Case Prediction in BPM systems: A Research Challenge

Hajo A. Reijers

Technische Universiteit Eindhoven, Department of Technology Management,
PO Box 513, NL-5600 MB Eindhoven, The Netherlands,
h.a.reijers@tm.tue.nl

Abstract

The capabilities of Business Process Management Systems (BPMS's) are continuously extended to increase the effectiveness of the management and enactment of business processes. This paper identifies the challenge of *case prediction*, which for a specific case under the control of a BPMS deals with the estimation of the remaining time until it is completed. An accurate case prediction facility is a valuable tool for the operational control of business processes, as it enables the pre-active monitoring of time violations. Little research has been carried out in this area and few commercial tools support case prediction. This paper lists the requirements on such a facility and sketches some directions to reach a solution. To illustrate the depth of the problem, a small aspect of the problem is treated in more detail. It involves the complex relations between tasks and resources in business processes, which makes an exact analytical approach infeasible.

Key words: Business Process Management, Workflow Management, case prediction, analysis techniques

---

5 November 2006

1 Introduction

Business process management systems (BPMS's) may result in considerable rewards for the companies adopting them. Typical advantages are: reduced lead times, less hand-off errors, and more flexibility to change business processes. In the research community, there is some consensus that the essential part of a BPMS is the functionality that has been attributed historically to Workflow Management Systems (WfMS's). A WfMS takes care of the automatic allocation of work to qualified and authorized resources – humans and/or applications – in accordance with a predefined schema of the process, the available resources, and their mutual dependencies (see e.g. van der Aalst and van Hee, 2002). Both BPMS's and WfMS's have been widely adopted in industry, in particular in the service industry (Reijers, 2003). Commercial BPMS's are offered by companies such as TIBCO Software, FileNet, Pallas Athena, and Intalio.

While WfMS's are mainly concerned with the *enactment* of business processes, contemporary BPMS's add some additional capabilities. For example, a historical problem with WfMS's has been their "limited interoperability with office applications, meeting specific platform, interface, and operating system requirements" (Georgakopoulos et al., 1995). This shortcoming is countered by today's BPMS's wider capabilities for enterprise application integration (EAI) and Business-to-Business Integration (B2Bi). Moreover, in comparison with their workflow predecessors, BPMS's offer more sophisticated capabilities for real-time monitoring of the evens that occur during execution. This Business Activity Monitoring (BAM) capability has been one of the primary reasons for the updraft of BPMS's in recent years (Gartner, 2002; Gartner, 2004). The goal of BAM is to provide decision makers with timely and accurate information about process execution. Examples of commercially available tools are TIBCO OpsFactor/BusinessFactor, the HP Business Process Intelligence (BPI) tool suite, the ARIS Proces Performance Monitor (PPM), and the TIBCO Staffware Process Monitor (SPM). BAM is concerned with providing a view on the present situation, either to answer questions on the level of an individual case (e.g. "What is the progress on handling Mr. Song's insurance claim?") or on an aggregated level (e.g. "What is the average throughput time of dealing with an insurance claim this month"?).

Given the widespread interest for gathering real-time information on process execution, this paper focuses on a relatively underdeveloped area within BAM: *case prediction*. It concerns *the forecasting of the remaining time that is needed to complete the handling of a specific case that is under the control of a BPMS*. Nowadays, it is common in many industries that products or services must be delivered within the time that is specified in a Service Level Agreement with consumers (SLA). Unfortunately, many uncertain factors affect the speed with which cases can be handled by a BPMS, such as the overall supply of cases, the priorities in dealing with various cases, the availability of resources, the response speed of third parties in delivering essential information or goods, the duration of the individual process steps, etc. etc. But process managers are in dire need of tools that help them anticipate time problems, pro-actively avoid time constraint violations, and make decisions about the relative process priorities and timing constraints when significant or unexpected delays occur (Eder and Pichler, 2002). Currently few BPMS's offer case prediction facilities, the notable exception being the TIBCO Staffware iProcess Suite.

This paper identifies the research challenge of case prediction in the context of BPMS's and gives an overview of the requirements that it should satisfy. Furthermore, it focuses on one of these particular issues that makes case prediction difficult and provides a direction to deal with this difficulty. But first, an overview of the state of the art will be given in the following section.

