Modeling of an Agent System to Support the Management of Cooperating and Rival Resources for Business Workflows

Ágnes Werner-Stark, Tibor Dulai and Gyula Ábrahám
Department of Electrical Engineering and Information Systems, University of Pannonia, Egyetem str. 10., Veszprém, Hungary

Keywords: Modeling, Agent, Workflow, Resource, Scheduling, Log.

Abstract: Modeling and analysis of business workflows may be strategic on behalf of the optimal execution. This paper proposes an innovative model-based approach, which can be used to resource scheduling of business workflows. To this we defined such functions that help the operation of processes, resource scheduling can be described formally during the modeling. The system uses all the information during the scheduling, which may be recorded in a log file in connection with the process execution. We can extract useful information concerning allocation of resources by analysis of the historical data, which are used to assign the resources to the implemented tasks. In the system the cooperation and contention of the resources as agents will play important role. The scheduling can be tested in an agent simulation environment. By the aid of this approach we can give decision proposal to the operator in real time to promote more optimal realization of the workflow.

1 INTRODUCTION

Increasingly, companies around the world are reengineering their core business processes to be more profitable and to improve customer satisfaction. Modeling and analysis are two critical steps in any process redesign effort. Bhaskar et al. (Bhaskar et al., 1994) discuss the need for simulation tools that can be used effectively to model, to document and to analysis business processes. Tumay (Tumay, 1996) concentrates also the simulation of business processes and how it works. In (Ghosh, 2013), the authors deal with a simulation tool which makes the scheduling of service tasks and the distributed human resources possible in case of industrial process. The major pillars by the side of which the model works and tries to minimize the time costs of certain tasks: fluctuation, short-term service demands which depend on availability of the resources, flexible task performance, replacing of the straggling behind tasks.

In our work the re-engineering and the simulation are very important. Re-engineering can be followed closely by the workflow process.

Scheduling is an important and widely used topic of operations research. Besides of its theoretical importance, industries can also benefit from optimal schedules and resource allocation. More and more different algorithms are born for organizing process elements on the time scale (Pinedo, 2012) and (Brucker and Brucker, 2007), related to business process (Barba and Del Valle, 2010) and (Xu et al., 2010), or industrial processes (Sule, 2008). The results are usually represented on a Gantt chart that we use also to illustrate the decision proposal.

Only some publications investigate the cooperation possibilities that are enabled by the similar functionality of resources in scheduling, see e.g., (Murthy et al., 1997) or (Dulai and Werner-Stark, 2012).

Merdan et al. (Merdan et al., 2013) consider the application of intelligent agent-based technology as a promising tool to improve system performance in complex and changeable environments. They examine it in the case of unforeseen events, e.g., machine breakdowns that usually leads to a deviation from the initial production schedule. A multi-agent approach can be used to enhance system flexibility and robustness. In their work the approach to revise and reoptimize the dynamic system schedule in response to unexpected events is proposed. Our goal is also the modeling of a special multi-agent system with the help of which - detecting workflow changes and analyzing historical data - a new decision proposal can be determined referring to a new allocation of resources as the modification of workflow.

So that, we can specify accurately the decision proposal, we used a further method. As under the

Werner-Stark Á., Dulai T. and Ábrahám G..
Modeling of an Agent System to Support the Management of Cooperating and Rival Resources for Business Workflows.
DOI: 10.5220/0005001404070412
In *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2014), pages 407-412
ISBN: 978-989-758-038-3

current process of workflows the most important information is stored in a log file, therefore the current state analysis can be carried out using these logs and in case of unexpected events the multi-agent system can react with a new resource scheduling proposal. Van der Aalst et al. play an important role in area of log analysis see e.g., (van der Aalst et al., 2003) and (van der Aalst et al., 2007).

This paper builds on concepts from workflow management, agent based technology, log analysis and resources scheduling. We have modeled a new solution to the resource scheduling of workflows in random changing execution environment. This paper describes the modeling phase of an agent oriented resource scheduler which is based on log analysis. In our modeled method the role of cooperation and competition of the resources as agents plays an important. By the aid of this approach we can give decision proposal in real time to the operator, promoting more optimal realization of the workflow.

2 BASIC NOTIONS

2.1 Relationship between Workflow and Agent

Business processes are modularized and these modules can be reorganized by the workflows to form new business processes, so as to react quickly to unexpected changing business needs and conditions. The user in the workflow system does not have to deal with the structure of the business process, but only concentrates on the task itself. The current generation of workflow management systems has some shortcomings, e.g.:

- *Lack of Automation*: workflow systems only determine the process logic, but most of the activities are still fulfilled by human. The workflow system can't even start a workflow without human's intervention.
- *Lack of Reactivity*: the workflow systems require a pre-defined representation of a business process and all potential deviations from that process.
- *Lack of Resources Management*: the workflow systems do not control the resources of a business process.

