Cost-Informed Operational Process Support

M. T. Wynn[1], H. A. Reijers[2,3], M. Adams[1], C. Ouyang[1], A. H. M. ter Hofstede[1,2], W. M. P. van der Aalst[2,1], M. Rosemann[1], and Z. Hoque[4]

[1] Queensland University of Technology, Brisbane, Australia.
{m.wynn,mj.adams,c.ouyang,a.terhofstede,m.rosemann}@qut.edu.au
[2] Eindhoven University of Technology, Eindhoven, The Netherlands.
{h.a.reijers,w.m.p.v.d.aalst}@tue.nl
[3] Perceptive Software, Apeldoorn, The Netherlands.
hajo.reijers@perceptivesoftware.com
[4] La Trobe University, Melbourne, Australia.
z.hoque@latrobe.edu.au

Abstract. The ability to steer business operations in alignment with the true origins of costs, and to be informed about this on a *real-time basis*, allows businesses to increase profitability. In most organisations however, high-level cost-based managerial decisions are still being made separately from process-related operational decisions. In this paper, we describe how *process-related decisions at the operational level can be guided by cost considerations* and how these *cost-informed decision rules can be supported by a workflow management system*. The paper presents the conceptual framework together with data requirements and technical challenges that need to be addressed to realise cost-informed workflow execution. The feasibility of our approach is demonstrated using a prototype implementation in the YAWL workflow environment.

Keywords: Cost-Informed Process Enactment, Business Process Management, Workflow Management, Process Modelling, Prototype

1 Introduction

Organisations are eager to implement cost-based considerations in their day-to-day operations. In most organisations, however, tying cost considerations to process-related decisions forms a challenge. Our observation is that most Workflow Management Systems (WfMSs) offer no support for cost considerations beyond the use of generic attributes (e.g. FileNet Business Process Manager) or some basic cost recognition and reporting (e.g. TIBCO Staffware Process Suite). Detailed cost information is typically not available at runtime and, as a result, cost information is not used for monitoring or for operational decision support.

Our motivation for this paper is to provide a conceptual framework to enable WfMSs[1] to achieve a higher level of support for cost-informed operational

[1] In the remainder, we use the term WfMS to refer to all process-aware information systems, including Business Process Management Systems (BPMSs).

decisions. More specifically, such a cost-aware WfMS is able to record historical cost information and makes use of it for (real-time cost) monitoring and escalation purposes, as well as supporting simulation and cost prediction capabilities. Ideally, it can also support process improvement decisions based on cost considerations, such as determining cost profiles of different processes/process variants and using them for selection/redesign purposes. To this end, we propose methods for the capture of cost-based decision rules for process, activity and resource selections within business processes and how automated support could be provided for cost-informed process enactment within a WfMS.

It is worth noting that cost is traditionally considered as one of many non-functional requirements (NFR) for a software system or service in the same manner as maintainability, usability, reliability, traceability, quality or safety [2]. However, the cost perspective has a very close and direct link with BPM/WfM, much more so than most other NFRs. First of all, consider that cost is relevant from the viewpoint of individual activities, resources, and entire processes – all of which are in scope for a WfMS. This *versatility* typically does not hold for many other NFRs. Quality, for example, is relevant in the context of a whole process, but not necessarily for a single activity; usability can be tied to a single activity, but not to resources; reliability may be relevant for a single activity, but is too fine-grained for cross-functional processes. Secondly, when we refer to the *dynamic* nature of cost we mean that it is relevant for both design and run time decisions. This aspect differs from NFRs such as maintainability and usability, which are important concerns at design time, but out of scope for operational decision making. Again, both the design and run time perspectives are in scope for a WfMS. In summary, a WfMS is a natural platform to manage cost concerns since it connects the many levels of cost interests and allows for implementing cost-informed design and operational decisions. Hence, we propose a conceptual framework which is tailored towards incorporating the cost perspective within a WfMS with the specific goal of supporting cost-informed operational decisions.

