All That Glitters Is Not Gold: Selecting the Right Tool for Your BPM Needs

by Nick Russell, Wil M.P. van der Aalst, and Arthur H.M. ter Hofstede

As the BPM marketplace continues its rapid evolution, there is an increasing array of technology offerings available for modeling and enacting business processes. Yet despite the advances that have been made in the process technology area, it is more difficult than ever for organizations to select an appropriate tool on which to base their business processes. These difficulties stem from two major causes: (1) the increasing diversity of offerings that fall under the BPM technology umbrella, and (2) the complexity associated with reconciling the needs of the organization and the capabilities of available products.

As illustrated in Figure 1, the potential range of technologies on which a BPM solution might be based is incredibly diverse, and the suitability of any given tool is influenced markedly by both the degree of flexibility that the to-be-enacted process demonstrates and the nature of the resources (human and/or system) that need to be coordinated.

Moreover, the capabilities of individual tools differ significantly, and one of the main difficulties organizations experience when evaluating individual offerings is finding a suitable basis for comparison. The fact that each tool is usually based on a distinct modeling and enactment formalism, and that vendors often choose to use varying terminology for the same concepts, only serves to further complicate the issue.

What is required is a means of benchmarking the capabilities of a BPM solution in a manner that is independent of specific technological and implementation considerations. This would allow the capabilities of individual BPM tools to be directly compared and would provide a basis for assessing the ability of specific products to meet your organization's specific BPM needs. In the following pages, we present a framework for doing just that.

THE SCOPE OF A BUSINESS PROCESS

Central to establishing a set of benchmarks for BPM solutions is the issue of setting the scope for these benchmarks. It seems self-evident that the benchmarks

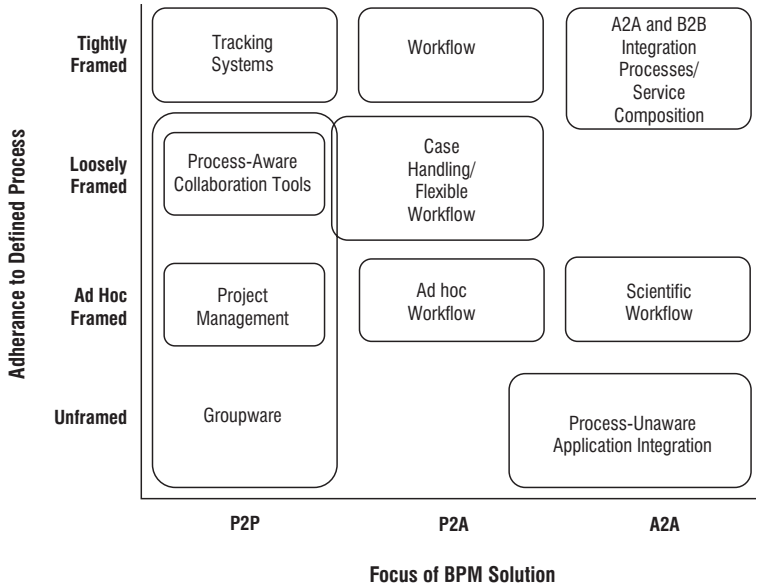

Figure 1 — The wide range of BPM-relevant technologies.

should be framed with reference to the notion of a business process; however, there is a surprisingly wide range of views as to what constitute the relevant components of a business process, both for modeling and enactment purposes. This diversity is reflected in the broad range of models that underpin distinct BPM offerings.

In order to circumvent these considerations, we take a broad view of a business process and consider it to be composed of three distinct (but interrelated) perspectives:

1. **The control-flow perspective**, which describes the structure of a business process in terms of its constituent activities; the manner in which the process is implemented (considering both activities that have a direct implementation and also those that are defined in terms of a subprocess); and the interconnections between them in terms of the overall flow of control
2. **The data perspective**, which describes how data elements are defined and utilized during the execution of a business process
3. **The resource perspective**, which describes the overall organizational context in which a business process functions and the manner in which individual activities can be assigned to human resources for subsequent execution

By setting a comprehensive basis for characterizing business processes, we allow a wide range of factors to be considered when establishing benchmarks. The process of determining individual benchmarks is based on the identification of components within business processes that have generic applicability and are recurrent in form. We call these components "patterns."

