Workflow Models Enhanced with Process Algebra Verification for Industrial Business Processes

IOAN SALOMIE, TUDOR CIOARA, IONUT ANGHEL,
MIHAELA DINSOREANU, TUDOR IOAN SALOMIE
Department of Computer Science
Technical University of Cluj-Napoca
15 C.Daicoviciu street, Cluj-Napoca, 400020
ROMANIA
http://dsrl.coned.utcluj.ro

Abstract: This paper addresses the problems of modeling and verification of complex industrial production lines. The paper proposes a methodology for building business models, organized on layers of increasing complexity, from production line elementary machines and sensors to complex business workflows. The resulted models could be translated and executed by any workflow execution engine. For workflow verification purposes, the model is represented in Process Algebra which allows the verification of the correctness of the business process model and the identification of logical faults since the model design phase. The proposed methodology was used for modeling and verification of a sausage processing line developed in the context of Food Trace [3] research project.

Key-Words: Workflow, Orchestration, Business process, Process Algebra, Machine modeling, Web service

1 Introduction

Business processes recently became the fundamental element of large scale industrial systems, therefore the need of modeling and simulation became a fundamental problem. In the internet distributed computing, business processes are represented as workflows and should be able to collaborate and to be executed. Currently, a number of methodologies and specific languages for defining process interactions and collaborations were developed. Business processes (BPs) should be defined according to business domain rules and can be classified in internal BPs and external BPs. Internal BPs are modeling complex company-specific processes such as industrial workflows. External BP usually implies an active collaboration of a set of business partners based on specific business rules. Invoking the direct execution of internal or external BPs, before testing them, may cause errors that could lead to improper operation of the industrial system.

Activities like simulation and online monitoring play a very important role in every organizations business domain. The use of process simulation leads to detecting errors in the process design such as structural errors due to improper workflow and uncertainty errors deriving from business process representation. BP reengineering can be used to remodel and correct the workflow faulty processes detected during the simulation process.

For process simulation purposes, a model that reproduces the real situation has to be created. Relevant simulation results are obtained only if the model accurately reproduces the original physical system. Due to their level of complexity, modeling real systems is not always a simple task, being difficult to be described with precise deterministic or mathematic models.

Modeling of industrial business processes involving physical machines is targeting a better integration of the business processes into workflows of processes on one hand and workflow execution in a simulated manner on the other hand.

In the context of business process modeling and execution, many approaches have been proposed. In [1], the authors propose the SQMA (Situation-based Qualitative Modeling and Analysis) model for representing and simulating industrial systems using Rough Set Intervals. The proposed model uses interval-based representation for qualitative models for implementing the behavior of real systems. The SQMA model hierarchically structures the whole system and decomposes system's levels into components. After that, component variables are modeled using intervals and characteristic values represented as a one-value interval. Physical rules that are used for the model verification are formulated using interval arithmetic to complete the description of each component. Using Rough Intervals and physical rules, a transition matrix between components is constructed and used in simulation. The main disadvantages of this approach consist of inaccurate representation of machines business logic and difficulties in model management. Another disadvantage is that using Rough Set Intervals it is difficult to model complex business scenarios involving more cooperating machines.

Another approach is to use formal description techniques for specifying business processes, especially for those including concurrent and communicating components represented as web services. WSs and their interaction are best described using process description languages like Process Algebra (PA) [2]. Being simple, abstract and formally defined, PA makes it easier to formally specify the message exchanges between WSs, and to reason on the specified business processes during the design stage. The main advantage of using PA for the description and modeling of simple and complex business processes is that it allows the verification of the correctness of the obtained business process model on one hand and the identification of the model logical faults on the other hand.

Our approach on modeling and simulation of business processes is presented in the context of the FOOD-TRACE research project [3]. The FOOD-TRACE project aims to study and design an integrated IT system for food industry processing organizations, in response to the EU requirements regarding food traceability and quality assurance. The system models the production lines using business processes.

