

Copyright © 2005 IEEE

Reprinted from:

2005 3rd IEEE International Conference on Industrial Informatics
(INDIN) Perth, Australia 10-12 August 2005

IEEE Catalog Number ISBN 05EX1057
ISBN 0-7803-9094-6

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Curtin University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Coloured Petrinet for Flexible Business Workflow Modelling

Leo Pudhota & Elizabeth Chang

School of Information Systems,

Curtin University of Technology, Perth, Australia 6845, Pudhotal@cbs.curtin.edu.au, Change@cbs.curtin.edu.au

Abstract --Petrinet diagrams are a recognized form of modelling real time systems and have been included in the list of UML techniques for modelling the dynamic aspects of Object Oriented Systems. In this paper, we explain how workflow techniques can be used for developing models for larger independent systems. These chunks or independent systems which are part of the large systems are expressed through the use of worklets. This can then be used as the basis of modelling systems where timeliness is of importance. However, this requires extending workflow ideas to incorporate flexibility, handling of exceptions and adaptability. Extensions of the petrinet diagrams are then proposed for expressing the models obtained from such workflow techniques.

Keywords: Dynamic workflow modelling, extended petrinet diagram, workflow, worklet, Objects

I. INTRODUCTION

Workflow management is a fast evolving technology being exploited by business in a variety of industries. [1]. Its primary characteristic is the automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with IT application and tools. Although its most prevalent use is within the office environment in staff intensive operations such as banking, legal and general administration, it is also applicable to some classes of industrial and manufacturing applications.

Workflow management systems allow one to design, sequence, and monitor extended business or industrial processes over a finite time horizon to carry out several interlinked activities. It includes process modelling, executing, re-engineering, monitoring and automation of workflow. Unified Modelling Language (UML) is a language for modelling. Its primary function is to model a software application by utilizing object oriented methodology. It has several diagrams for modelling; One of the diagrams is the Petrinet diagram. It is used to model the dynamic behaviour of a software application, but it also can be used to model a workflow of a non-software application.

II ISSUES RELATED TO ADAPTIVE WORKFLOW

There are some important issues in adaptive workflow that have been identified by Aalst in [2]. These issues are correctness, dynamic change and management info. Some

researches also mentioned another issue, which is flexibility [11].

Flexibility: There are many interpretations of flexibility. Our interpretation of flexibility in this paper is that we focus on flexibility of the workflow management applications in contrast to flexibility of the workflow management system itself.

Correctness: 1. Syntactic correctness.

Which refers to the minimal requirements any workflow should satisfy. Relevant to both the structure of the workflow and the dynamic behaviour, e.g., potential deadlocks and livelocks.

2. Semantic correctness: Which deals with similarities between the capabilities of the old workflow and the capabilities of new workflow.

Dynamic Change: Dynamic means that the changes are made while the system is executing.

III CONTENTS OF A WORKFLOW MODEL

In general a workflow is considered as a collection/ combination of states and transitions that make up a process. Each workflow consists of configurable states and transitions that must be followed from the time an issue or feature is opened to the time it is closed. Each sequence of activities or transactions has to be executed in accordance with the business process of the organization at hand [8].

IV WORKFLOW MODELLING LANGUAGE

The use of a workflow language for representation, analysis and design has led naturally to the need for computer support. A modelling system provides facilities for creating and browsing a representation model, for applying various algorithms to an analysis model and for collaborative interaction and information archiving for design models.

V CHOICE OF METHODOLOGY

It seems natural to work towards object-oriented technologies and methodologies when we discuss about flexibility and adaptability. The fact is that object orientation is very good in bringing solutions to the problems of expandability, maintainability and specialization of software applications. The intrinsic objects characteristics; especially inheritance, encapsulation and polymorphism make objects particularly attractive for finding solutions to the flexibility problems of workflow systems.

Some of the advantages that can be achieved from UML for the modelling of workflows are:

UML is an object modelling language; therefore models built using this notation are easily capable of integrating changes and are easily implementable.

