation Pattern. Both Patterns result in a work item being reassigned to another resource. The main difference is that Delegation can only occur for a work item that has not yet commenced execution where as this Pattern applies to work items that are currently being executed.

Implementation

Staffware, WebSphere MQ and COSA all support the notion of reallocating a work item to another resource with preservation of state albeit in slightly differing ways. Staffware only allows pending work items to be reallocated. WebSphere MQ requires that the work item is either pending or suspended in order for it to be reallocated. COSA provides support for reallocation through the reroute function.

Issues

There are two potential issues associated with the reallocation of a work item to another resource whilst still preserving state information:

- Managing the transfer of state data.
- Ensuring the resource to which the work item is reallocated is entitled to execute it and access the associated state information.

Solutions

There are a number of potential solutions to the first of these issues. One of these is to limit access to relevant state data elements to the resource executing the work item. This is the approach adopted by WebSphere MQ and COSA which use data containers to manage the data elements being passed between work items and work item specific data elements to manage state respectively. Staffware neatly avoids this issue by only allowing work items that have not been started to be reallocated. The second of these issues is potentially more problematic. Staffware and WebSphere MQ do not impose any restrictions on the resources to which work items can be reallocated and any reassignments that a resource makes may be potentially inconsistent with the work distribution strategy implied by the process model. COSA provides an authorisation framework over work items in addition to the work distribution mechanism. Where a work item is reallocated to another resource, that resource must have the required authorisation to execute the task otherwise they will not be able to undertake it and will be required to further reallocate it to a resource that can.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+/-	Only supported for pending (not started) activities
Websphere MQ Workflow	3.4	+	Supprted for pending and suspended items
FLOWer	3.0	-	Not supported
COSA	4	+	Supported though the reroute function
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Any work item can be "reassigned" to a new set of users/group. If the user has updated a task, after the reassignment the data provided by this user is visible to the new assignee, i.e. the state data is not lost

Summary of Evaluation

+ Rating	+/- Rating
1. A resource is able to reallocate a work item on which it commenced execution to another resource without loss of state (i.e. values of current state variables).	1. N/A.

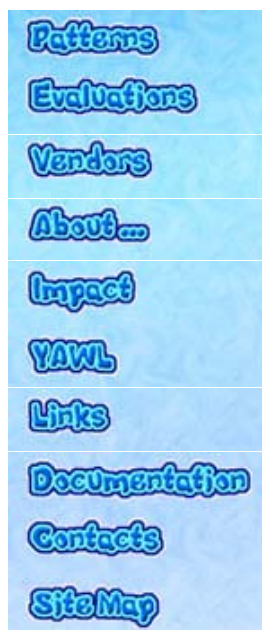
© 2007 *Workflow Patterns Initiative*

0025930

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 31 (Stateless Reallocation)

[FLASH animation of Stateless Reallocation pattern](#)

Description

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

Example

As progress on the *Recondition Engine* work item is not sufficient, it has been reallocated to another *Mechanic* who will restart it.

Related Patterns

[R-PR \(Stateful Reallocation\)](#), [R-D \(Delegation\)](#), [R-E \(Escalation\)](#)

Motivation

Stateless reallocation provides a lightweight means of reallocating a work item to another resource without needing to consider the complexities of state preservation. In effect, when this type of reallocation occurs all state information associated with the work item (and hence any record of effective progress) is lost and the work item is basically restarted by the resource to which it is reassigned. This Pattern is illustrated by the R:reallocation_no_state arc in Figure [10](#). It has similarities in terms of outcome with Delegation and Escalation Patterns in that the work item is restarted except that in this scenario, the work item has already been partially executed prior to the restart. This Pattern can only be implemented for work items that are capable of being redone without any consequences relating to the previous execution instance(s).