2 State of the art

Forecasting is widely used in logistics, marketing and modern computer architectures to increase performance by assisting in decision-making and planning. However, it is not so widely investigated or applied in the BPM domain. According to van der Aalst et al. (2003a), there are basically three types of business process analysis:

(1) *Validation* is testing whether the specified business process (or *workflow*) behaves as expected. It focuses on the gap between the specified business process and the intended one. Validation can to be done by domain experts or through the use of process mining, e.g. conformance testing tools(Rozinat and van der Aalst, 2005). BPMS's can provide process mining and simulation tools to assist in validation of the business process.

(2) *Verification* is establishing the correctness of a process model and focuses on the logical correctness of process definitions. Depending on the modelling language used, there may be different properties that must be satisfied. Todays BPMS's only support some syntactical checks at build-time. Verification should be done through the use of methods such as model checking and structural analysis based on the graph structure which can be used to detect inconsistencies. For example, the Staffware Process Definer only checks the linkage of modeling objects during design. A more advanced to tool to check the correctness of Staffware procedures is Woflan. Woflan analyzes workflow process definitions for soundness using Petri-net-based analysis tools (Verbeek et al., 2001). In practice, however, human reasoning or simulation are mostly used to verify the process model.

(3) *Performance analysis* is concerned with evaluating the ability to meet requirements with respect to throughput times, service levels, and resource utilization. Known methods for performance analysis are: business activ-

ity monitoring (BAM), data mining, simulation, and the application of queuing theory.

Case prediction is obviously related to existing performance analysis techniques, but should be clearly distinguished from existing work in this area. Various quantitative techniques have already been proposed for the performance analysis of business processes in the context of BPMS's, e.g. (van der Aalst et al., 2000b; Eder and Pichler, 2002; Ha et al., 2006). A common element in these approaches is that queueing theory is used to arrive at, for instance, estimations of average throughput times of cases, assuming a given process structure including routing probabilities and stochastic durations of tasks. It should be noted, however, that these techniques aim at providing *design-time* support, i.e. to evaluate a process model before it is put into production. In contrast, case prediction is concerned with the *run-time* side of a BPMS, focusing on the remaining execution time of a case that is *already being processed.*

Aside from performance analysis techniques building on queueing theory, other quantitative approaches build on simulation. Traditionally, simulation of business processes is used to support strategic decision making. In this case, simulation is used as a tool to analyze long-term effects of certain decisions. Simulation is rarely used for management control and operational control, because building a simulation model takes too much time to evaluate short-term effects. In earlier work, however, we introduced the concept of *short-term simulation* (Reijers, 2003). Short-term simulation uses the proces definition used by a BPMS as the simulation model and takes as the initial simulation state the current state of the BPMS. One can think of short-term simulation as a quick look in the near future, i.e. a kind of "fast forward" button. By pushing this button, it is possible to see what happens if the current situation in the BPMS is extrapolated. Some years ago, we build a prototype within an industrial setting to show the feasibility of this concept, as described in more detail before (Reijers, 2003). In contrast to case prediction, however, short-term simulation is not concerned with the performance of an individual case, but rather works for the entire population of cases.

The only other work that makes an attempt to integrate forecasting in the BPM domain is by Grigori et al. (2004). They focus on the analysis, prediction, and prevention of the occurrence of deviations from the desired behavior of a business process through the use of decision trees. These decision trees are generated through data mining in specifically designed process analysis tables, which are created from labelled execution logs (the data warehouse). The label on a log – usually attached manually – indicates that it showed a specific (unwanted) effect during execution. Each path from the root to the leaf of the decision tree represents one classification rule which can be used to identify an effect with a specific accuracy. Through real-time application of these classification rules (unwanted) effects can be recognized before they occur without having to forecast the future state of the entire business process. Immediately after recognition of an unwanted effect, the user can be notified or a corrective mechanism that can be triggered. It should be noted that the classification rules (or the decision tree) can only be applied if all attributes of the case needed for application of the rule are known. The value of some of these case attributes, however, will only be determined during execution and are therefore not known in all states of the case.

With respect to the market place, we already noted that only TIBCO Staffware iProcess Suite provides a form of case prediction. Other available BAM tools are not covering case prediction, but merely report on the historic performance of the BPMS or its current state. An analysis of the case prediction functionality of the TIBCO Staffware iProcess Suite, which was carried out in 2005 at the Eindhoven Digital Laboratory for Business Processes (see http://is.tm.tue.nl/research/edlbp/), identified various shortcomings. In the first place, the case prediction only takes into account fixed, constant execution times for all the activities in a process, leading to very inaccurate predictions in cases of great variability. Secondly, the implemented algorithm solely predicts the remaining execution time of a specific case assuming the shortest path to completion. Clearly, this kind of case prediction will deliver in general overly optimistic results.