But for an event to be able to occur, a transition must have sufficient tokens on its input places, and these tokens must have token values that match the arc expressions.

The rules for executing the system are:

- a transition is "active" when each of its input places contains a token
- each active transition in the diagram is "fired" by removing 1 token from each input place and generating one in each output place

A Change of State ...

is denoted by a movement of *token(s)* (black dots) from place(s) to place(s); and is caused by the *firing* of a transition.

The firing represents an occurrence of the event or an action taken.

The firing is subject to the input conditions, denoted by token availability. A transition is *firable* or *enabled* when there are sufficient tokens in its input places.

After firing, tokens will be transferred from the input places (old state) to the output places, denoting the new state. Petrinet diagrams allow representation of all the elements included in a workflow model: roles, transitions.

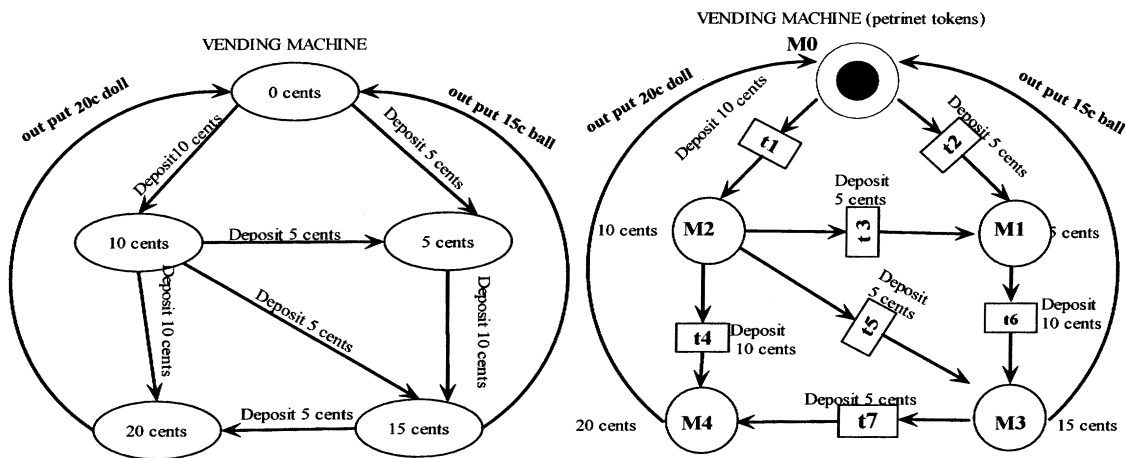


Figure 1. Example of Petrinet diagram



Figure 1.1 A firing or occurrence sequence

events, control and information flows, conditions and relevant data. It also has a capability to model concurrent events by making use of a synchronization bar to specify the forking and joining of parallel flows of control. Figure 1 shows an example of vending machine on the left hand side and the representation of tokens in petrinet diagram on the right hand side; we assume the machine dispenses two kinds of items for kids ball and doll – 20c and 15c, only two types of coins can be used – 10c coins and 5c coins, the machine does not return any change. It shows various inputs which are possible for an out put to be either a ball or a doll. Scenario 1: Deposit 5c, deposit 5c, deposit 5c, deposit 5c, returns 20c doll.

Scenario 2: Deposit 10c, deposit 5c, returns 15c ball.

Scenario 3: Deposit 5c, deposit 10c, deposit 5c, returns 20c doll.

A *marking* is a state ... Initial marking: M0

M0 = (1,0,0,0,0)

M1 = (0,1,0,0,0)

M2 = (0,0,1,0,0)

M3 = (0,0,0,1,0)

M4 = (0,0,0,0,1)

Petrinet have some behavioural Properties like

1. Reachability

- “Can we reach one particular state from another?”
- “M2 is *reachable* from M1 and M4 is *reachable* from M0.”
- In fact, in the vending machine example, all markings are reachable from every marking.