Implementation

None of the workflow engines directly implement this approach to reallocation.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM does not allow task rollback, thus offering no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
1. A resource is able to reallocate a work item on which it commenced execution although state information is not preserved (and the work item must be recommenced).	1. N/A

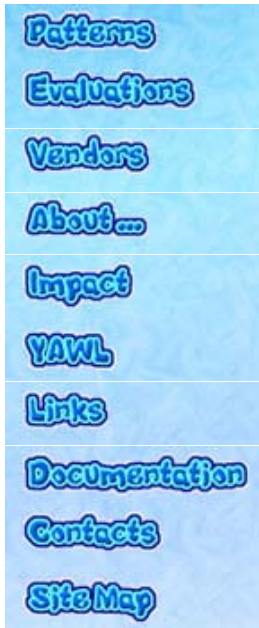
© 2007 *Workflow Patterns Initiative*

0025930

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 32 (Suspension/Resumption)

[FLASH animation of Suspension/Resumption pattern](#)

Description

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

Example

The *Secretary* has suspended all *Board Meeting* work items whilst the *Board* is being reconstituted.

Related Patterns

None

Motivation

In some situations, during the course of executing a work item, a resource reaches a point where it is not possible to progress it any further. Suspension provides the ability for the resource to signal a temporary halt to the workflow engine of any work on the particular work item and switch its attention to another. The work item remains in the resource's work list but is generally notated as suspended. It is able to be restarted at some future time. This Pattern is illustrated by the R:suspend and R:resume arcs in Figure [10](#).

Implementation

This Pattern is implemented in a variety of different ways. Staffware allows work items that utilise a form to be suspended at any stage via the Keep option. Kept work items stay on the resource's work list and can be re-started later. WebSphere MQ doesn't allow individual work items to be suspended but does support the suspension of an entire workflow case. COSA directly supports the notion of suspension and where a work item is suspended, it is removed from the resource's work list and placed in a resubmission queue. At the point of suspension, a timeframe is nominated and after this has expired, the work item is again placed on the resources work list.

Issues

One issue that can arise for suspended items that remain in a shared queue is whether they can be executed by other resources that may have access to the same queue.

Solutions

This situation arises in Staffware and is actually used as a means of sharing a work item to which several resources may wish to contribute. When an item is suspended, all data that is associated with the work item (e.g. form data elements) are saved and become available to any other resource that may wish to resume the task. Any resource that can access a work item can signal its completion via the Release function.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+/-	Supported for activities that have forms associated with them
Websphere MQ Workflow	3.4	+/-	Indirectly supported via case suspension
FLOWer	3.0	-	Not supported
COSA	4	+	Resources can suspend work items currently being executed
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. "suspend" and "resume" are available actions in the worklist application

Summary of Evaluation

+ Rating	+/- Rating
1. Resources can suspend the execution of a work item on which they are currently working. It remains allocated to them. They can resume it at a later time of their choosing.	1. Resources can suspend a work item by suspending execution of the entire case to which it relates. 2. A resource can suspend a work item but it may be picked up for execution by other resources.

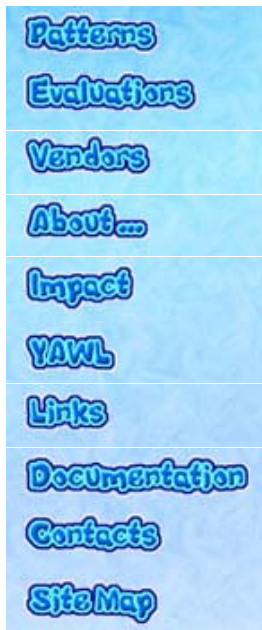
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 33 (Skip)

[FLASH animation of Skip pattern](#)

Description

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

Example

The *Ground Curator* has elected to skip the *Roll Pitch* work item previously allocated to it.

Related Patterns

None

Motivation

The ability to skip a work item reflects the common approach to expediting work processes by simply ignoring non-critical activities and assuming them to be complete such that subsequent work items can be commenced. This Pattern is illustrated by the R:skip arc in Figure [10](#).