We want to solve these problems by using a new agent-based approach. Agents can represent resources. Task assignments and resource allocations are done through negotiation among these agents. Agents react to changing circumstances and have the ability to generate alternative execution paths. The agents are independent to each other and each of them is responsible for process execution.

2.2 The Function of the Agent Technology in the Scheduling

The benefits of applying agent technology to business process management include:

- *Automation*: The inherent autonomy of software agents can fulfill activities as human substitution.
- *Resource Management*: Agents can represent resources. Task assignments and resource allocations are done through negotiation among these agents.
- *Reactivity*: Agents react to changing circumstances and have the ability to generate alternative execution paths.
- *Intelligent Decision-making*: We can use the learning ability of agents that is helpful in the workflow scheduling.

Our goal is that we integrate these benefits in special solving to a support system to organize cooperating and rival resources.

2.3 Log Analysis

Process mining techniques have emerged in the area of business process modeling (van der Aalst et al., 2004), theoretical computer science and artificial intelligence. Taking the data set of real process executions, the event logs, these techniques can be used for process discovery, i.e., to construct a discrete event model from them, or for conformance checking. An event log is a set of finite event sequences, whereas each event sequence corresponds to a particular materialization of the process. The execution of each event needs some resources which execute the activity in optimal time.

2.4 The Structure of the System

Figure 1 shows the structure of the multi-agent system for scheduling business workflows. To this we modeled a method for supporting of the user decisions, the decision support (DS) part of it we detail in this paper. The task of the agent simulation system is to simulate the remainder of the process based on the current state of the process and to give recommendations for situations where it needs to decide on the resource assignment. Thus, the agent system specifies that in the course of simulation which choice results

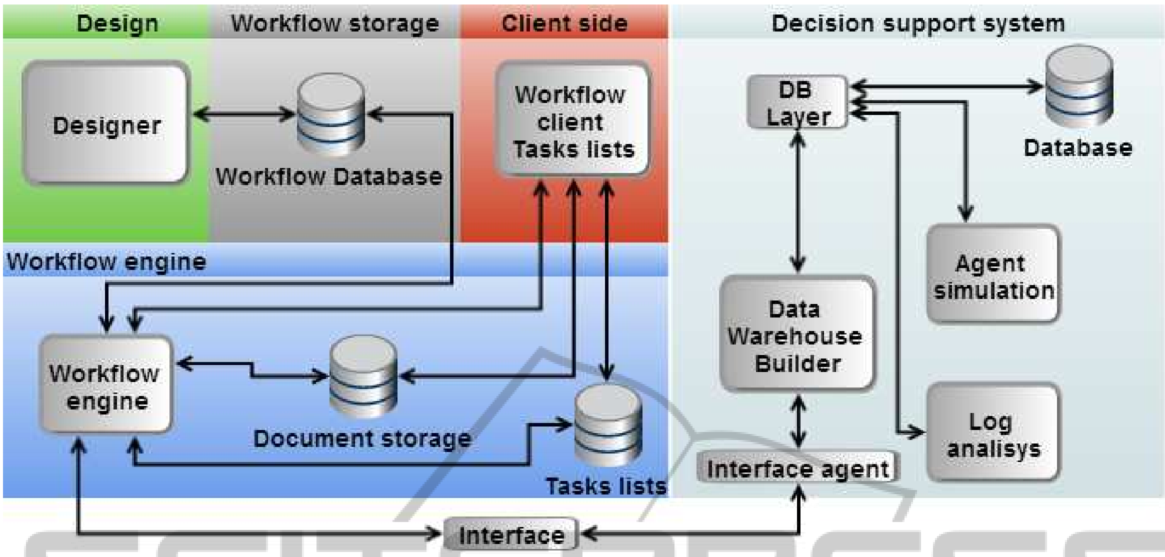

Figure 1: The full system architecture.

the best execution. We can use this information for determining the possibility of the cooperation among agents on behalf of reduce costs and execution time. We would like to use a DS system which gets a workflow from the main system (e.g., in xpdl format) and the DS system needs to analyze this workflow using event logs. Given the deadline of the process, the system examines from a given point whether it can give a new scheduling to the resources for reaching optimal performance.