2. Boundedness

• “Will a storage place overflow?”

- A Petri net is said to be *k-bounded* or simply *bounded* if the number of tokens in each place does not exceed a finite number *k* for any marking reachable from M0.
- The Petri net for vending machine is 1-bounded.
- A 1-bounded Petri net is also *safe*.

3. Liveness

- “Will the system die in a particular state?”
- A Petri net with initial marking M_0 is *live* if, no matter what marking has been reached from M_0 , it is possible to ultimately fire *any* transition by progressing through some further firing sequence.
- A live Petri net guarantees *deadlock-free* operation, no matter what firing sequence is chosen.
- The vending machine is live and the producer-consumer system is also live.
- A transition is *dead* if it can never be fired in any firing sequence.

The advantages of Petri-Nets are they have formal analysis to analyze the correctness of the workflow. By using the formal analysis, we could identify if a workflow model contains any deadlocks or live-locks.

Figure 4 shows the Petri-Net version of the CS-Dept worklet. The bubbles represent actions and the rectangles represent the transition. The black circle inside the bubble represents token. A Token moves from one bubble to other bubbles and a bubble can have many tokens. One could say that tokens are similar to conditions. One of the most used techniques for modelling and analyzing workflows is *Petri Nets [PN]*. By mapping tasks to transitions, resources to places, and cases to tokens, the three dimensions of the workflow space can be faithfully represented using this formalism. If colour is added to a classical Petri net, workflow attributes can also be included in the model. I.e., in a colour-extended Petri net, conditions can be set for the values of the tokens. Time is another extension of the classical PN. By adding time, a complete model of a real workflow system can be delivered.

VI EXTENDED PETRINET DIAGRAM FOR WORKFLOW MODELLING

The petrinet Diagram is one of the five diagrams in the UML for modelling the dynamic aspects of systems, they show flow of control from one transition to another transition. The transition is triggered by one or more events and the transition may result in one or more events that may trigger other transitions or processes. Events, which are message flows in UML, start from the token being at the beginning and end with the token at the finish state having transitions in between connected by events. Petrinet diagram can be thought of as defining the actions in that process, which continues until everything that needs to be done, is done. Petrinet diagrams represent the decisions, iterations and parallel/random behaviour of the processing. A common use of the petrinet diagram is to model the dynamic aspects of a system. When we model the details dynamic aspects of a workflow computation, we focus on transitions as viewed by tokens that capture flow of control. UML also has a very useful technique in modelling a workflow called swimlane. It is used in

petrinet diagram too. A swimlane is a group. A group here represents the business organizations, which are responsible for activities. Therefore, by having a swimlane, we partition the transition states on a petrinet diagram into groups. This technique gives us benefits in modelling and reading a workflow.

VI EXTENDED PETRINET DIAGRAM

By adopting the idea from the petrinet diagram of UML, we try to extend it so that it will provide us benefits to solve our initial problem.

In this paper, we suggest that a workflow model should contain these following objects:

Data: represents the data that will be used or produced in an action (e.g. e-mail, order form)

Conditions: represents the conditions associated with an action. Conditions can be divided into two kinds, namely *pre-conditions* and *post-conditions*. Pre-conditions are conditions that need to be satisfied so that an action can be enacted. Post-conditions are conditions that need to be satisfied on completion of an action.

Action: represents a piece of work, it can be manual action or automatic action.

Worklet: represents an activity/business process that is carried out by an organizational unit (e.g. Department)

VI.I WORKLET

Worklet is a small workflow, which is carried out by organizational unit. It contains one or more actions. One could think of a worklet as simply a sub-workflow, however, the main difference is a worklet is a sub-workflow that is carried out only by one organizational unit. Each worklet has a unique *WorkletID*

The idea of worklet is to reduce the complexity of business process models. It provides the modeller less difficulty in both modelling and maintaining the models. Another objective of the worklet is to achieve independency of enactment of business processes across a company. Let us consider a company, which produces a product where more than one department is involved in production. We believe that the workflow models will be very large. Moreover, if the production time is very long then departments, which have done their jobs, have to wait till the process finishes. This will be very time consuming and inefficient.