Implementation

WebSphere MQ, FLOWer and COSA directly support the ability for a resource to skip work items allocated to them with the process client application.

Issues

The main consideration that arises where work items could potentially be skipped is how to deal with data gathering requirements (e.g. forms that need to be completed by the resource) that are embodied within the work item. In the situation where a work item is skipped, it is generally just marked as complete and no execution is attempted. Subsequent work items that may be expecting data elements or other side-effects resulting from the skipped work item could potentially be compromised.

Solutions

Where a workflow system supports the ability for work items to be skipped, it is important that subsequent work items do not necessarily rely on the output of previous work items unless absolutely necessary. The use of static data elements such as default parameter values can avoid much of the consequences of data not being received. More generally however in order to avoid these problems, the ability is

required within a workflow system to specify work items that must be completed in full.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	+	Directly supported in the worklist handler
FLOWer	3.0	+	Directly supported via skip action
COSA	4	+	Directly supported via skip option in worklist handler
iPlanet	3.1	-	Achieved by changing the status of a work item to COMPLETED where it is in the PENDING state
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. As such any work item can be "withdrawn" by the task creator or the administrator. However, there is also possibility to model a user-action "skip", which marks a work item as completed and passes the flow of control to the subsequent task.

Summary of Evaluation

+ Rating	+/- Rating
1. A resource can skip the execution of a work item allocated to it by marking it as completed.	1. N/A

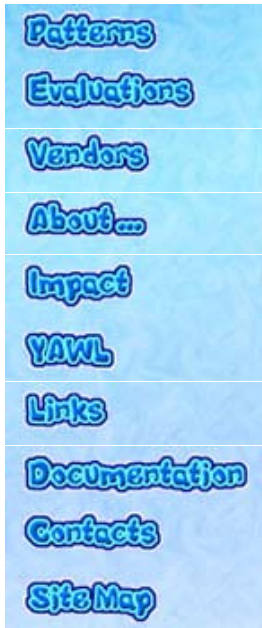
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 34 (Redo)

[FLASH animation of Redo pattern](#)

Description

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

Example

The *Inspector* has decided to redo the *Interview Key Witness* work item.

Related Patterns

None

Motivation

The Redo Pattern allows a resource to repeat a work item that has previously been completed. This may be based on a decision that the work item was not undertaken properly or because more information has become available that alters the potential outcome of the work item. This Pattern is illustrated by the R:redo arc in Figure [10](#).

Implementation

Of the workflow systems examined, only FLOWer provides the ability to redo a previously completed work item.

Issues

Redoing a previously completed work item can have significant consequences on the execution of a case. In particular, the validity of any subsequent work items is questionable as redoing a preceding work item may impact data elements utilised by these work items during their execution.

Solutions

FLOWer addresses this issue by requiring any work items that depend on a "redone" work item to also be repeated before the case can be marked as complete.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.

- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Users can redo a work item that has already been completed
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
<ol style="list-style-type: none"> 1. A resource can repeat the execution of a work item that has already been completed. 2. Any subsequent work items are also repeated. 	<ol style="list-style-type: none"> 1. Subsequent work items are not repeated.

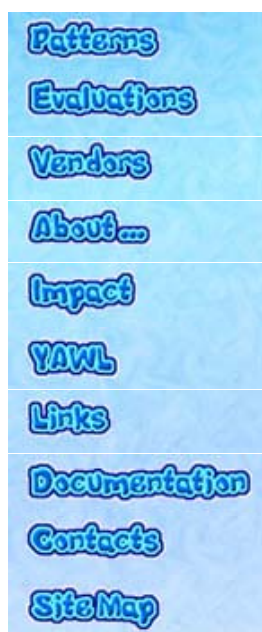
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
 (webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 35 (Pre-Do)

[FLASH animation of Pre-Do pattern](#)

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.

Example

The *Inspector* has completed the *Charge Suspect* work item even though the preceding *Interview Witness* work items have not yet been completed.