3 DEVELOPED MODELS

For the above mentioned problem set, we created a method to carry out the scheduling. The work intends to be the basis of several different tasks with different goals (in some cases the total cost of the processes is the critical parameter, while in other cases the operation time should be minimal). In this section we introduce our approach to which some models are attached and these models represent the function to contribute to the decision making for the resource scheduling.

The model and its parameters are designed in such a way, that it should be applied for different kinds of scheduling problems.

For this reason we collected the necessary parameters that make possible the development of a general framework which uses different kinds of resources in different processes and is able to handle different needs.

3.1 Agent Model

This model specifies the characteristics of the agents and is a reference point to the other models. The agent identification is based on a previous conceptualization phase where actors and use cases were modeled.

$R = \{r_1, \dots, r_m\}$, is the set of resources;

A resource can be anything what does an operation on a product type in an operation mode in a given time.

$O(r_i) = \{o_{i1}, \dots, o_{ik}\}$, is the set of the operation modes of resource i;

The same operation of the same resource on the same product type can be carried out in different operation modes, which may differ e.g., in their cost or the time they required.

$A = \{a_1, \dots, a_n, pause(t)\}$, is the set of basic activities, where $pause(t)$ is an empty activity with length of t hour;

Each basic activity is an operation which can be carried out on a product type by a resource in a given operation mode. Each basic activity has a length and a cost which depends on the previously mentioned parameters, too (product type, resource, operation mode).

$ra : R \times A \to \{0, 1\}$, is a function for determining whether a resource is able to perform an activity;

More basic activities could be carried out by the same resource and the same activities could be performed by different resources.

$rap : R \times A \times P \to \{0, 1\}$, is a function for determining whether a special activity of a resource can be applied for a product type;

This function is needed because it may happen, that a special resource while carries out a basic activity is not able to be used for a given product type (e.g., the resource doesn't have the physical dimensions which is needed to put the given product type into it).

$t : R \times A \times P \times O \to \mathbb{N}$, provides the suggested operation time of a resource in a given operation mode performing a given activity on a given product type;

E.g., a test cabin may do the same test in 80 minutes with 70 °C temperature, while the same test takes only 50 minutes on the same product type with the same resource in case of 90 °C. The different temperatures represent different operation modes of the test cabin.

$c : R \times A \times P \times O \times N \to \mathbb{N}$, provides the cost of an activity of a resource in an operation mode on a given product type with a given operation time;

This function works similar to function $t()$.

In the model 4 storages are distinguished: Set of resources, Set of activities, Set of products, Set of operation modes.

3.2 Organization Model

This model represents the humming replace ability of agents among one another.

We used the cooperation to scheduling because we introduced a special vector, i.e., cooperation vector.

$coop(r_i, a_j, p_k, o_{il}) = [h_{i1}, \ldots, h_{im}]$, is the cooperation vector, where m is the number of resources, h_{in} is a natural numbers for all $1 \leq n \leq m$, r_i is the i^{th} resource, a_j is a basic activity, p_k is a product type, o_{il} is an operation mode of resource i, h_{in} denotes how much percent of the productivity of resource i is needed in a given activity in a given operation mode on a given product type to substitute totally resource n supposing unchained operation time.

E.g., $h_{ij} = 30$ means that resource i is able to substitute resource j, moreover, resource i is able to substitute 3 pieces of resource j in the given operation mode in the given activity on the given product type. $h_{ij} = 200$ means that resource i is physically able to substitute resource j, however, two pieces of resource i is needed to substitute one piece of resource j in the given operation mode in the given activity on the given product type. $h_{ij} = 0$ means that resource i is unable to substitute resource j in the given operation mode in the given activity on the given product type.

3.3 Knowledge Model

We introduced some special functions:

$q : R \times A \times P \times O \to [0, 100]$, provides capacity information: how many percent of a resource capacity is occupied by one piece of a given product type in a given operation mode of a resource while performing a given activity;

The occupation information is given in percent of the resource's total capacity because this parameter can be used more universal in this way. E.g., a gas tester during a normal gas test in normal operation related to a special pressure sensor as product type has capacity 20%: it means that the gas tester can handle 5 pieces of the same product type at the same time; however, next to 2 pieces of the special pressure sensor the gas tester could handle one other type of product, too, which needs at most 60% of the tester's capacity.