VI.II ACTION

An action is a single task which done by an actor or actors. Actions can be categorized into two types in terms how they are done, *manual-action* and *automatic-action*. A Manual-action is an action that is manually done by actors (e.g. filling an order). Meanwhile, an automatic-action is an action that is automatically done by the workflow management system (e.g. sending an e-mail).

Ideally, an action cannot be done in arbitrary way. There should be conditions that need to be satisfied first to enact the action. These conditions are called pre-conditions. These pre-conditions could be the business rules of the business process. Post-conditions are conditions that need to be satisfied so that a task can be said to be 'complete' or 'done'. An action also can have input and output, in the form of data, documents or other objects (e.g. order form). As for a worklet, human intervention is needed for manual-action. There is no way for the system to know if a manual-action is done or not. Users must let the system know if they have finished doing their jobs (e.g. by pressing "done" button).

VI.III NOTATIONS FOR EXTEND PETRINET DIAGRAM

Since there have been several new objects that we just introduced in the previous section, therefore we need to propose some extensions to petrinet diagram and new notation for the extended petrinet diagram.

An action can be viewed as a big object that contains other objects. It is represented as a rectangle shape, but it has inwards and outbound arrows above and below it. Figure 2.1 shows the notation. Note that, if the value of actor is none, it means that the action is an automatic one; otherwise it is a manual one.

To represent the interface of one action in a given worklet to another action in another worklet we use the pentagon symbol (Figure 2.2)

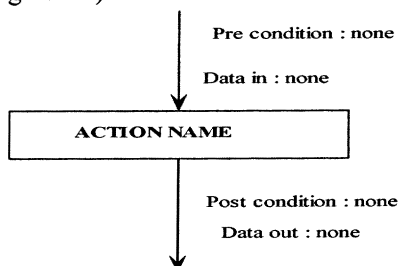


Figure 2.1: An action notation

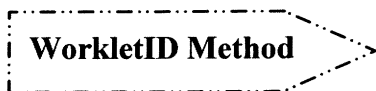


Figure 2.2: An interface notation

VII APPLICATION OF THE PROPOSED METHODOLOGY TO A REAL WORLD EXAMPLE

In this section we will apply our proposed methodology to a real world business process. This business process example is taken from a warehouse and logistic company, in the real world. Logistic Management, in its widest definition, is concerned with the strategy and management of the movement and storage of materials and products from suppliers, through the firm's distribution systems to retail outlets and customer. The scope of logistic

management for the physical movement of goods starts with the sources of supply and ends at the point of consumption [10]. The Logistic Management is part of Supply Chain management.

The company modelled is one of the largest Warehouse and Logistics Company in the Asia Pacific region. Its main service is to provide space for customers who want to store their goods in a warehouse and shifts their goods from an origin location to destination location. We will use the acronym **LC** to indicate the **Logistic Company**. The company LC provides detailed logistic services for its customer to move their goods from one place to another place. In this project, we only concentrate on logistic part. LC has many types of logistic orders and each order has its own flow of work. Some of these orders are:

Import: means that customer wants LC to pick up goods from one place and bring it to LC's warehouse. These goods are coming from outside Australia and the goods have to be declared/checked in Custom Office before entering Australia. So it is usually picked from either a seaport or an airport.

Export: means that customer wants LC to deliver customer's goods from LC's warehouse to either seaports or airports in Australia. These goods also need to be declared/checked before leaving Australia.

Local Delivery: means that the customer wants LC to deliver customer's good from one place to another place within Australia.