Related Patterns

None

Motivation

The Pre-Do Pattern provides resources with the ability to complete work items in a case ahead of the time that they are required to be executed i.e. prior to them being offered or allocated to resources working on the case. This Pattern is not illustrated in Figure [10](#).

Implementation

Of the workflow systems examined, only FLOWer provides the ability to pre-do a work item.

Issues

One consideration associated with pre-doing work items is the fact that outcomes of preceding work items that are executed after the time at which the "pre-done" work item is completed may result in the "pre-done" work item being repeatedly re-executed.

Solutions

There is no immediate solution to this problem other than careful selection of work items that are to be done in advance. As a general rule, work items that are to be "pre-done" should not be dependent on data elements that are shared with preceding work items or the outcome of these work items.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Users can undertake a work item ahead of the wavefront
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
1. A resource can execute a work item that is ahead of the current execution point of the case.	1. N/A

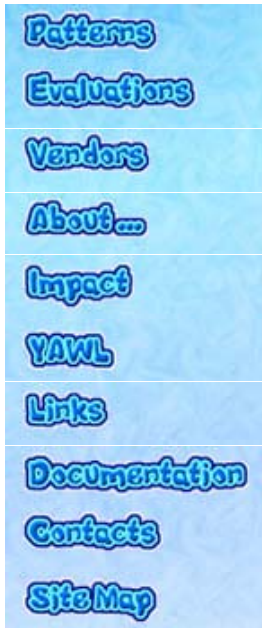
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
 (webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 36 (Commencement on Creation)

[FLASH animation of Commencement on Creation pattern](#)

Description

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

Example

The *End of Month* work item commences execution as soon as it is created.

Related Patterns

[R-CE \(Chained Execution\)](#)

Motivation

The ability to commence execution on a work item as soon as it is created offers a means of expediting the overall throughput of a workflow case as it removes the delays associated with allocating the work item to a suitable resource and also the time that the work item remains in the resource's work queue prior to it being started. This Pattern is illustrated by the transition S:start_on_create in Figure [11](#).

Implementation

All workflow engines which support automatic work items (i.e. work items that can execute without requiring allocation to a resource) provide limited support for the notion of commencement on creation. More complex however is the situation where a work item must be allocated to a resource as this implies that both creation and allocation must occur simultaneously. COSA can support this method of operation where a work item is initiated via a trigger. It provides for a work item to be created and assigned to a specific resource in the same command.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+	Supported for tasks initiated via a trigger
iPlanet	3.1	-	Not supported
BPMN	1.0	+	An activity is assumed to be live as soon as it receives the specified Start Quantity control flow tokens
UML	2.0	+	Directly supported. An action is assumed to be live as soon as it receives a control-flow token
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern since a resource needs to "accept" or "acquire" a work item from the worklist in order to start the execution

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine supports simultaneous creation, allocation and commencement of a work item.	1. N/A

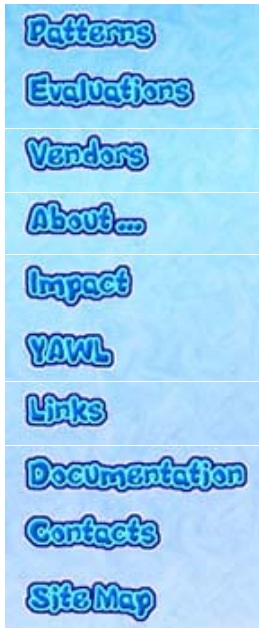
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 37 (Commencement on Allocation)

[FLASH animation of Commencement on Allocation pattern](#)

Description

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

Example

Work on the *Fight Tower Block Fire* work item commences as soon as it is allocated to a *Fire Team* resource.