RECURRENT COMPONENTS (I.E., PATTERNS)

In an effort to gain a better understanding of the fundamental concepts underpinning business processes, the Workflow Patterns Initiative was conceived in the late 1990s with the goal of identifying the core architectural constructs inherent in process technology. Our original objective was to delineate the fundamental requirements that arise during business process modeling on a recurring basis and describe them in a solutions-oriented way.

We and our fellow researchers took a patterns-based approach to describing these requirements, as it offered both a language-independent and technology-independent means of expressing their core characteristics in a form that was sufficiently generic to allow for its application to a wide variety of tools. The use of patterns to identify recurrent concepts in a given domain and propose general solutions to them was first advocated by Christopher Alexander [1] as a means of describing general architectural principles for building design. It was subsequently introduced with great success into the IT domain by the Gang of Four [2], who described a series of software design patterns for object-oriented systems.

In line with these approaches, which are based on a broad survey of existing problems and practices within a particular field, we (and other researchers affiliated with the Workflow Patterns Initiative) identified a basic selection of 20 control-flow patterns [6] through a comprehensive evaluation of workflow systems and process modeling formalisms. These patterns describe a series of common requirements that arise when modeling control-flow structures within a business process. The imperative approach employed in their description ensures that their intent and function are clearly presented without mandating a specific implementation approach. An overriding objective of the patterns was to describe control-flow characteristics that are useful and therefore need to be supported in a given offering. Each pattern is presented using a standard format, which includes the details shown in Table 1.

After almost a decade of research, we and our colleagues have identified more than 120 patterns in the

Table 1 — Standard Pattern Contents

Description	A summary of its functionality
Examples	Illustrative examples of its usage
Motivation	The rationale for the use of the pattern
Overview	An explanation of its operation, including a detailed operational definition where necessary
Context	Other conditions that must hold in order for the pattern to be used in a process context
Implementation	How the pattern is typically realized in practice
Issues	Problems potentially encountered when using the pattern
Solutions	How these problems can be overcome
Evaluation criteria	The conditions that an offering must satisfy in order to be considered to support the pattern

control-flow [4], data [3], and resource [5] perspectives. All of these are relevant for the purposes of benchmarking the capabilities of BPM offerings. In the following sections, we outline the patterns in each of these perspectives.

Control-Flow Patterns

Control-flow patterns describe structural characteristics of a business process and the manner in which the thread of control flows through the process model. There are 43 distinct control-flow patterns, which are divided into nine distinct groups on the basis of their area of focus:

1. **Fundamental control-flow patterns** capture elementary aspects of control-flow.
2. **Branching patterns** describe branching scenarios in processes where the thread of control in a given incoming branch is split into one or more subsequent branches on the basis of criteria specified in the process model.
3. **Synchronization patterns** describe synchronization scenarios in processes where the thread of control in one or more incoming branches is synchronized (and possibly merged) before being passed into a subsequent branch on the basis of criteria specified in the process model.
4. **Multiple instance patterns** delineate situations where there are multiple threads of execution in a process that relate to the same case/activity.
5. **Repetition patterns** describe various ways in which iteration may be achieved in a process.
6. **State-based patterns** reflect situations that are most easily modeled in process languages with an explicit notion of state.
7. **Trigger patterns** define situations where external events are used to synchronize the commencement of an activity.
8. **Cancellation and completion patterns** categorize the various cancellation and forced-completion scenarios that may be relevant to activities within a process.
9. **Termination patterns** address the issue of when the execution of a process is considered to be finished.

In order to illustrate the operation of the control-flow patterns, it is worthwhile to consider an example. The *Deferred choice* pattern operates in the control-flow perspective. It provides a decision point in a given branch of a process where one of two (or more) alternate branches is selected based on the result of an explicit decision. The actual decision of which branch to choose is made at the last possible moment (i.e., when the chosen branch is actually started). It may take into account a variety of factors (not just control-flow considerations, but also data values, resource availability, etc.) and results in an explicit choice made between the various outgoing branches.

As an example, Figure 2 shows a fragment of the "commute to work" process. After the commuter leaves the house, he faces a choice of walking or taking the bus to work. Only one of these options can be chosen, and typically the commuter also takes additional (i.e., environmental) information into account when making the decision, such as whether it is raining and how much time he has for the journey. Hence the deferred choice exists between the "walk to work" and "take the bus" activities, and the deferred choice node marks the point at which the moment of choice exists. Note that unlike the "normal choice" present in all languages, the deferred choice is not determined based on data or some other decision activity; that is, the choice is made by doing.