3 The challenge

The grand challenge of case prediction is that the remaining time to handle a specific case within a BPMS must be provided while taking into account a set of basic requirements:

- The forecast must be highly *accurate*. As revealed in a study conducted by by Yokum and Armstrong (1995), the most important criteria in selecting a forecasting method for decision makers, practitioners, educators, and researchers alike is *accuracy*. The more accurate the forecast, the more accurate the decisions that can be made.
- The case prediction must take place nearly *instantaneously*. It is undesirable to integrate functionality into a BPMS that will require hours or even days of processing, as case prediction must support the operational control of business processes. Clearly, when the estimation of the remaining time to handle a case lasts longer than this remaining time, the forecast is useless.
- The case prediction functionality must be *easy to use*, as its aim is to support business professionals and managers. It is undesirable that an invocation will require deep knowledge from a user on the process itself or quantitative theory. Also, manual operations from the user must be limited to a minimum.
- The case prediction may *not interfere* with the efficient operation of the BPMS. It is undesirable that the invocation of a case prediction request hinders the performance of the BPMS in any significant way.

The first requirement receives our focus in this paper, while we believe that for an actual implementation all requirements must be met in a satisfactory way.

Considering this challenge, three main streams of solutions seem viable:

(1) *Simulation*: The process definition that the BPMS uses could be used as a simulation model for conducting simulation experiments; the current situation could then be used as the initial state of the simulation model; configuration data for the simulation model, e.g. the service times of tasks, could be extracted from the BPMS's database, while other relevant simulation data must be added to such a model. This approach resembles the short term simulation approach as described in (Reijers, 2003).

(2) *Analytical*: The use of an algorithm that applies queueing theory; the BPMS's process definition is then transformed into a queuing network, on which exact and approximation techniques can be applied to determine throughput behavior; parameter settings must once more be derived from the BPMS database or added from another source.

(3) *Heuristic*: An approximate approach which not necessarily takes into account the actual process model; a heuristic may not at all rely (solely) on simulation or queuing theory or may use a mix of simulation and analytical techniques.

We will discuss the various types of solutions one by one. Even though simulation is a highly flexible technique which requires little assumptions on the stochastic behavior of the process, it is unlikely that taking this research direction will result in a case prediction facility that delivers instantaneous results. After all, reliable simulation results require great numbers of replications, which interfere with our requirement on *instantaneous* results. Furthermore, in earlier work we have identified a set of business process characteristics that are hard to capture in simulation models. This is the case, for example, for resources that only work on a part-time bases (see Reijers and van der Aalst, 2005).

The analytical solution direction, in its turn, is hampered by the many assumptions on the queuing network to allow for its analytical evaluation (e.g. see Baskett et al., 1995). Of course, the use of analytical approximations may circumvent such restrictions, but will inevitably lead to less accurate results. It is an open question whether existing approximation techniques for the performance analysis of business process models (e.g. van der Aalst et al., 2000b; Eder and Pichler, 2002; Ha et al., 2006) can be adjusted to take the specific distribution of cases into account as starting point to accurately predict the remaining time in the system of a specific case.

Finally, it should be noted that little work has been carried out in the heuristic domain of business process performance evaluation. This makes it difficult to say whether it will be possible at all to arrive at accurate estimates when the actual process model and the queueing effects that occur are not taken into account or when a hybrid approach is pursued. One could imagine that predictions may perhaps be computed through precalculated branch totals of a business process model. Another direction would be to typify a remaining case time on the basis of case-based reasoning, i.e. to seek for similar cases and return the remaining time they required to become completed from a specific point. Also, the application of regression techniques may be considered, using the data that is logged by the BPMS system on previous executions of cases.

It is an open issue which of the described direction is the most viable. If accurateness is considered as most important requirement, however, it seems reasonable to focus one's intents on finding a good *analytical* technique to make a forecast of the time before that remains before a specific case is completed. However, even when we would be satisfied with an approximate solution, it must be noted that the actual characteristics of business processes that are supported by BPMS's can be very different from those of queuing networks. To deal with the various restrictions that the use of queuing theory brings along, it may be wise to look for ways to simplify the business process model into an analyzable form first. Of course, the problem is which simplifications can be applied that do little harm to the overall accuracy of the case prediction.