2 A Framework for Cost-Informed Decisions

Different types of actions can be performed by a WfMS or by a resource interacting with a WfMS to support *cost-informed decision making during process execution*. We propose that in addition to the ability to specify cost-informed control flow definitions and resource allocation rules *at design time*, a cost-informed WfMS should provide support for system-based decisions and system-supported user decisions *at runtime*. Figure 1 depicts our conceptual framework which describes 1) *data input*, i.e. the information requirements to enact actions that can be undertaken by or with a WfMS, 2) the *actions* that can be taken on the levels of process, activity, and resource (work distribution), and 3) the *cost-informed support* that is delivered, either through decisions by the WfMS itself or people using its support.

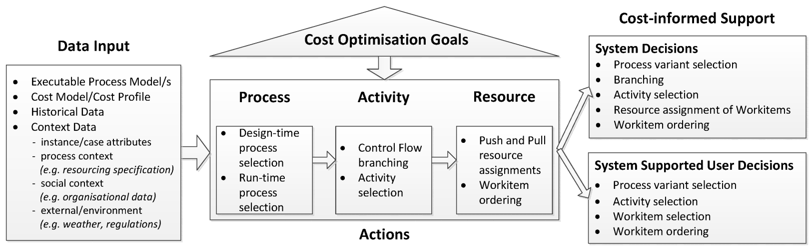

Fig. 1. A framework supporting cost-informed process execution with a WfMS.

2.1 Data Input

A number of key objects need to be provided to a WfMS as *data inputs* to support cost-informed actions. In addition to an *executable process model*, we need access to a *cost model* that can be associated with different elements within a process model. Cost data could be as simple or as complex as an organisation requires it to be. For instance, it could be a variable cost that describes the hourly rate of a resource, but it could also be a dynamic scheme that ties overhead costs to each case depending on seasonal factors. Cost information, together with *historical data* as stored in a so-called *process log* regarding past executions, can be used to determine the cost of process executions as illustrated in our earlier work [14]. Since a business process is always executed in a particular context, we also adopt the four levels of *context data* described in [11]: case attributes, process context, social context, and the environment.

2.2 Actions

All cost-informed actions are based on the data inputs that we discussed on the one hand, while they are governed by the strategic considerations within an organisation on the other. We refer to these as *cost optimisation goals*. Typical examples are: cost minimisation, cost overrun prevention, profit maximisation, incorporation of opportunity cost, etc. The concrete cost-informed actions supported by a WfMS, informed by data input and governed by cost optimisation goals, can then be classified into three levels: process, activity, and resource.

- **Process.** The *process* level is concerned with carrying out *process selection* based on cost information of processes or process variants at design time or at runtime. This may involve the selection among different processes or selection among different process variants (which are created from individual processes during the execution phase). It should also be possible to assign a (whole) process or process variant to a certain resource team for execution (i.e. outsourcing) based on the cost profile.
- **Activity.** For cases that have been started under the control of a WMFS, it is necessary to decide at certain points about the *activity* (or activities) to be executed next. In its coordination capability, a WfMS may decide on which

workitems are enabled in a specific case, based on the branching conditions specified in the control-flow of the underlying process model. A WfMS could also start, skip, and cancel a workitem, among other actions, based on that cost information.

– **Resource.** After a workitem has been enabled, further choices related to distributing work to *resources* become possible. For workitems that need to be carried out by a user, both "push" and "pull" patterns of activity-resource assignment [8] should be supported.