Data Patterns

From a data perspective, there are a series of characteristics that occur repeatedly when modeling business processes. These can be divided into four distinct groups:

1. **Data visibility patterns** define the scope (i.e., the extent of the process) in which a data element is defined and can be used.
2. **Data interaction patterns** focus on the manner in which data is communicated between active components (e.g., activities, subprocesses, and parent activities) within a process and also between those components and the operating environment in which the process is situated.
3. **Data transfer patterns** describe various means by which the actual transfer of data elements occurs between components in a process.

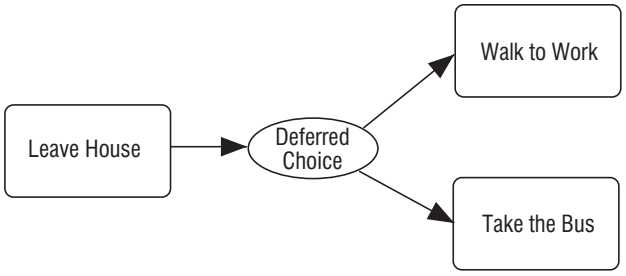

Figure 2 — Example of the *Deferred choice* pattern.

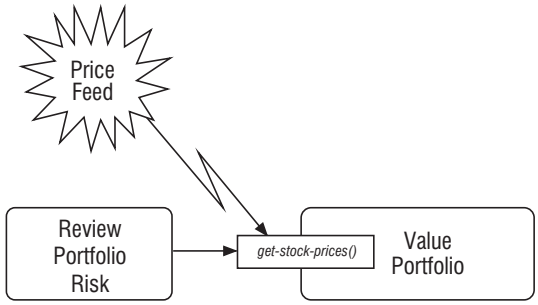

Figure 3 — Example of the *Data transformation – input* pattern.

4. **Data-based routing patterns** characterize the manner in which data elements can influence the operation of other aspects of the process, particularly the control-flow perspective.

Data transformation – input is a data transfer pattern that provides a means of changing the format or value of an incoming data parameter to an activity before (or at the time that) the activity commences. An example of this is illustrated in Figure 3 with the "value portfolio" activity receiving price feed data from the stock exchange at commencement but only requiring price data for the portfolio it is valuing. Hence the `get-stock-prices()` function is called to extract the stock prices for items in the portfolio from all of those that were provided.

This brings us to the third group of patterns, which describe the resource perspective and provide a means of defining how a process (and its constituent activities) should be executed in the organizational context in which it is situated.

Resource Patterns

There are 43 resource patterns, which are divided into seven distinct groups as follows:

1. **Creation patterns** correspond to design-time work allocation directives for individual activities.
2. **Push patterns** are those in which the system proactively distributes activities to human resources.
3. **Pull patterns** describe situations where resources proactively identify and commit to executing specific activities.
4. **Detour patterns** involve the rerouting of activities that have already been distributed to one or more resources, either at the instigation of the resource(s) or the system.
5. **Auto-start patterns** describe the automated commencement of individual activities based on various criteria.
6. **Visibility patterns** describe the observability of activities (and their current status details) to resources associated with a process.
7. **Multiple resource patterns** correspond to work allocations involving multiple participants or resources.

The *Delegation* pattern operates in the resource perspective. Figure 4 illustrates the normal sequence of states through which an activity passes from the time that it is created through to the point at which it is completed by a resource. Usually this involves allocation of the activity to a specific resource, who will undertake it at a later time. *Delegation* provides a resource with a means of reallocating activities that she is unable to complete to another resource for execution.

There are 126 distinct patterns corresponding to the three perspectives described above. Additional patterns have also been defined for other aspects of processes, such as exception handling. The conceptual nature of these patterns means that they provide an excellent basis for describing the capabilities of a BPM solution from a conceptual standpoint. In the next section, we describe the manner in which this is done.

BENCHMARKING TOOL CAPABILITIES

Whilst traditional tool evaluations provide useful insights into product functionality, they often do so at a relatively high level and consequently do not provide a means of evaluating specific capabilities of individual offerings. In contrast, using patterns for benchmarking tool capabilities provides detailed insights into specific abilities and shortcomings of individual tools.

