COMPONENT-BASED MODELLING OF ORGANISATIONAL RESOURCES USING COLOURED PETRI NETS

Khodakaram Salimifard
Department of Industrial Management, Persian Gulf University, Booshehr 75168, Iran

Keywords: Workflow, Organisational Modelling, Coloured Petri Nets, Component-based Modelling

Abstract: Collaborative software applications such as workflow management systems require a clear separation between process model and resource model. The process model realises the partial order of business processes while the organisation model provides the structure of the resources to be utilised. In this paper, we propose a CPN-based framework for modelling organisational functioning units. The models are developed independent of the process layer. Maintaining the flexibility and scalability, the model, hence, is capable to be modified without altering the process model.

1 INTRODUCTION

The term "workflow modelling" implies the requirement to represent the processes through which the work is accomplished. This requirement becomes a vital issue in designing and re-designing a workflow system. A workflow model defines *what* has to be done. That is, it represents a functional perspective (Koulopoulos, 1995). On the other hand, the model frames the behavioural and organisational perspectives (Jablonski et al., 1996). Many of the existing approaches to workflow modelling are based on using Petri nets (Ellis et al., 1993) (Ferscha, 1994) (Aalst, 1998) (van He et al., 2000) (Salimifard and Wright, 2001). The standardisation of model elements has captured less attention, although the compositionality of Petri nest has been discussed in (Voorhoeve, 2000). Recently, the requirements for modelling workflow have been investigated (www.minicom,com). The requirements have been classified into modelling patterns representing comprehensive functionality necessary for workflow software. In this paper, we introduce a CPN-based compositional approach for modelling workflow processes and organisational units.

2 MODELLING REQUIREMENTS

The behavioural perspective (Koulopoulos, 1995) of a workflow is represented by *control flows* (Aalst, 1998). A control flow specifies the *execution order* (Jablonski et al., 1996) and dependencies between workflows. It determines all the possible execution orders of embedded workflows. A control flow construct is either an *elementary* or *composite*. A composite control flow construct is a composition of the other control flow constructs. This hierarchical representation allows the workflow designer to design a complex workflow using a set of control flow constructs.

In order to have a standard definition, a formal specification of the semantics of control flow constructs is required. The formal definition makes it possible to validate the proposed constructs and to extend any construct, if it is necessary. In our approach, we define the semantic of control flow constructs in Coloured Petri Nets (CPNs) (Jensen, 1997). Each control flow construct is modelled as a Coloured Petri Net (CPN) model, namely a *workflow component*. A component is constructed according to the following modelling rules:

- The component is framed between an *input place* and an *output place*,
- A place holds the unique *identification code* for the component,
- Apart from the input and output places, a component may have other places which are *local* to the component,
- Components can be linked (chained) using their input and output places.

A component may contain other components. This characteristic allows modelling of a very complex workflow. The modelling approach we present in this paper is a high level representation of

Salimifard K. (2004).
COMPONENT-BASED MODELLING OF ORGANISATIONAL RESOURCES USING COLOURED PETRI NETS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 577-580
DOI: 10.5220/0002652005770580

control flow constructs. We introduce different modelling components and their semantics using CPNs. The CPN models have been designed using Design/CPN software tool (Design/CPN Reference Manual for X-Windows, 1993).

3 ORGANISATION MODELLING

The capability of thorough representation of organisational resource has become an important feature of workflow management systems. While the researchers have mainly focused on structural modelling of organisational resource, the dynamic aspects such as resource efficiency and non-operational behaviour of human resources have captured less attention.

In our approach an organisation model is composed of four parts which are modelled as CPN pages. In general, an employee as a human being has a very complex behaviour. We draw a model representing the abstract behaviour of employees in an organisation. The model is depicted in Figure 1.

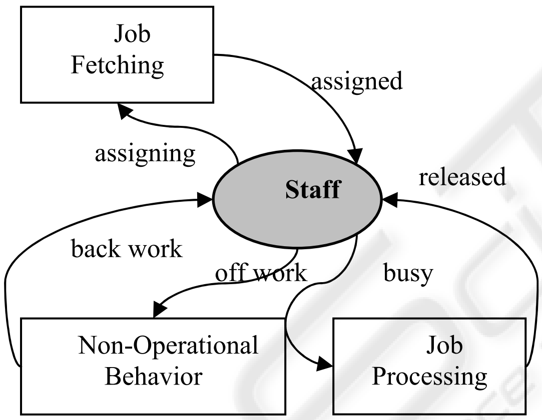

Figure 1: Staffs' abstract behaviour

Any available member of staff may have either of three abstract class of behaviour at a time. *Job fetching* refers to a situation where an employee is dealing with the selecting (assigning) an available work-item. A member of staff may be processing a work-item, which he already has been assigned to. This behaviour is represented by the *Job processing* entity. Behaviours that are not productive have been classified under the term *Non-Operational* behaviour. The term implies that such behaviour is not part of the employee's organisational duties.