Figure 2 shows possible cost-based decision points within the lifecycle of a workitem (transitions that can be cost-informed are depicted using bold arrows). After a workitem is created, the system can offer the workitem to one or more resources for execution (which is depicted as "S:offer_s" and "S:offer_m" decisions). An additional "C:selection" annotation indicates that it is possible for this system decision to be cost-informed. i.e. a resource could be selected based on its cost characteristics. After a workitem is started by a resource, it can still be suspended/resumed or cancelled by a resource. The "R:suspend", "R:resume", and "R:cancel" transitions reflect these possibilities and similarly the "C:decision" annotations in these transitions indicate that these user decisions can be guided by cost information. When more than one workitem is assigned to a resource and/or when a workitem is offered to multiple resources, a WfMS can provide support for the prioritisation of workitems based on cost information using the concept of cost-based orderings, i.e., "C:ordering".

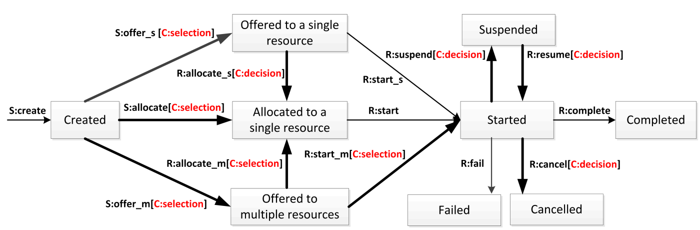

Fig. 2. Lifecycle of a workitem (based on [8]) – enriched with potential cost-based rules for system decisions and system-supported user decisions.

2.3 Cost-informed Support

As we mentioned, our framework identifies the two types of *cost-informed support* that result from the discussed ingredients: *systems decisions*, which can be taken by the WfMS itself, and *system-supported user decisions*, which are taken by resources on the basis of information provided by the WfMS. For instance, it is possible for the WfMS to make an automated selection of the process variant based on its cost profile and context information. Alternatively, the WfMS can provide the resource with cost profiles of different process variants and the resource can make the selection. This is also true for decisions on which activities to execute. The WfMS can either make a cost-informed decision based on a pre-defined business rule to enable/start an activity or allow the resource to start/skip/suspend/resume/cancel a particular activity based on cost information. Decisions on which paths to choose in a process are exclusively taken care of by the WfMS using predefined cost-informed business rules. Workitems can be assigned by the WfMS or can be selected by a resource based on their cost (historical or predicted values).

2.4 Technical Challenges

For a WfMS to be capable of cost-informed enactment, execution and support across the three levels (process, activity and resource), the following key criteria would need to be satisfied:

1. *Association of cost data and cost-based rules with a process/workflow.* This support is required prior to the execution phase. Relevant cost rates for different process elements such as activities and resources must be specified in advance. Some values would include salary and incidental costs for human resources, the costs of materials required, fixed costs associated with activity enactments, rentals, depreciation, and so on.
2. *Runtime calculation of the cost of execution of each process instance and its component activity instances.* Such calculations may be based on time, usage, measurement, invocation, a fixed cost, or a combination of the above.
3. *Logging and analysis of cost data.* The ability to archive all calculated costs for each process instance (incorporated into the process event logs) and to perform extrapolated calculations over archived data.
4. *Support for cost-informed decisions.* The ability to use the calculated cost for the current process instance, and/or those of all previous instances of the process, to make cost-informed decisions.

3 Realisation

We have developed a prototype implementation within the YAWL workflow environment [9]. YAWL was chosen as the implementation platform because it is built on an expressive workflow language that provides extensive support for identified workflow and resource patterns, together with a formal semantics.

A new YAWL custom service, known as the *Cost Service*, is responsible for performing the required cost calculations by applying the data to the relevant cost model components; and for storing all interactions and results in its process logs. The workflow engine and other interested services such as the *Resource Service*, which manages all resourcing allocations, notify the Cost Service throughout the life-cycle of each process instance, passing the appropriate data for cost calculations. The workflow engine has also been extended to accommodate control-flow predicates that include cost-based expressions. When process execution reaches a control-flow predicate that contains a cost-based expression, the workflow engine calls the *Cost Service*, passing the expression, along with all associated data. In addition, the set of resource allocation strategies within the YAWL Resource service has been extended with a number of cost-based allocators, such as *Cheapest Resource, Cheapest to Start, Cheapest Completer* and so on. When the Resource Service enacts a cost-based allocator at runtime, the allocator will directly query the *Cost Service*, requesting a calculation based on previous case histories (stored within the process logs) for the resources involved, based on the particular allocation strategy in question. Both push and pull based resource interaction styles are supported. With regards to process variants, the *Worklet Service* [9] will be extended with cost-based rule expressions, which may then be used to determine which process variant is the ideal selection for the current context of a case.