By definition, patterns identify meaningful constructs that exist in a given problem domain. Therefore it is crucial that the identification of patterns be experientially based. Typically this occurs through a survey of their actual occurrence in practice. Our identification of the workflow patterns was no different in this regard, and we based their identification on a comprehensive evaluation of workflow and case-handling systems, business process modeling and execution languages, and Web service composition standards. The process we adopted for identifying and validating individual patterns is illustrated in Figure 5. A crucial part of this activity is the definition of specific pattern assessment criteria that allow the degree of support for individual patterns in a given offering to be evaluated on an objec-

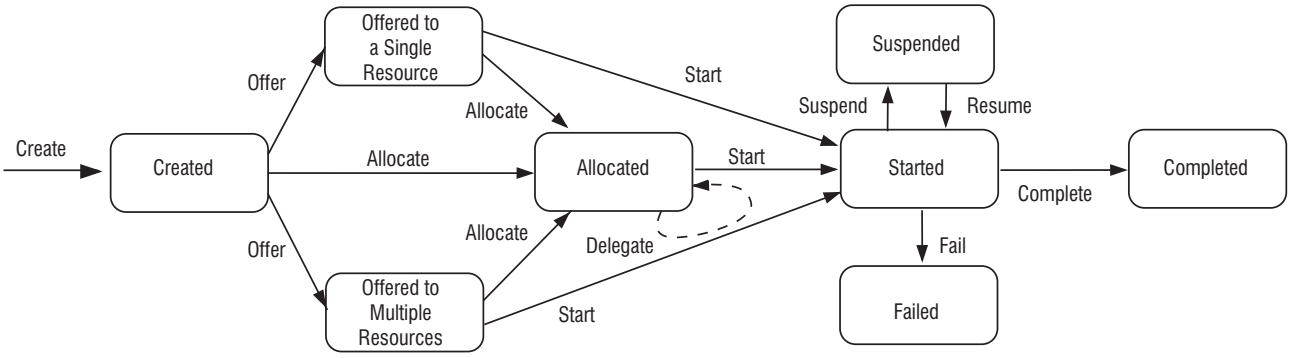

Figure 4 — Illustration of the operation of the *Delegation* pattern in the context of the overall activity lifecycle.

Figure 5 —Pattern identification and validation.

tive basis. Subsequent review of the evaluation results with relevant vendors and domain experts is also vital in order to ensure their correctness and validity.

As a consequence of their technological neutrality, the patterns have proven to be extremely useful for providing a comprehensive assessment of the capabilities of individual products and standards. They have been found to be especially useful for comparing the capabilities of individual offerings in order to identify their strengths and weaknesses, and more generally they provide an effective set of evaluation criteria organizations can use when selecting a BPM tool. The Workflow Patterns Initiative has undertaken a multitude of patterns-based assessments that have revealed problematic aspects of these offerings and provided suggestions for improvement.[1]

Tables 2-4 provide a brief summary of the many patterns-based evaluations of systems and standards we have conducted over the past seven years. As an illustration of the broad applicability of the patterns for benchmarking purposes, we present the evaluation results for a variety of distinct offerings, including:

- Two workflow systems: Staffware Process Suite 10 and WebSphere "classic" 3.4
- A case-handling system: FLOWer 3.5.1
- A business process modeling formalism: BPMN 1.0
- A business process execution language: WS-BPEL 2.0
- A BPEL execution engine: Oracle BPEL v10.1.2.

The results indicate the capabilities of each tool. In each case, we use a three-point evaluation scale, indicating complete (+), partial (+/–), or lack of (–) support for the pattern.

Table 2 summarizes the support for state-based control-flow patterns; that is, just five of the 43 control-flow patterns. The *Deferred choice* pattern, which is one of these, is discussed above. The other patterns are as follows:

- The *Milestone* pattern describes a situation where the execution of an activity depends on the process of which it is part being in a nominated state.
- The *Interleaved routing* pattern describes situations where a set of activities can be executed in any order on a sequential basis.
- The *Interleaved parallel routing* pattern extends this to cover situations where there is an implied partial order in which the activities must be executed.
- The *Critical section* pattern describes the situation where two or more subsections of a process are identified that cannot execute concurrently.

Interestingly, of the offerings examined, the broadest support for this range of patterns is demonstrated by a case-handling system, FLOWer.