The objective of this paper is to define a method for capturing and representing business models involving physical machines targeting their workflow integration and simulation. This objective was achieved by: (i) defining a methodology for the construction of workflow models that comply with specific business rules and (ii) verifying model consistency using the Process Algebra formalism

The rest of the paper is organized as follows. In section 2, we present the design and execution requirements of workflow models. In section 3, the methodology for workflow-based modeling and verification of industrial processes is presented. Process Algebra is used for verifying model correctness. In section 4, a business scenario for food processing and traceability is modeled using the methodology presented in section 3. Section 5 gives conclusions and promising future work.

2 Workflow model - design and execution requirements

One of the best ways to present high-level business collaborations among different heterogeneous and autonomous business processes is by using workflows. Mapping and modeling real processes onto workflows is an open research problem. Usually, this mapping is achieved in two steps:

Step1 - The real processes are divided into simple atomic processes;

Step 2 - The atomic processes are represented as web services interconnected by a workflow model.

In the context of business process modeling, three abstraction levels can be identified: business level, workflow level and machine level.

Business level contains organization’s entities that compete or interact to achieve their goals. Every organization has its own culture that is formulated through policies or rules.

Workflow level defines the set of company specific workflows. The life cycle of a workflow is governed by the life cycle of the company documented procedures related to policies, recipes, etc. which have been used to generate the workflow. For example, in the case of a meet processing company, the sausage production workflow is determined by the product recipe and company specific quality and control policies. When the recipe and company policies change, the sausage production workflow should be modified accordingly.

Machine level represents physical machines, part of the production lines, on which simple workflow orchestrated services are mapped on.

The next section focuses on workflow level by proposing a layered architecture for modeling industrial business processes. When representing workflows, the main idea is to move business process modeling closer to the user knowledge. Currently, two approaches are used for describing business processes and their internal collaboration and execution. The first one uses a visual modeling language that generates an intermediary representation (for example BPMN [4]) which is then converted into an executable language such as BPEL [5]. The second approach describes the processes directly in BPEL.

We have identified the following requirements that should be addressed during workflow model design:

1. The need to abstract business processes by eliminating the irrelevant details for the workflow model.
2. The need to represent real processes into workflow activities including traceability features. The model should allow both upstream and downstream traceability. Upstream traceability starts from row materials and concludes to the final product. Downstream traceability takes the product and decomposes it into sub-products and traces them down until the row materials.
3. The need to associate web services to workflow activities.

The resulted workflow can be executed by different BPEL Servers such as Oracle BPEL [6], Microsoft BizTalk [7] or IBM Web Sphere [8].

Our approach uses BPEL and Microsoft BizTalk Server for process modeling and workflow representation and execution. Although BizTalk Server is a friendly environment for designing organization specific workflows, there are some problems that arise from mapping the workflow to BPEL. The main problem that should be addressed is that not all the elements used to model the workflow can be converted into BPEL elements thus leading to incomplete workflow-BPEL mapping. For example, BizTalk Server workflow element Transform, that associates two complex messages, doesn't have a BPEL corresponding element.

3 Workflow-based Modeling and Verification of Industrial Processes

3.1 The Layered Architecture for Workflow–based Modeling

For workflow-based modeling of industrial business processes we propose a layered architecture which is presented in Figure 1. The model is based on service orchestration [9] in which the services communicate only with simple messages. The proposed SOA-based architecture facilitates the reuse of organization specific services and allows the modeling of a wide range of business domains while eliminating the incomplete workflow-BPEL mapping.