In the remainder of this paper, we will investigate only one of the many difficulties when an analytical solution is pursued. This difficulty is related to the observation that in a business process the various resources may be involved in many tasks, while in most queuing models it is assumed that a resource is totally committed to a particular task. For example, in the scheduling literature usually the mapping between steps and machines is either $1:1$, that is, there is a single machine to execute a step, or $1:N$, that is, there are multiple (or parallel) machines to execute a single step. In the domain where BPMS's are applied, the relation is more likely to be $M:N$ (Reijers, 2003). We will refer to this as the phenomenon of *cross-trained resources*.

4 The issue of cross-trained resources

4.1 Introduction

Cross-training refers to the activity of "developing staff capable of performing each others' jobs" (Poyssick and Hannaford, 1996). As major benefit, an organization may expect to improve its agility to deal with the variety of tasks it needs to perform. Developing cross-trained *generalists* may also lead to a more balanced utilization of resources (Reijers and Limam, 2005).

As we stated in the previous section, it is difficult to deal with cross-trained resources in an analytical way. Therefore, it would be very welcome if we could approximate the performance of a business process *with* cross-trained workers by a similar business process where each of the resources is *only* dedicated to one particular task. Before we can present this approach, we will introduce some terminology.

4.2 Terminology

We adopt the terminology and concepts from van der Aalst and van Hee (2002), where a workflow process consists of tasks needed to handle a class of *cases* (also “orders” or “process instances”). A BPMS works on the basis of a predefined *workflow definition*, specifying which tasks need to be performed and in what order. For example, a workflow definition may specify that on receipt of a mortgage application, it must always be registered (task `A`), after which a decision must follow whether or not a mortgage proposal will be issued (task `B`). We refer to a *work item* as a task that needs to be executed for a specific case. For example, a work item may be the registration of Mr. Smith’s mortgage application.

When enacting a business process, a BPMS must ensure that work items are assigned to proper *resources*. In office environments this term primarily refers to human staff members. Usually, two criteria need to be taken into account: a resource must both be *authorized* and *qualified*. In the example of the mortgage handling process, any clerk may be eligible to perform task `A`, regardless of the case in question. On the other hand, only Mr. Smith’s account manager may carry out task `B` for applications by Mr. Smith.

In most BPMS settings, there is more than one resource that may carry out a particular work item (van der Aalst and van Hee, 2002; Tramontina et al., 2004). The usual way of dealing with this situation is that the BPMS makes a work item available to a set of similar resources, until one of these resources selects it for execution. After such a selection, the work item is no longer available to other resources. This mechanism is referred to as the *pull-mechanism* (zur Mühlen, 2004). At the same time, a single resource may be capable to carry out work items associated with different tasks. In other words, there may very well be an $N : M$ relation between resources and tasks in applications of BPMS's (although this is not always the case).

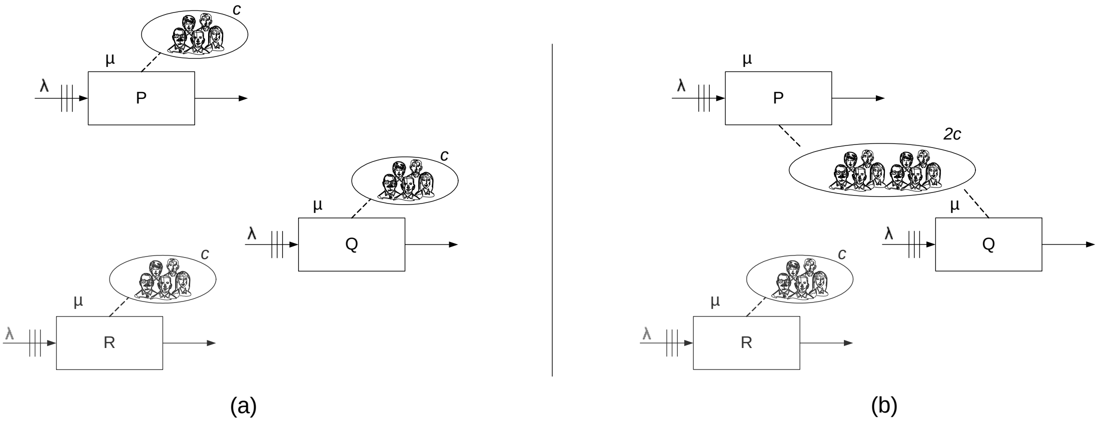


Fig. 1. Abstract workflow process with dedicated resources.