Related Patterns

[R-LD \(Late Distribution\)](#)

Motivation

Although combined creation, allocation and commencement of work items promotes more efficient workflow throughput, it effectively requires "hardcoding" of resource identities in order to manage work item allocation at creation time. This obviates much of the advantage of the flexible resource assignment strategies offered by workflow systems. Commencing work items at the point of allocation does not require resource identity to be predetermined and offers a means of expediting workflow throughput without necessitating changes to the underlying process model. This Pattern is illustrated by the transition `S:start_on_allocate` in Figure [11](#).

Implementation

The potential exists to implement this Pattern in one of two ways:

- Commencement on allocation can be specified within the workflow model.
- Individual resources can indicate that items in their work list are to be initiated as soon as they are received.

WebSphere MQ provides support for the second approach.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	+	Resources can configure work queues to initiate work items on arrival
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern since a resource needs to "accept" or "acquire" a work item from the worklist in order to start the execution

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine supports automatic commencement of a work item at the point of allocation.	1. N/A

© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 38 (Piled Execution)

[FLASH animation of Piled Execution pattern](#)

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.

Example

The next *Clean Hotel Room* work item can commence immediately after the previous one has finished and it can be allocated to the same *Cleaner*.

Related Patterns

[R-CE \(Chained Execution\)](#)

Motivation

Piled execution provides a means of optimising task execution by pipelining instances of the same task and allocating them to the same resource. The resource undertakes work items sequentially and once a work item is completed, if another work item of the same type is present in the work queue, it immediately commences work on it - in effect it attempts to work on *piles* of the same types of work items. The aim with this approach to work distribution is to allocate similar work items to the same resource which aims to undertake them one after the other thus gaining from the benefit of exposure to the same type of task. This Pattern is illustrated by the transition R:piled_execution in Figure [11](#). It is important to note that this transition is represented by a dashed line because it jumps from one work item to another, i.e., it links the life-cycles of two different work items.

Implementation

To implement this Pattern requires like work items to be allocated to the same resource and the ability for the resource to undertake related work items on a sequential basis, immediately commencing the next one when the previous one is complete. This is a relatively sophisticated requirement and none of the workflow engines examined support it.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine is able to allocate and initiate work items associated with the same task to the same resource such that the completion of one work item results in automatic initiation of the next work item.	1. N/A

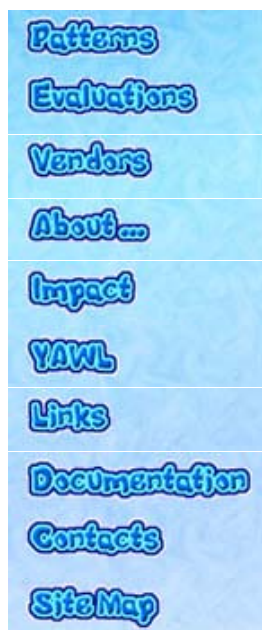
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
 (webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 39 (Chained Execution)

[FLASH animation of Chained Execution pattern](#)

Description

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

Example

Immediately commence the next work item in the *Emergency Rescue Coordination* process when the preceding one has completed.

Related Patterns

[R-PE \(Piled Execution\)](#)

Motivation

The rationale for this Pattern is that case throughput is expedited when a resource is allocated sequential work items within a case and when a work item is completed, its successor is immediately initiated. This has the effect of keeping the resource constantly progressing a given case. This Pattern is illustrated by the transition R:chained_execution in Figure [11](#). Note that, similar to Pattern R-PE (Piled Execution), the transition is dashed because it connects the life cycles of different work items.

Implementation

In order to implement this Pattern effectively, the majority (if not all) of the work items for a given case need to be allocated to the same resource and it must execute them in a strict sequential order. This approach to work distribution is best addressed by a case handling system and not surprisingly FLOWer offers direct support for it.

Issues

Chained execution offers a means of achieving rapid throughput for a given workflow case however in order to ensure that this does not result in an arbitrary delay of other cases, it is important that cases are distributed across the widest possible range of resources and that the distribution only occurs when a resource is ready to undertake a new case.