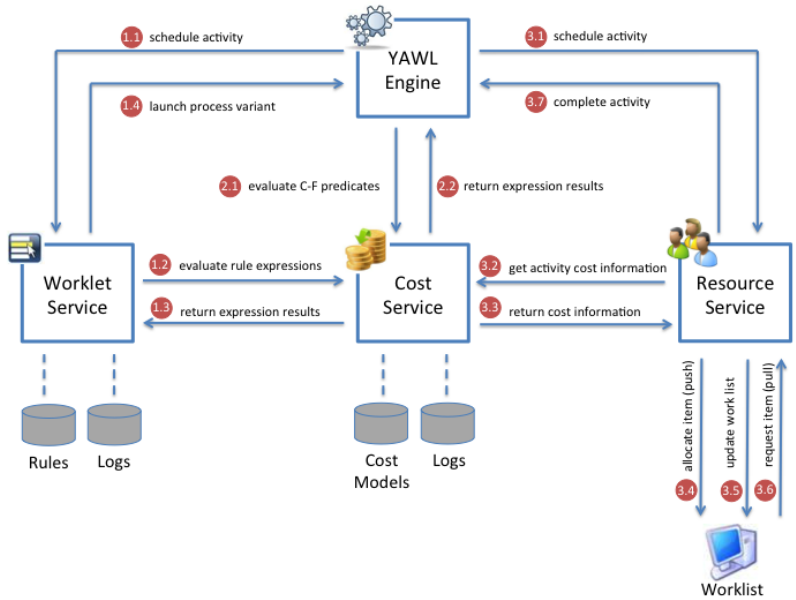

Fig. 3. Prototype architectural flow in the YAWL environment.

Figure 3 shows the flow of information through the prototype for each level of cost-informed support. At the *process* level, the workflow engine schedules an activity for execution by the Worklet Service (1.1). The Worklet Service traverses its rule set for the activity, querying the Cost Service to evaluate cost-based rule expressions (1.2). The Cost Services evaluates and returns the results (1.3), which the Worklet Service uses to select the appropriate process variant for the activity, and launches the variant in the engine (1.4). At the *activity* level, when the workflow engine encounters an branching construct in the control-flow of a process instance, it queries the Cost Service to evaluate the predicate of each outgoing branch (2.1). The engine then uses the results of the predicate evaluations to fire the branch that evaluates to true (2.2). At the *resource* level, where the distribution of work takes place, the workflow engine schedules an activity for a (human) resource (3.1) with the Resource Service. The Resource

Service then queries the Cost Service for all cost information pertaining to the activity (3.2), which the Cost Service returns (3.3). If the activity is configured for system-based allocation (push pattern), the specified allocation strategy (e.g. Cheapest Resource) is employed using the cost information in its calculations, then the activity is routed to the worklist of the selected resource (3.4). If the activity is configured for resource-based allocation (pull pattern), the affected resources' worklists are updated with the retrieved cost information (3.5) allowing a resource to select the appropriate activity based on the cost information presented to them (3.6).

In addition to deploying process examples in the above prototype implementation realising our conceptual framework, we plan to evaluate the conceptual framework with stakeholders' input (e.g. through interviews and case studies).