4.3 Approach

Building on the notions as introduced in the previous section, we focus on the abstract workflow process as shown in Fig. 1(a). With this model, we abstract from specific characteristics of real workflow processes, such as topology, resource numbers, work load, service times, etc. Depicted are rectangles labelled `P`, `Q`, and `R` representing a number of tasks that may well be a subset of a larger number of tasks. Some partial order is assumed to exist, for example a sequential ordering of all tasks. To each of these tasks, a dedicated pool of c resources is assigned. Resource pools are shown as ovals. Each resource pool is capable of working on the work items related to the task it is assigned to (and these work items only). We will refer to a set of $n \in \mathbb{N}$ tasks that can be handled by the same pool of resources as a *work center* of size n. A work center of size 1 is called a *dedicated work center*.

At some entry point in the process, new cases arrive which are translated into work items by the BPMS. An arbitrary number of tasks need to be executed for each task. New work items arrive at each task according to a Poisson process with intensity λ. The time it takes to handle a work item, i.e., its *service time*, has a negative exponential distribution with an average of $1/\mu$ time units. Each work item is handled by a single resource. A FIFO selection discipline of work items is assumed, being the most popular dispatching rule in BPMS's (van der Aalst and van Hee, 2002). Completion of one or more activities may lead to the creation by the BPMS of one or more new work items. If some final activity is completed, the handling of a case is finished.

Given the process of Fig. 1(a), the involved management may, for instance, decide to cross-train the resources of the dedicated work centers `P` and `Q`. In this way, work items that arrive at either the queue of task `P` or `Q` can be handled by a resource from either pool (once again in FIFO mode). The process resulting from this measure, including work center `PQ` of size 2, is shown in Fig. 1(b). Note that additional cross-training (or "pooling") may proceed in various ways, for example, by adding resource pool `R` to work center `PQ` creating a work center of size 3 or by combining work centers `S` and `T` (which are not shown).

To properly assess the performance of the various configurations we will take the following approach. We will focus on the average queueing times that work items *locally* experience at a work center. Note that all work items arrive in the same, single queue of a work center. As baseline for further comparisons, we will use the average queueing time at a dedicated work center, e.g. `P` in Fig. 1(a).

Note that it would not provide much insight to consider the performance of the *entire* workflow process, as it would be influenced too much by the specific topology of the process. Also, a focus on *lead time* instead of queueing time

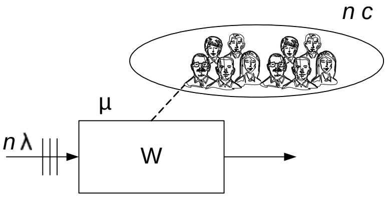
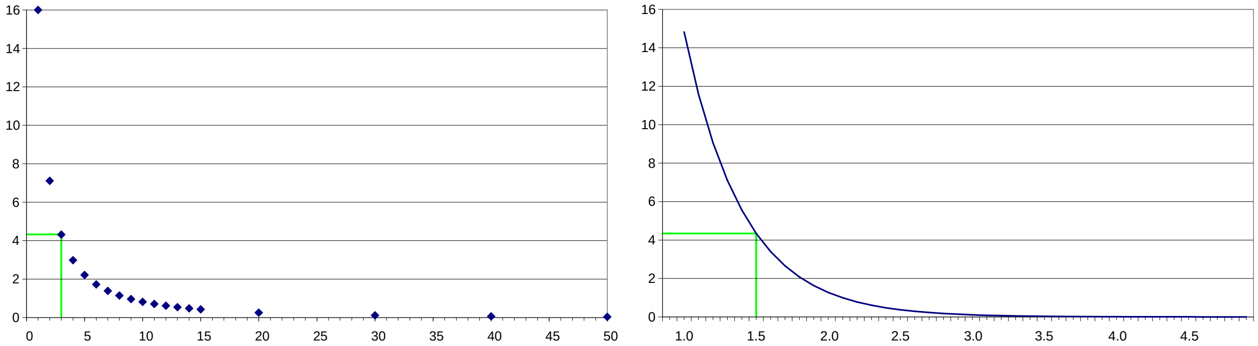


Fig. 2. Single task queueing system.

would have been possible too – assuming it is locally measured at a work center – but this notion unnecessarily incorporates an arbitrary portion of service time. At the same time, our approach is very simplistic as it does not take into account the many difficult patterns that may occur in a process definition (see van der Aalst et al., 2003b).