Solutions

This issue is managed in FLOWer by defining Work Profiles that distribute cases

appropriately and ensuring that resources only request new case allocation when they are ready to commence the associated work items.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Directly supported by Open Action Mode setting
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	+	Once an activity is completed, subsequent activity(ies) receive a control flow token and are triggered immediately when the specified Start Quantity of tokens is reached
UML	2.0	+	Directly supported. Once an action is completed, subsequent actions receive a control-flow token and are triggered immediately
Oracle BPEL	10.1.2	-	Although Oracle BPEL PM offers the "continue task" pattern which allows one workflow to be continued with another workflow, the transition between the workflows is not automatic and requires a work item to be selected for the worklist. Therefore, Oracle BPEL PM offers no support for this pattern

Summary of Evaluation

+ Rating	+/- Rating
1. The ability of the workflow engine to automatically trigger the execution of subsequent work items in a case once the preceding work item is finished. Subsequent work items may not necessarily be routed to the same resource.	1. N/A

Workflow Patterns



Pattern 40 (Configurable Unallocated Work Item Visibility)

[FLASH animation of Configurable Unallocated Work Item Visibility pattern](http://www.workflowpatterns.com/patterns/resource/visibility/wrp40.php)

Description

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

Example

The *Process Worker* can only see the unallocated work items that may be subsequently allocated to them or they can volunteer to undertake.

Related Patterns

None

Motivation

The Pattern denotes the ability of a workflow engine to limit the visibility of unallocated work items - either to potential resources to which they may subsequently be offered or allocated, or to completely shield knowledge of created but not yet allocated work items from all workflow resources.

Implementation

None of the workflow engines examined support this Pattern.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern, since any user can see all unallocated work items and there is no option to limit the visibility of unallocated items

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine provides the ability to configure the extent to which knowledge of unallocated work items is accessible to workflow users.	1. N/A

© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 41 (Configurable Allocated Work Item Visibility)

[FLASH animation of Configurable Allocated Work Item Visibility pattern](#)

Description

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

Example

All *site workers* can view the allocated work items list for the day.

Related Patterns

None

Motivation

The Pattern indicates the ability of a workflow engine to limit the visibility of allocated work items.

Implementation

Of the workflow engines examined, only FLOWer provides support for this Pattern. It does this by limiting the visibility of allocated work items to those resources that have the same role as the resource to which a work item is allocated.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	+	Supported through use of roles to limit visibility of case activities
COSA	4	-	Not supported
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	-	Oracle BPEL PM offers no support for this pattern, since any user can see all allocated work items and there is no option to limit the visibility of allocated items

Summary of Evaluation

+ Rating	+/- Rating
1. The workflow engine provides the ability to configure the extent to which knowledge of allocated work items is accessible to workflow users.	1. N/A

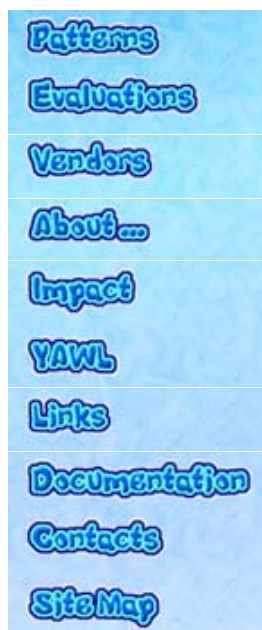
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site](#)
[Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
(webmaster@workflowpatterns.com)

Workflow Patterns



Pattern 42 (Simultaneous Execution)

[FLASH animation of Simultaneous Execution pattern](#)

Description

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

Example

The *Bank Teller* can conduct multiple *foreign exchange* work items at the same time.

Related Patterns

None

Motivation

In many situations, a resource does not undertake work items allocated to it on a sequential basis, but rather it commences work on a series of work items and multi-tasks between them.