4 Related Work

Cost has always been one of the key factors under consideration in the context of business process reengineering [5] and process improvements [7]. Through the iterative application of BPM techniques, processes can be improved in terms of quality, flexibility, time and/or cost [7]. Although WfMSs support planning, execution, (re)design and deployment of workflows [13], direct support for cost-informed execution is currently lacking. We have previously taken a first step by proposing a generic cost model [14], which is one of the ingredients of the encompassing framework we presented and demonstrated in the current paper.

The interrelationships between processes, resources and cost are also highlighted in the report produced by the International Federation of Accountants [6]. Notwithstanding these works, few studies exist where a structured approach to the analysis of cost factors in a process-aware information system is undertaken. Since the introduction of ERP systems, a number of studies have been conducted on the effects of ERP systems on traditional management accounting practices [1, 3, 4]. Recently, Vom Brocke et al. proposed an information model to link the ARIS accounting structure with ARIS process semantics using Event Driven Process Chains (EPC) [12]. Cost-informed operational process support is related to the notion of operational support studied in the context of process mining [10]. As shown in this paper, operational support based on cost considerations can be provided though an external cost service tightly coupled to the WfMS.

5 Conclusion and Future Work

The paper proposes a conceptual framework to enable workflow management systems to be cost-informed during enactment. In particular, we proposed how cost-based decision rules for process variant selections, activity related decisions (e.g., execution, cancellation, deferment), and resource assignment decisions can all be supported within a WfMS. We proposed an architecture for cost-informed process execution and presented a realisation of such a cost-informed workflow environment using the YAWL workflow management system.

We believe that our approach will enable organizations to more easily translate cost strategies into operational fulfilment using a WfMS and we have plans to evaluate the framework with stakeholders' input (e.g. through interviews and case studies). This work takes an important step towards achieving a higher level of support for WfMSs in terms of the cost perspective. For the future, we are interested in the development of predictive capabilities that may help to project the cost that is incurred by alternative operational decisions.

Acknowledgments. This work is supported by an ARC Discovery grant with number DP120101624.

References

1. P. Booth, Z. Matolcsy, and B. Wieder. The impacts of enterprise resource planning systems on accounting practice–the Australian experience. *Australian Accounting Review*, 10(22):4–18, 2000.
2. L. Chung, B. Nixon, E. Yu, and J. Mylopoulos. *Non-functional requirements in software engineering*. Kluwer, 2000.
3. S. Grabski, S. Leech, and A. Sangster. *Management accounting in enterprise resource planning systems*. CIMA Publishing, 2009.
4. T. Hyvönen. *Exploring Management Accounting Change in ERP Context*. PhD thesis, University of Tampere, 2010.
5. W. Kettinger, J. Teng, and S. Guha. Business process change: a study of methodologies, techniques, and tools. *MIS quarterly*, 21(1):55–80, 1997.
6. Professional Accountants in Business Committee. Evaluating and improving costing in organizations, July 2009.
7. H. Reijers and S. Mansar. Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega*, 33(4):283–306, 2005.
8. N. Russell, W. van der Aalst, A. ter Hofstede, and D. Edmond. Workflow resource patterns: Identification, representation and tool support. In *Proceedings of the Conference on Advanced Information Systems Engineering*, volume 3520 of *LNCS*, pages 216–232. Springer, 2005.
9. A. ter Hofstede, W. van der Aalst, M. Adams, and N. Russell, editors. *Modern Business Process Automation: YAWL and its support environment*. Springer, 2010.
10. W. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
11. W. van der Aalst and S. Dustdar. Process mining put into context. *IEEE Internet Computing*, 16:82–86, 2012.
12. J. vom Brocke, C. Sonnenberg, and U. Baumoel. Linking Accounting and Process-Aware Information Systems - Towards a Generalized Information Model for Process-Oriented Accounting. *European Conference on Information Systems*, pages 1–13, 2011.
13. M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2007.
14. M. T. Wynn, W. Z. Low, and W. Nauta. A framework for cost-aware process management: Generation of accurate and timely management accounting cost reports. In *Conferences in Research and Practice in Information Technology (CRPIT)*, 2013.