When choosing a set of specific values for λ, μ, and c it is possible to analytically determine the average queueing time of work items handled by work centers of any size. After all, the performance of a work center of size $n \in \mathbb{N}$, is equivalent in terms of resource utilization and throughput to the single-task queueing system as shown in Fig. 2. Note that this only holds on the basis of the equivalent arrival pattern, service pattern, and handling discipline of the dedicated work centers in Fig. 1(a). The combined work center can be analyzed using the standard formulas for an $M/M/c$ queueing system (Kleinrock, 1975). With $\lambda = 1/5, \mu = 1/4, c = 1$ as values for the baseline system, for example, the average queueing time for different sizes of a work center is shown in Fig. 3(a). As can be seen, the average queueing time of work items handled by a work center of size 3 approximately equals 4.3 time units.

Now it is clear that the exact average queueing time can be determined at work centers of arbitrary size, we consider it to be of interest how an equivalent performance is delivered by a *dedicated* work centers. For example, instead of having a set of cross-trained resources working on both `P` and `Q`, a similar performance could theoretically be delivered by a combination of resources that are completely dedicated to `P` and others who are dedicated to `Q`. If we



(a) for work centers of different sizes

(b) for different numbers of resources at a dedicated work center

Fig. 3. Average queueing times for work centers ($\lambda = 1/5$, $\mu = 1/4$)

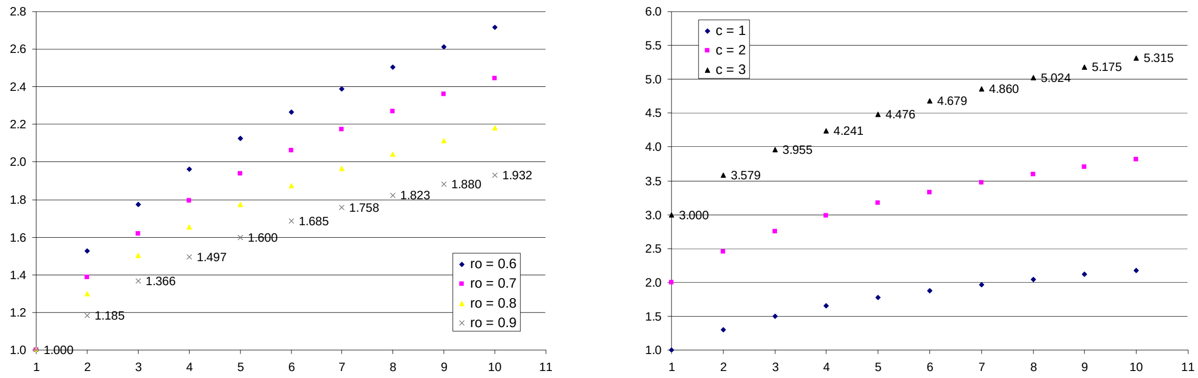
could establish this relation, this would be a valuable step in developing an analytical case prediction facility. Note that at this point in time our focus is on the average throughput time and that we do not consider the vulnerability of a process for disturbances in the arrival of cases. Clearly, in the latter case there *will* be differences in the behavior of the different configurations.

To determine a distributed amount of *dedicated* resources which would deliver an equal performance of a distribution of *cross-trained* resources, we need to settle the issue of *part-time resources*. After all, when we would have a discrete number of cross-trained resources, it would be mere coincidence that we could find a discrete number a distribution of a discrete number of dedicated resources with an equal performance. Rather, we should also consider dedicated resources that are only available for a part of their time to work on tasks, a so-called part-time resource. Obviously, by considering part-time resources, we have to resort to other than analytical queueing models (there is no such thing as the $M/M/\pi$ queueing system).

To determine the performance of a dedicated work center with part-timers we have to turn to discrete event simulation. For a dedicated work center of size $k+l$ with $k \in \mathbb{N}$ and $l \in [0, 1)$, we assume that there are k resources working full-time and one resource that works for $l \times 100\%$ of the time. Without discussing the details here, we have constructed a simulation module within the package `ExSpect` (van der Aalst et al., 2000a) that implements this policy. For the duration of the simulation, it makes the part-time resource available for the proper amount of time, alternating periods of full availability and unavailability. Note that this is but one of the various ways to implement a scenario with part-timers.