[1]Further details on the workflow patterns, including detailed definitions, product evaluations, animations, vendor feedback, and an assessment of their overall impact, can be found at www.workflowpatterns.com.

Table 2 — Support for State-Based Patterns

	Staffware	WebSphere	FLOWer	BPMN	WS-BPEL	Oracle BPEL
Deferred choice	–	–	+	+	+	+
Milestone	–	–	+/–	–	–	–
Interleaved routing	–	–	+/–	+/–	+	+
Interleaved parallel routing	–	–	+/–	–	+/–	+/–
Critical section	–	–	+/–	–	+	+

Table 3 — Support for Data Routing Patterns

	Staffware	WebSphere	FLOWer	BPMN	WS-BPEL	Oracle BPEL
Task precondition — data existence	+	–	+	+	–	–
Task precondition — data value	+	–	+	–	+	+
Task postcondition — data existence	+/–	+	+	+	–	–
Task postcondition — data value	+/–	+	+	–	–	–
Event-based task trigger	+	+/–	+	+	+	+
Data-based task trigger	–	–	+	+	–	–
Data-based routing	+/–	+	+/–	+	+	+

Table 4 — Support for Detour Patterns

	Staffware	WebSphere	FLOWer	BPMN	WS-BPEL	Oracle BPEL
Delegation	+	+	–	–	–	+
Escalation	+	+	–	–	–	+
Deallocation	–	–	–	–	–	+
Stateful reallocation	+/–	+	–	–	–	+
Stateless reallocation	–	–	–	–	–	–
Suspension/ resumption	+/–	+/–	–	–	–	+
Skip	–	+	+	–	–	+
Redo	–	–	+	–	–	–
Pre-do	–	–	+	–	–	–

Table 3 illustrates support for data routing patterns — one of the groups of data patterns — amongst the selected tools. Although the naming of these patterns makes their intent relatively self-evident in most cases, it is worth describing the last three of them to avoid ambiguity:

- The *Event-based task trigger* pattern describes an activity whose execution is contingent on the receipt of a trigger containing a specific data element from the operating environment.
- The *Data-based task trigger* pattern is similar, except that the activity is triggered when an internal data condition is satisfied.
- The *Data-based routing* pattern describes a situation where the routing of control-flow is dependent on conditions specified as part of the process model.

Most of these patterns enjoy relatively broad support amongst the offerings examined, although there are some notable variations.

Table 4 shows the degree of support for detour patterns, one of the groups of resource patterns. The *Delegation* pattern discussed earlier is a member of this group. It is notable that BPMN and WS-BPEL 2.0 do not provide any support for resource-related capabilities. Interestingly, Oracle BPEL does offer a range of capabilities in this area that are not specified as part of the BPEL standard.

Worthy of mention at this point is the YAWL system, an open-source initiative inspired by the identified patterns, which further investigates their implementation and semantics. Further details are available at www.yawl-system.com.

MEETING YOUR NEEDS

We hope the preceding sections have demonstrated how the workflow patterns we've identified can be used to describe the capabilities of individual BPM solutions with a degree of precision that is not possible with other evaluation frameworks. This raises the question of how you can harness the benefits of this research in order to select the most appropriate BPM tool for your needs. To best match the capabilities of available offerings to your requirements, you need to work through the following activities.

Understand Your Business Imperatives

The first step in selecting a BPM solution is assessing what you want the tool for. Although many offerings are relatively flexible and are capable of meeting a broad range of requirements, it is possible that there is no single tool that will meet with all of your needs. Conversely, many of the high-profile solutions offer an extremely broad range of capabilities at a commensurate price, and it's possible that your needs might be adequately met by a less expansive (and, likely, less expensive) offering. To understand your business imperatives, the sort of questions you should be asking are:

- Which business processes will this tool be used to automate?
- What is it coordinating — staff members, software execution, message distribution, external services?
- Who are the stakeholders in this process, and what support do they require in managing it?
- Where do the potential costs/benefits lie?

Identify Mandatory, Important, and Desired Capabilities

With a better understanding of the overall business imperative for acquiring a BPM tool, it becomes possible to think at an operational level about the functions that it will be required to support. The various patterns catalogs — control-flow, data, and resource — provide a useful checklist for identifying specific functional requirements. From a pragmatic standpoint, it is worthwhile to divide these requirements into mandatory, important, and desirable categories, so that there is a scalar across the overall set of functional requirements that ranks their relative degree of importance.