$e : R \times A \times P \times O \to [0, 1]$, provides the probability of resource failure during performing a given activity on a given product type in a given operation mode;

$s : R \times A \times P \times O \to \{0, 1\}$, results in a binary decision: whether a resource activity in a given operation mode on a given product type can be suspended without restarting it from its beginning;

Assume a basic activity of a resource, which has 5 days operation time. Suppose, that after 3 days from the start, the resource fails. The impact of the failure depends on whether the basic activity of the resource in the given operation mode on the given product type was suspendable. When it was, the continuation of the activity needs only 2 days. If not, it needs another 5 days. This parameter has importance not only in case of device failure, but in normal operation, too, e.g., in case of shared usage of a resource between time-shifted activities.

$rreq : R \times A \times P \times O \times DateTime \to P(R, \mathbb{N})$ (power set on pairs of a resource and a natural number), provides the additional resource need of a resource's given activity on a given product type in a given operation mode in a given hour;

Some resources require other resources, too, while perform an activity. This additional need may depend on the phase of the operation, too. E.g., a temperature shock test requires not only the tester device (the tester has to be used during the whole duration of the test), but a human helper, too. However, the human operator is needed only in the first and the last hours of the 5 days long test.

$bound(r_i, a_j, p_k, o_{il}, c_m) = o_{in}$, provides the impact of circumstances based on the experiences, where r_i is the i^{th} resource, a_j is a basic activity, p_k is a product type, o_{il} is an operation mode of resource i, and c_m is a regular expression on time and basic activity information;

It is written in the form, that the "normal" operation mode can be transformed into a ”modified” operation mode because of the circumstances which depend on the time and the preceding activities. E.g., a human may perform a work slower just after lunch time or after a preceding heavy task. This factor influences the most the human resources. Process mining may serve the necessary information for determining this function based on the experiences from past behaviors.

$map : R \times DateTime \to \mathbb{N}$, provides the information on accessibility of resources: how many resource of a given type is accessible in a given hour;

$F : R \to \mathbb{N}$, provides the expected number of hours how long a given resource is unavailable in case of its failure;

This function may reflect the average time needed to repair a resource type in case of technical failure.

This parameter can be used when schedules are investigated from the viewpoint of failure-awareness.

3.4 Coordination Model

We can represent the communication and cooperation of the software agents operating in the system by this model. In the model three main agents can be seen: AMS, Controllers and Task agents.

The *Agent Management System* (AMS) is an agent, which operates built into the system. The AMS realizes system services, any agents can make use of which. Such services are, e.g., starting a new store, starting a new agent, migration of agents between stores, etc.

Controllers: In this group there are two agents. The first agent is the sentinel agent, the task of which is, the communication with the external system, i.e. taking over of data coming from there or sending data from the decision support system. The second agent will be responsible for the efficient allocation of the tasks – in the arriving process model – among resources.

The resources may apply for each task in definite time interval, then, when the timer expires; the winner agent gets the right to perform this task.

Task agents: these agents represent the resources. The task of these agents is the execution of the activities of the process model.

Each agent is connected to all other agents in the system. The resources can send requests and answers to one another or can receive from another. In the following we will look at the operation of the more important communication channels by some sequence diagrams.

The further coordination model describes the dynamic interactions between software agents by using sequence diagrams to represent the conversations and interventions carried out between them.

3.5 The Process

During the development phase of DS system, it is necessary to simulate the system feasibility before the formal implementation. Because of it, we develop a process model to represent the behaviour of the DS system and then to simulate the model performance.

$Process = \{proc_1, \ldots, proc_j\}$, is the set of processes;

Contains all the processes; the elements can be, e.g., a test sequence of a company or a production process.

$P = \{p_1, \ldots, p_l\}$, is the set of product types;

We need product type because a resource can handle the different product types in different ways (for example needs different working time); however, products belong to the same product type are handled in the same way in a production process.

$t_{start} : Process \to DateTime$, shows the start time of a process;

This parameter describes the time when the first basic activity of a process begins.

$t_{maxend} : Process \to DateTime$, shows the maximum finish time of a process;

This parameter is to describe the latest time when a process has to be finished. It means a constraint in scheduling: basic activities of a process can not be pushed forward in time with arbitrary time units.

$prev : Process \times A \times \mathbb{N}^+ \to \{A, \emptyset\}$, provides the prior basic activity of the n^{th} occurrence of a given basic activity in a given process;

This function is to describe the sequences of basic activities in a process. In our model processes are built up by sequences of basic activities. No other structure of them are allowed.

$next : Process \times A \times \mathbb{N}^+ \to \{A, \emptyset\}$, provides the following basic activity of the n^{th} occurrence of a given basic activity in a given process;

This function is the opposite function of $prev()$.