By using the simulation module for non-discrete sizes of work centers, it is possible to characterize the performance of a dedicated work center for any given set of values for λ, μ, and c ($c \in \mathbb{R}^+$). To get a characterization of the average queuing time as an exponential function of the number of resources at a dedicated resource center, we followed the procedure to generate a sufficiently large set of measurements by simulation and interpolated these. Note that it takes considerable simulation time to produce reliable results, which could never be instantaneously performed at run-time. A performance characterization determined in this way is given in Fig. 3(b). The shown function can be used to establish that the performance of the earlier example – an average queueing time of 4.3 time units at a work center of size 3, see Fig. 3(a) – could also be achieved by assigning 1.5 resources to each of three dedicated work centers for each of the three tasks.

By now, we have clarified our approach to approximate the performance of a workflow process with cross-trained resources by the performance of a system with totally dedicated resources. We will present some results that are based on this approach in the next section.



(a) At different occupation levels ρ ($\mu = 1/4, c = 1$)

(b) At different sizes of the initial resource pool c ($\mu = 1/4, \rho = 0.8$)

Fig. 4. Number of desired, dedicated resources to equal the performance of cross-trained resources.

4.4 Results

We have applied the described approach to determine the relation between the performance of work centers with cross-trained resources and those with dedicated workers. To configure the baseline system, we have chosen an arbitrary value of $\mu = 1/4$ to set the service intensity and $c = 1$ as initial size of the resource pool. As it can be expected that the relation of interest is influenced by the load of the system, we varied the arrival intensity to create occupation levels of 60%, 70%, 80% and 90%. Previous research indicated that these levels are typical for workflow settings (Reijers and van der Aalst, 2005). The results can be seen in Fig. 4(a). For $\rho = 90\%$ the specific values of the relation are given too. For example, to deliver a performance that is equal to the results of a pool of 6 cross-trained resources that can deal with 6 different tasks, the same performance would be delivered by 6 dedicated work centers for each of the tasks with 1.685 resources each. (In total this means that overall we would need to employ 4.11 resources more in the dedicated case than in the cross-trained alternative.) Note that for lower occupation levels, more dedicated resources at work centers are required. The insight that can be derived here is that dedicated resources are particularly effective when a work center is heavily utilized.

Another property that is of particular influence is parameter c, the initial number of resources of the baseline work center. In Fig. 4(b), the relation can be seen for different values of c, assuming an occupation level of 80% of the baseline system. Also shown are the particular values for the case that the number of resources in the initial resource pool equals 3. For example, if we would have 2 groups of 3 workers each which are cross-trained to perform each other's tasks resulting in a pool of 6 cross-trained resources that can deal with 2 tasks, this would deliver the same performance as 2 groups of 3.579 workers each of which just works on a single task. As can be expected, proportionally more dedicated resources are required at work centers to emulate the performance of pools of cross-trained workers. Note that in practice resource pools may be much larger than we investigated here, which underlines the potential of cross-training from a performance perspective.

4.5 Application

The described approach in this section is, obviously, very simplistic and neglects many realistic issues. Nonetheless, it could be used to transform a complex business process model into a more simple one, i.e. with only dedicated work centers, that delivers approximately the same performance as the realistic business process where cross-trained workers work on several cases. This transformation could be done "off-line", so before the case prediction functionality is actually invoked. In this way, the analysis that needs to be carried out instantaneously will become far less complex. Obviously, when the allocation of resources over the various tasks is highly dynamic over time, the accuracy of this approach will decrease significantly.

5 Conclusion

In this paper, we have introduced the problem of case prediction. Also, we have listed a set of basic requirements a solution must provide and an overview of various research directions. To stress the magnitude of the problem: We have focused on yet a small aspect of business processes that complicates case prediction, i.e. the existence of cross-trained workers. Our approach deals with simplifying a business process with many cross-trained workers into one with only dedicated resources. Many other issues have not been dealt with in this paper, in particular not how a current state of a BPMS can be transferred to a simulation/analytical model to increase the accuracy of the estimation. It seems worthwhile to investigate how current analytical approaches for the performance evaluation of business processes can be adapted to carry out case prediction in practice, e.g. (van der Aalst et al., 2000b; Eder and Pichler, 2002; Ha et al., 2006).