3.1 Player Query and Scheduling Work Items

In a workflow system environment, an organisation unit, e.g. Accounting department, processes those workflow activities that require a service from that organisation unit. An organisation unit offers its services to the system through the exchange of *request for service* (RfS). That is, the process level submits a work item to the organisation unit as an RfS. A work item is a representation of a workflow activity modelled as a CPN *binding element*. It carries information about processing requirements such as department, role needed, and the estimated processing time.

The generic model of the query part of the organisation model is depicted in Fig. 2. The figure is modified for the department with the code 1, as it is appeared as `D1` in Fig. 2. The place `D1ID` holds the department identification code as the initial marking.

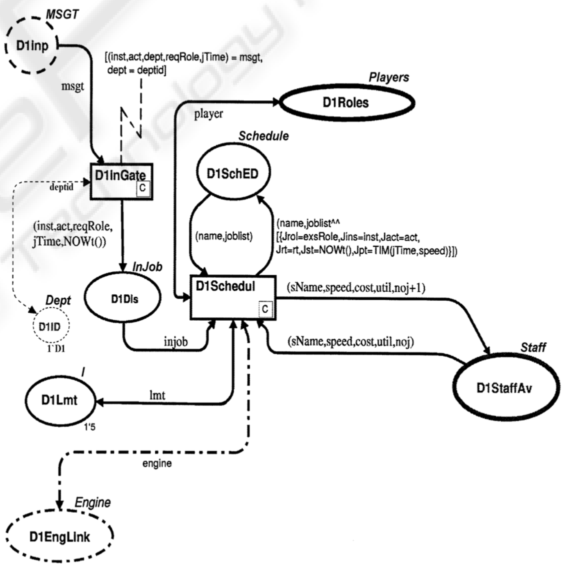

Figure 2: CPN model of query part

Upon receiving an a *msgt* token through *D1inp*, the transition *D1inGate* is enabled. Occurrence of this transition consumes a token from *D1inp* and creates a new token in *D1Dis*. The created token is of color-The function *NOWt*() determines the arrival time of the work item.

The place *D1StaffAv* holds all the staffs available at the department. Based on the organisation policy, the maximum number of jobs each staff is assigned to is determined as the initial marking for *D1Lmt*. The initial marking of *D1StaffAv* determines employees who are initially available. The role classification for the department is defined as the marking of *D1Roles*.

3.2 Work Item Assignment

All the work items have been accepted to be processed by the department are kept in the department in-tray, accessible by employees. A work item in the in-tray can be fetched via the occurrence of the transition *D1Schedul*. It has a four-facet and complex behaviour described as follows:

- *Role matching*: upon the availability of a token in `D1Dis`, it checks whether any role defined in `D1Roles` matches with the *reqRole* parameter of the work item.
- *Staff availability* if a player is found, the availability of the player is checked.
- *Quota checking*: if the result of the previous check is positive, it checks the number of jobs currently the employee is assigned to, i.e. *noj*, against the maximum number of jobs an employee can be assigned to, i.e. [$noj \leq lmt$].
- *Job list updating*: if the results of both previous tests are positive, the job list associated with the employee has to be updated.

The newly work item selected will be appended at the end of the employee's job list. The actual processing time of the work item is determined using the function *TIM*().

3.3 Processing Work Items

The processing of a work item is diagrammatically represented in Fig. 3. The start of the execution of a work item is modelled using the transition *D1JobStart*. Its occurrence is explicitly conditioned by the exact matching of the player whose job list is available in *D1SchEd* and the employee available in *D1StaffAv*.

A job list is in fact a simple queue from which the owner selects the work items to process. The selection can be done using different queue disciplines (Kleinrock, 1975). In our approach, we assume that the organisation committed to use the FIFO discipline. The output arc between *D1JobStart* and *D1JobinPro* (see Fig. 3) inscribed with the work item and its "end_of_process" time, i.e. *#Jpt job*. This arc also carries the information about the employee who is processing the work item.

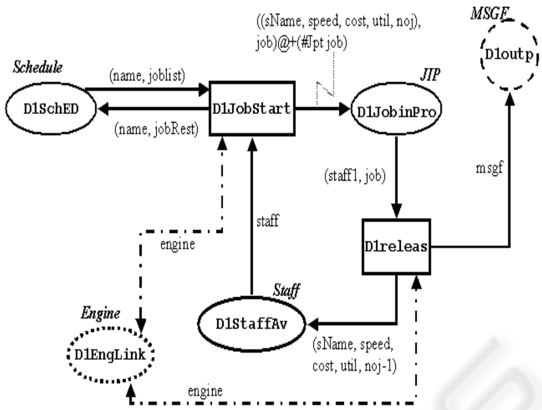

Figure 3: The job execution model

3.4 Releasing Resources

The sojourn time of the token in *D1JobinPro* indicates the processing time of the corresponding work item. The transition *D1releas* occurs and an *msgf* token is sent to *D1outp* indicating that the work item has been successfully processed. Additionally, the employee who was performing the job is released and the corresponding token is sent back to *D1StaffAv*.