$maxdelay : Process \times A \times \mathbb{N}^+ \to \mathbb{N}$, provides the maximal duration of time out, which is tolerated by the n^{th} occurrence of a given basic activity in a given process just after its execution;

There may exist neighboring basic activities in a process, which do not tolerate arbitrarily huge time gap between them. E.g., during cooking tea, after boiling the water we may not to wait too much before putting into the water the tea filter (anyway we should repeat the previous process - the boiling).

$dur : Process \times A \times \mathbb{N}^+ \to Q$, provides the time scale, which number the default operation time of the of the n^{th} occurrence of a given basic activity in a given process has to be multiplied with, for getting the real operation time of the activity.

Users of the scheduling software may modify the default schedule of different activities. E.g., a manager - based on the special requirement of a customer - chooses that the temperature shock test should last 7 days instead of the default 5 days. In this case the given test process's chosen temperature shock test has time scale with value $7/5$.

4 CONCLUSION AND FUTURE WORK

In this work we introduced a novel approach, which is based upon some models for resource scheduling of workflow-elements. Resources of the workflow are represented by cooperating and rival agents, whose capabilities are carrying out activities on products. We intended to model these agents and to work out their coordination, management in such a way, that the whole system realizes a well-scheduled workflow which carries out the desired operation in a robust, fault tolerant way making possible the substitution of the system-elements by each other. A system which is created based on the presented models is applicable in wide spectrum of practical use for solving scheduling problems related to business workflows. In current development phase the system is suitable for time-based or cost-based scheduling. The system is able to compare the unscheduled and scheduled workflows to each other and display the result. Moreover, it is well applicable for comparing the results of different scheduler algorithms.

In our future work we would like to create a complete workflow management system which integrates this method to schedule workflows of real applications. The set of system modules will be expanded with a log analysis module to make the scheduling more efficient using historical data from log files. The system is created also for analyzing the efficiency of different log analyzer methods - similar to the case of different scheduling algorithms. Furthermore we plan to add a mixed scheduling factor which includes the time and the cost with any weight at the same time.

ACKNOWLEDGEMENTS

This publication has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TAMOP-4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

REFERENCES

Barba, I. and Del Valle, C. (2010). A job-shop scheduling model of software development planning for constraint-based local search. *International Journal of Software Engineering & Its Applications*, 4(4).

Bhaskar, R., Lee, H. S., Levas, A., Pétrakian, R., Tsai, F., and Tulskie, B. (1994). Analyzing and re-engineering business processes using simulation. In *Simulation Conference Proceedings, 1994. Winter*, pages 1206–1213. IEEE.

Brucker, P. and Brucker, P. (2007). *Scheduling algorithms*, volume 3. Springer.

Dulai, T. and Werner-Stark, A. (2012). Immediate event-aware routing based on cooperative agents. In *Proceeding of Factory Automation*, pages 144–148.

Ghosh, J. (2013). Distributed task scheduling and human resource distribution in industrial service solution production: a simulation application. *International Journal of Production Research*, 51(10):2900–2914.

Merdan, M., Moser, T., Sunindyo, W., Biffl, S., and Vrba, P. (2013). Workflow scheduling using multi-agent systems in a dynamically changing environment. *Journal of Simulation*, 7(3):144–158.

Murthy, S., Akkiraju, R., Rachlin, J., and Wu, F. (1997). Agent-based cooperative scheduling. In *Proceedings of AAAI Workshop on Constraints and Agents*, pages 112–117.

Pinedo, M. (2012). *Scheduling: theory, algorithms, and systems*. Springer.

Sule, D. R. (2008). *Production Planning and Industrial Scheduling: Examples, case studies, and applications*. CRC press.

Tumay, K. (1996). Business process simulation. In *Proceedings of the 28th conference on Winter simulation*, pages 93–98. IEEE Computer Society.

van der Aalst, W. M., Reijers, H. A., Weijters, A. J., van Dongen, B. F., Alves de Medeiros, A., Song, M., and Verbeek, H. (2007). Business process mining: An industrial application. *Information Systems*, 32(5):713–732.

van der Aalst, W. M., van Dongen, B. F., Herbst, J., Maruster, L., Schimm, G., and Weijters, A. (2003). Workflow mining: A survey of issues and approaches. *Data & knowledge engineering*, 47(2):237–267.

van der Aalst, W. M., Weijters, T., and Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *Knowledge and Data Engineering, IEEE Transactions*, 16(9):1128–1142.

Xu, J., Liu, C., Zhao, X., and Yongchareon, S. (2010). Business process scheduling with resource availability constraints. In *On the Move to Meaningful Internet Systems: OTM 2010*, pages 419–427. Springer.