It is suggested that, rather than receiving merit or critique of the presented "replacement" method, this approach should rather be seen as an example of a promising approach to simplify the challenge of accurate case prediction. It is very clear that this research area shows a high level of industrial relevance, with as of yet a very limited number of solutions and few industrial implementations. Perhaps this paper may serve as an inspiration for others to make a next step forward in this area.

Acknowledgement

This work was supported by the Netherlands Organisation for Scientific Research (NWO) and the Korea Science and Engineering Foundation (KOSEF), who provided a grant to visit BPM researchers in South-Korea in 2006.

References

[1] van der Aalst, W.M.P., de Crom, P.J.N., Goverde, R.R.H.M.J., van Hee, K.M., Hofman, W.J., Reijers, H.A., and van der Toorn, R.A. (2000a) ExSpect 6.4: An Executable Specification Tool for Hierarchical Colored Petri Nets, *Proc. 21st Conf. on Application and Theory of Petri Nets*, 455-464.

[2] van der Aalst, W.M.P. and van Hee, K.M. (2002), *Workflow Management: Models, Methods, and Systems*, MIT Press, Cambridge, USA.

[3] van der Aalst, W.M.P., van Hee, K.M., and Reijers, H.A. (2000b), Analysis of Discrete-Time Stochastic Petri Nets, *Statistica Neerlandica*, **54**(2), 237-255.

[4] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P., (2003b), Workflow Patterns, *Distributed and Parallel Databases,* **14**(1), 5-51.

[5] van der Aalst, W.M.P., Weske, M., and Wirtz, G. (2003a), Advanced Topics in Workflow Management: Issues, Requirements, and Solutions, *Journal of Integrated Design and Process Science*, **7**(1), 49-77.

[6] Baskett, F., Chandy, K.M., Muntz, R.R., and Palacios, F. (1975), Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, *Journal of the ACM*, **22**(2), 248-260.

[7] Eder, J. and Pichler, H. (2002), Duration Histograms for Workflow Systems, *Proc. of the IFIP TC8/WG8.1 Working Conference on Engineering Information Systems in the Internet Context,* 239-253.

[8] Gartner (2002), The BPA Market Cathes another Major Updraft, Gartners Research Note M-16-8153

[9] Gartner (2004), Magic Quadrant for Pure-Play BPM, 2Q04, Gartner's Research Note M-22-9774.

[10] Georgakopoulos, D., Hornick, M., and Sheth, A. (1995), An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases* **3**, 119-153.

[11] Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., and Shan, M. (2004), Business Process Intelligence, *Computers in Industry*, **53**(3), 321-343.

[12] Ha, B-H., Reijers, H.A., Bae, J., and Bae, H. (2006), An Approximate Analysis of Expected Cycle Time in Business Process Execution. *Proc. of the 2nd International Workshop on Business Process Design (BPD'06)*, 65-74.

[13] Kleinrock, L. (1975), *Queueing Systems, Volume 1: Theory*, John Wiley and Sons, New York, USA.

[14] zur Mühlen, M. (2004), Organizational Management in Workflow Applications, *Information Technology and Management Journal*, **5**(3-4), 271-291.

[15] Poyssick, G. and Hannaford, S. (1996), *Workflow Reengineering*, Adobe Press, Mountain View, USA.

[16] Reijers, H.A. (2003), *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, Springer-Verlag, Berlin, Germany.

[17] Reijers, H.A. and van der Aalst, W.M.P. (2005), The Effectiveness of Workflow Management Systems: Predictions and Lessons Learned, *International Journal of Information Management*, **56**(5), 457-471.

[18] Reijers, H.A. and Limam Mansar, S. (2005), Best Practices in Business Process Design: An Overview and Qualitative Evaluation of Succesful Redesign Heuristics, *Omega*, 33(**4**), 283-306.

[19] Rozinat, A. and van der Aalst, W.M.P. (2005), Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models, *Proc. of the 1st International Workshop on Business Process Intelligence (BPI05)*, 112.

[20] Tramontina, G.B., Wainer, J., and Ellis, C. (2004), Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems. *Proc. of the 2004 ACM Symposium on Applied Computing*, 1396-1403.

[21] Verbeek, H.M.W., Basten, T., and van der Aalst, W.M.P. (2001), Diagnosing Workflow Processes using Woflan, *Computer Journal,* **44**(4), 246-279.

[22] Yokum, J.T. and Armstrong, J.S. (1995), Beyond Accuracy: Comparison of Criteria Used to Select Forecasting Methods, *International Journal of Forecasting,* **11**(4), 591-597.