4 MODELLING NON-PRODUCTIVE ASPECTS

The need to represent the behaviour of individual employee as well as organisations has been expanding as a result of the increasing use of simulation and systems analysis. The generic model of non-operational behaviour of human resources is depicted in Fig. 4. Transitions *D1off* and *D1back* model the off-work and back-to-work behaviour, respectively. The time duration between two successive off-works is a stochastic random variable. The place *D1NoE* is a counter keeps the current number of available employees.

The occurrence of *D1off* has three effects on the rest of the model. Firstly, it takes a staff token from *D1StaffAv* and decreases the counter value by one. Secondly, it determines the next time that another off-work happens. Thirdly, it specifies the duration of the off-work period using the function *offDur* on the output arc to *D1Ret*. For the employee who is off-work his corresponding token is kept in *D1Away* while *D1Ret* keeps the corresponding off-work period. The occurrence of *D1back* takes the staff token from *D1Ret* and sends the token back to *D1StaffAv*.

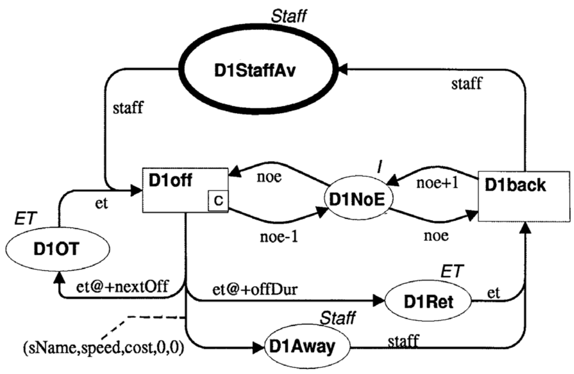

Figure 4: Model of non-operational behaviour

5 SUMMARY

In this paper, we have presented a component-based resource model. This method incorporates both the static and the dynamic issues of human resources. Role-based classification as it is required by most of leading workflow management systems, as well as the individual efficiency of employees has been taken into account. The behaviour of human resources has been classified into fetching work items, processing work items, and non-operational behaviour. Each class of behaviour has been used to derive an appropriate CPN representation.

Using the CPN as the modelling schema, allows the separation between the process model and the organisation model. This separation gives much flexibility in modelling and modification. Additionally, CPN-based modelling characterises the proposed method with powerful expressiveness and capable to be linked to the process model.

In this paper we mainly focused on the designing a workflow process regardless of the performance of the model. To finalise a workflow process model, the performance of the model has to be studied. The performance of a workflow model depends on the WfMS and the management of resources. Therefore, it is required to integrate a workflow process model into organisational model, in order to study the performance of the workflow process.

REFERENCES

T.M. Koulopoulos, The Workflow Imperative, Van Nostrand Reinhold, New York, USA, 1995.

S. Jablonski, C. Bussler, Workflow Management: Modeling Concepts, Architecture, and Implementation, International Thompson Computer Press, 1996.

C.A. Ellis, G.J. Nutt, Modelling and Enactment of Workflow Systems, 14th International Conference on Application and Theory of Petri Nets, Chicago, Illinois, USA, 1993, pp 1-16.

A. Ferscha, Qualitative and Quantitative Analysis of Business Workflows Using Generalized Stochastic Petri Nets, in G. Chroust and A. Benezur (Eds), Proceedings of CON'94: Workflow Management, Paradigms and Products, pages 222-234, Oldenburg Verlag, 1994.

W. M. P. van der Aalst, The Application of Petri Nets to Workflow Management, The Journal of Circuits, Systems and Computers, 8 (1) 1998, 21-66.

K.M. van Hee, H.A. Reijers, Using Formal Analysis Techniques in Business Process Redesign, in van der Aalst, Oberweis (Eds.), Business Process Management, Lecture Notes in Computer Science 1806, Springer, 2000, pp 142-160.

K. Salimifard, M. Wright, MORaD-net: A Visual Modelling Language for Business Processes, in N. Krivulin (editor) Proceedings of International Workshop "New Models of Business: Managerial Aspects and Enabling Technology", St. Petersburg, Russia, June 28-29 2001, pp 213-222.

M. Voorhoeve, Compositional Modelling and Verification of Workflow Processes, in van der Aalst, Oberweis (Eds.), Business Process Management, Lecture Notes in Computer Science 1806, Springer, 2000, pp184-200.

Workflow Template Site at http://www.minicom.com

K. Jensen, Coloured Petri Nets, Basic Concepts, Analysis Methods and Practical Use, Vol. 1, Springer, 1997.

Design/CPN Reference Manual for X-Windows, version 2.0, Meta Software Corporation, 1993.

L. Kleinrock, Queuing Systems, Vol. 1: Theory, John Whily & Sons, 1975.