Establish Satisfaction Criteria

In order to ensure that the tool selection process is objective, it is important to define satisfaction criteria before undertaking the tool evaluation. The overall set of selection criteria will probably include a wide range of considerations, but for the purposes of this discussion, we will confine ourselves to those related to the workflow patterns. Possible approaches to specifying satisfaction criteria include scoring approaches based on quantitative pattern support, nomination of mandatory patterns, and comparative rankings.

An important part of this activity is establishing the minimum satisfaction criteria. Where no offerings are identified that meet the satisfaction criteria, there is then the opportunity to consciously review them rather than merely procuring the least unacceptable tool. It's all very well to set tight satisfaction criteria, but if they are so tight that no commercially available products meet them, then you have only two choices: (1) abandon the procurement initiative, or (2) recognize that the require-

ments are too tight and objectively consider which ones you're prepared to relax.

Benchmark Potential Solutions

In many cases, comprehensive patterns reviews are already available for specific BPM offerings.[2] Where this is not the case, you will need to undertake a patterns-based assessment of the tools in which you are interested. There is a multitude of information available on the Workflow Patterns Web site (www.workflowpatterns.com) to assist with this process.

Select the Tool

Finally, it's D-Day! Armed with your benchmark results and satisfaction criteria, you're in a position to select your BPM tool, knowing that the entire process has been undertaken in an objective way.

REAL-WORLD SUCCESS

Several large Dutch organizations have already adopted a patterns-based approach to tool selection with extremely beneficial results. Moreover, we know from experience that a patterns-based approach to evaluating BPM solutions offers insights into the operational characteristics of tools that are difficult to obtain in other ways. By following this approach to selecting a BPM solution, you will know more about your BPM needs and the ability of the offerings you examine to deliver on those requirements.

REFERENCES

1. Alexander, C., S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

2. Gamma, E., R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

3. Russell, N., A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. "Workflow Data Patterns: Identification, Representation, and Tool Support." In *Proceedings of the 24th International Conference on Conceptual Modeling (ER 2005)*, Vol. 3716 of *Lecture Notes in Computer Science*, edited by L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, pp. 353-368. Springer, 2005.

4. Russell, N., A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar. *Workflow Control-Flow Patterns: A Revised View*, BPM Center Report BPM-06-22. BPMcenter.org, 2006 (http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-22.pdf).

5. Russell, N., W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. "Workflow Resource Patterns: Identification, Representation, and Tool Support." In *Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, Vol. 3520 of *Lecture Notes in Computer Science*, edited by O. Pastor and J. Falcão e Cunha, pp. 216-232. Springer, 2005.

6. Van der Aalst, W.M.P. , A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. "Workflow Patterns." *Distributed and Parallel Databases*, Vol. 14, No. 3, July 2003, pp. 5-51.

Nick Russell has 20 years' experience in the Australian IT industry in a variety of technical and senior management roles. During this time, he has led a number of high-profile systems integration and product development initiatives for organizations in the financial and retail sectors. He recently completed his PhD studies at Queensland University of Technology in Brisbane, Australia, and is currently conducting research into business process management and process-aware information systems at the Technische Universiteit Eindhoven in the Netherlands. [any mention of your work with wpi?] Dr. Russell can be reached at N.C.Russell@tue.nl.

Wil van der Aalst is a professor of information systems at the Technische Universiteit Eindhoven in the Netherlands. His research interests include workflow management, process mining, Petri nets, business process management, process modeling, and process analysis. Many of his papers are highly cited, and his ideas have influenced researchers, software developers, and standardization committees working on process support. [work with wpi?] He is committed to open-source initiatives such as ProM and YAWL. Dr. van der Aalst can be reached at w.m.p.v.d.aalst@tue.nl.

Arthur ter Hofstede is an associate professor in the Faculty of Information Technology of Queensland University of Technology in Brisbane, Australia, and is co-leader of the BPM Group in this faculty. Dr. ter Hofstede is committed to the Workflow Patterns Initiative and YAWL. He is a steering committee member of the BPM Conference Series. Dr. ter Hofstede can be reached at arthur@yawlfoundation.org.

[2]See www.workflowpatterns.com/documentation/index.php for a summary of the relevant papers under "Evaluations."