All these orders are involve other types of work such as *Devanning* and *Upvanning*. Devanning is the process where goods are moved from trucks and put in the warehouse. However, the process is not this simple. Some good-checking procedures have to be done to put goods in right warehouse (e.g. frozen fish has to be put in a cold-warehouse) or to find out if there are any discrepancies between the number of goods in the order and the actual number of goods that are coming to the warehouse. Upvanning is simply the opposite of Devanning. Figure 3.1 shows the basic services, which LC provides to its customer

VII.I PROBLEM DESCRIPTION

LC consists of many departments. There are five departments: Warehouse Department, Logistic Department, Account Department, Customer Service Department and Transport Department. Each department has its own responsibility; however they are connected to each other. The Warehouse Department now already has its own system, so does the Accounts Department. Currently, the workflow of the logistic management is not integrated into the existing Warehouse Management System (WMS) or the accounting system. Co-ordination and delivery of goods are initiated by clients using traditional transmission of requests either through telephone or fax. The system uses spreadsheets to manage and update the delivery and transportation charges. The

information is fed manually to the accounting department for necessary updating of the account receivables [9]. All these works are done manually, time consuming and prone to error. The complexity of works are become bigger and bigger when the customers' order are increasing [9].

VII.II SOLUTION

To help LC, we try to build a new LC business process model by using our extended petrinet diagram. Our goal is to manage the complexity of the business process model and also to build a flexible model so it can be changed easily as the business processes are changing. Furthermore, the model can be used as guidance for developing LC's logistic management system. In this paper, we only want to demonstrate the Local Delivery order type. The high-level workflow representation for Local Delivery is shown in figure 3. From this high-level representation, we develop a more detailed model using our methodology. We decided to have five worklets, Customer Service (CS)-Dept worklet, Logistics(LOG)-Dept worklet, Warehouse(WH)-Dept worklet, Transportation(TRN)-Dept worklet and Accounting (ACC)-Dept worklet. All the worklets communicate through e-mails. We assume that customers place order on LC's web site (client side), and once they finish filling all information, it will be sent to LC as e-mail, called e-Order. LC staff also have their own web site (server-side) to process all the e-Orders.

The main duty of the CS-Dept in LC is to take care of all inquires from LC's customer. The CS-Dept also can place orders on behalf of customers. This is because the customers may not have a connection to the Internet so they cannot fill in the order from the LC client web site.

Another duty of the CS-Dept is to check and validate all the e-Orders. If there are problems on the e-Order then the CS-Dept will report the problems to the customer and the order will be suspended until all the problems are fixed.

After checking and validating the e-Orders, the CS-Dept will make a copy of the e-Order and forward it to either the Logistic Department or Warehouse Department through e-mail. These two latter tasks are done automatically by the system. After sending the copy of e-Order, the CS-Dept worklet then reaches the finish state. The entry condition of the CS-Dept worklet is an e-Order received from a customer by the system. For convenience the WorkletID

for CS-Dept is 0001. The Logistic Department has many duties here. The first one is to decide if LC wants to accept or reject the order. An order can be rejected for several reasons. For example, Customer A wants to export 20 tons of chicken wings to Customer B, however, when the Logistic Department checks the warehouse database, Customer A only has 10 tons of chicken wings. Therefore, the e-Order has to be rejected. The Logistic Department then sends a rejection letter to the customer along with the reason and the worklet reaches the finish state. If the customer is on the internet this notification would be done by e-mail. If the e-Order is accepted, the Logistic Department will send an acceptance letter and the next task is to book trucks for delivery. The Logistic Department sends all the information about the delivery, date of delivery, type of truck, etc. From here, we are waiting for confirmation from the Transportation Department. Once the confirmation is received, the Logistic Department then reports to the customer that the order is confirmed.