Implementation

All of the workflow engines examined support the ability for a resource to execute multiple work items simultaneously. In most tools, the resource can undertake any combination of work items although FLOWer (being a case handling tool) limits the group of simultaneous work items to those which comprise the activities in a dynamic plan.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.

- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	+	Directly supported. All resources can execute multiple activities simultaneously
Websphere MQ Workflow	3.4	+	Resources can execute multiple work items simultaneously
FLOWer	3.0	+/-	Only for elements of a dynamic plan
COSA	4	+	Resources can execute multiple work items simultaneously
iPlanet	3.1	+	Resources can execute multiple work items simultaneously
BPMN	1.0	+	There are no constraints on how many instances of a task specified for one Swimlane can be active at any time
UML	2.0	+	Directly supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern partially by allowing a resource to work with multiple browsers related to a single worklist, and thus enabling and executing several work items simultaneously

Summary of Evaluation

+ Rating	+/- Rating
1. Resources are able to execute more than one work item simultaneously	1. N/A

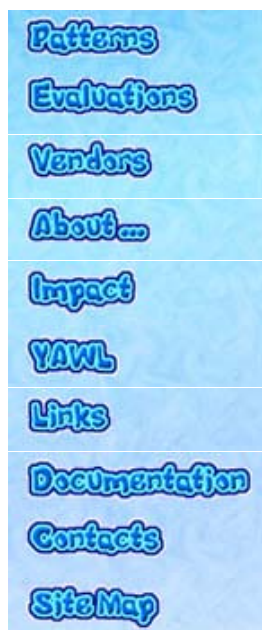
© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com

Workflow Patterns



Pattern 43 (Additional Resources)

[FLASH animation of Additional Resources pattern](#)

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.

Example

The *Blast Furnace Operator* has requested additional *Propane Gas Supplies* before continuing with the *Alloy Preparation* work item.

Related Patterns

None

Motivation

In more complex scenarios, a given work item may require the services of multiple resources in order for it to be completed (e.g. a machine operator, machine and fuel). These resources may be durable in nature and capable of continual reuse or they may be consumable. By providing the ability to model scenarios such as these, workflow engines provide a more accurate depiction of the way in which work is actually undertaken in a production environment.

Implementation

None of the tools examined provide direct support for this requirement in a production context, however COSA offers limited simulation capabilities which allow the operation of a workflow to be evaluated. Included with the simulation environment is the ability to model the various operational resources required by a task - both durable and consumable - together with the associated rate of use on a task-by-task basis.

Issues

None identified.

Solutions

N/A.

Product Evaluation

- To achieve a given + rating, a workflow engine must demonstrate that it complies with each of the criteria specified.
- To achieve a +/- rating it must satisfy at least one of the criteria listed.
- Otherwise a - rating is recorded.

Product/Language	Version	Score	Motivation
Staffware	9	-	Not supported
Websphere MQ Workflow	3.4	-	Not supported
FLOWer	3.0	-	Not supported
COSA	4	+/-	Simulation environment provides multiple resource modelling capabilities for a single work item
iPlanet	3.1	-	Not supported
BPMN	1.0	-	Not supported
UML	2.0	-	Not supported
Oracle BPEL	10.1.2	+	Oracle BPEL PM supports this pattern directly. It offers an "ad hoc" pattern which allows assigning the task to any other user run-time and "request for more information" from other users and have them submit information for tasks.

Summary of Evaluation

+ Rating	+/- Rating
1. A resource may request allocation of additional resources during execution of a work item.	1. Facilities exist to simulate complex resource requirements during workflow execution.

© 2007 *Workflow Patterns Initiative*

0025931

[Patterns](#) | [Evaluations](#) | [Vendors](#) | [About](#) |
[Impact](#)
[YAWL](#) | [Links](#) | [Documentation](#) | [Contacts](#) | [Site Map](#)

For any problems or questions, please [contact us](#)
Webmaster: Jessica Prestedge
webmaster@workflowpatterns.com