Next, The Logistic Department will notify the Warehouse Department to prepare the manpower and the goods for the upvaning process. After that, it will fill in a charges form (from staff web site) and send it to Accounts Department. The worklet then reaches the finish state. We assign the WorkletID for the Logistic Department to 0002. Task also acts as an Or-Split, a decision task, which after processing can control the direction of flow. At inventory update accounts dept transition t3 occurs when precondition and post condition of receiving voucher and the bill is sent to customer services dept are satisfied. Parallel transition for operation dept token are fired when pre and post conditions of receiving a voucher and delivery schedule sent to transport dept are met. The next transition is t4 which is local pickup, then the transition stage is notification and delivery acknowledgement is sent by the customer dept.

VIII CONCLUSION

In this paper, we have shown how to model a business process by using our extended petrinet diagram. The main concepts of our diagram are worklet, interface and exception handler. Worklet and Interface have addressed the problem of the complexity of a model. They also have addressed the problem for supporting inter-organizational companies.

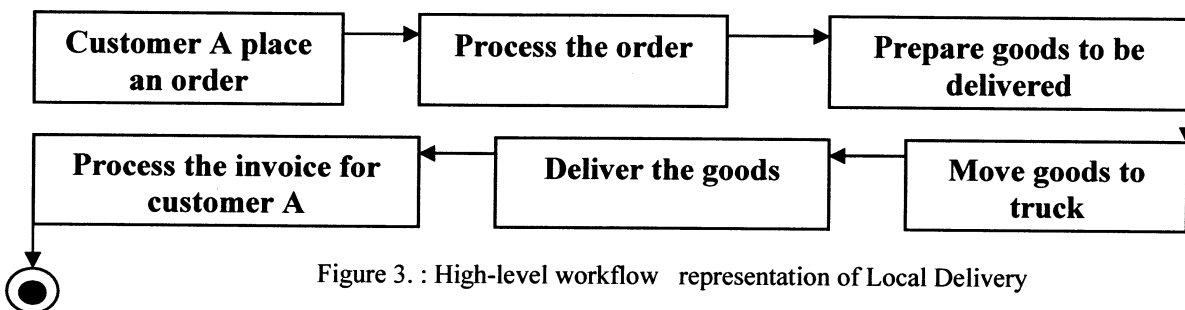


Figure 3. : High-level workflow representation of Local Delivery

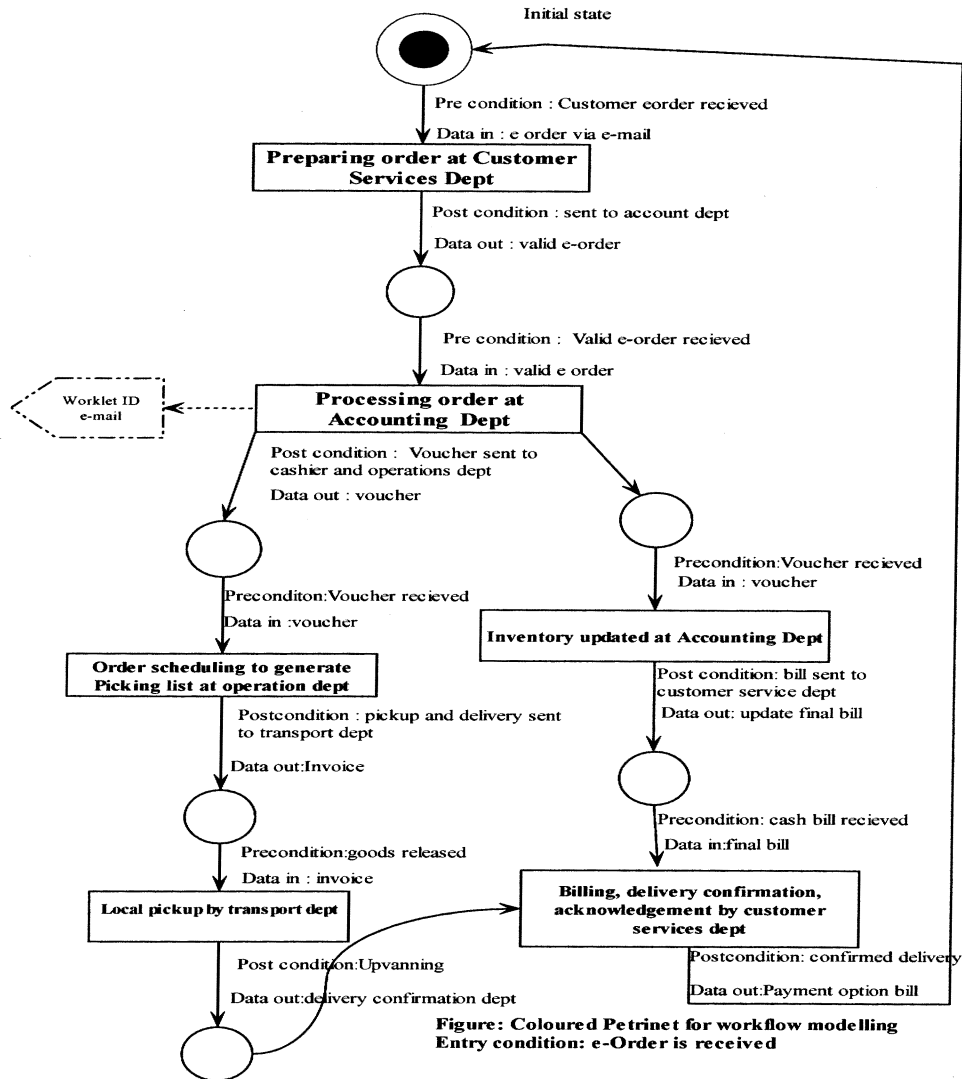


Figure 4. Colour Petri net for workflow modelling

Meanwhile, the exception handler has addressed the problem of coping with exceptions, which could happen in a business process. We also presented a method of “Chunking” large complex systems using the concept of a worklet and workflow. We then introduced extensions to the UML based petri net diagrams to allow modelling of these flexible adaptive workflows. This is illustrated using a real life example from a logistics system.

IX REFERENCES

[1] Georgakopoulos D.; Hornik, M.; Sheth, A.: An Overview of Workflow: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Database*, No. 3, 1995
 [2] Aalst, W.M.P. van der; Basten T.; Verbeek, H.M.W.; Verkoulen, P.A.C.; Voorhoeve, M.: Adaptive Workflow On the Interplay Between Flexibility and Support. *Proceedings of the first International Conference on Enterprise Information Systems*, Vol 2, pages 353-360, Setubal, Portugal, March 1999.
 [3] Kammer, P.J.; Bolcer, .B.; Bergman, M.: *Techniques fro Supporting Dynamic and Adaptive Workflow*.

[4] Horn, S.; Jablonski, S.: An Approach to Dynamic Instance Adaption in Workflow Management Applications, *Proceedings of the CSCW-98 Workshop Towards Adaptive Workflow Systems*, Seattle, USA, November 14, 1998
 [5] Nutt G.: The evolutions Toward Flexible Workflow Systems, *Distributed Systems Engineering*, Vol. 3, No. 4, pp. 276-294, December 1996.
 [6] Ortner, W., Starty C.: Virtualization Of Organizations: Consequences for Workflow Modeling, *Proceedings of the 32nd Hawaii International Conference on System Sciences*, 1999.
 [7] Starty, C.: Integrating Workflow Representation into User Interface Design Representation, *Software – Concepts and Tools*, Springer, 1996-17.
 [8] IBM: Flowmark, Modeling Workflow, Release 1.1, SH19-8175-01, 1994.
 [9] Chang, E: Requirement Specification of Logistic Manager for Software Engineering Project, Department of Computer Science and Software Engineering, The University of Newcastle, 2000.
 [10] Haake, J.M., Wang, W.: “Flexible Support for Business Processes: Extending Cooperative Hypermedia with Process Support”.
 [11] Guntama E., et al : Extension of Activity Diagrams for Flexible Business Workflow Modelling , published in International Journal on Computer Science, System and Engineering, 2003 UK.