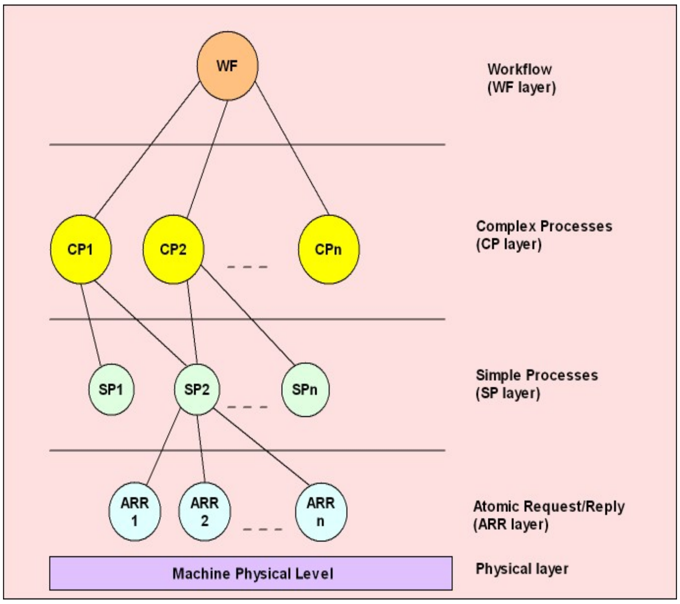

Fig. 1 Hierarchical Architecture for Workflow-based Modeling of Industrial Processes

We propose an incremental methodology for each layer construction. We start from an initial layer that contains physical or simulated machines of the production line on which simple services from the ARR layer are mapped. The rest of the layers are incrementally generated, each increment generating a new layer. The new layer is created if both of the following two conditions hold:

- At least two processes could be identified on top of the existing layers.
- There is at least one specific business rule that leads to the interaction of the processes identified on the topmost layer.

Business rules are usually derived from the business domain or from company specific standards, policies and rules [10]. The way of defining business rules is shown in Figure 2. Using the business rules and process orchestration, new business processes can be created.

In the context of our architecture, an orchestration operator must be defined. The Orchestrator operator ORCS(M, BR)->P, takes the set of processes M, and generate a new composite process P based on a subset of business rules BR.

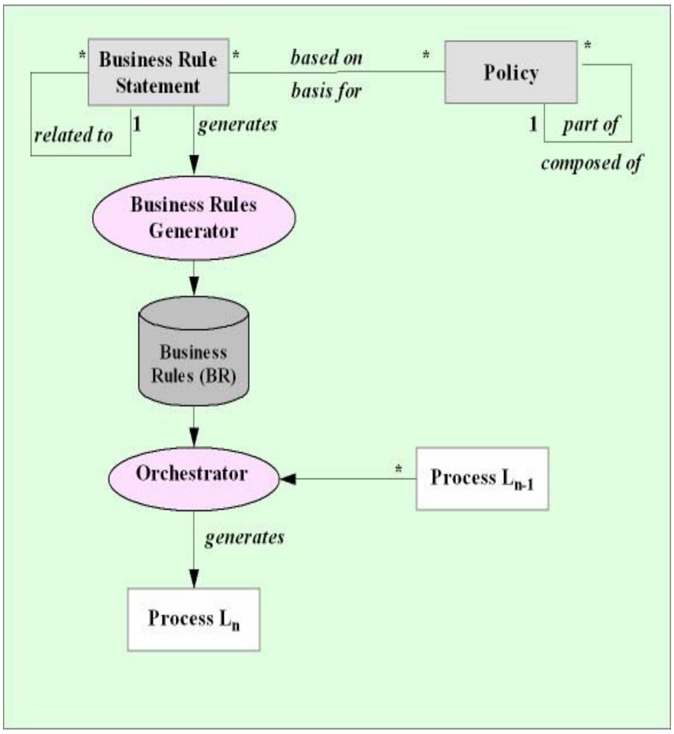

Fig. 2 Business rules and their orchestration

The formalism corresponding to layer construction methodology is given below:

(1) (L_n is created) $\Leftrightarrow$ ($\exists$ M={ P_1 , P_2…, P_k | k>1}) and (($\exists$ N$\subset$M, N,M$\in L_{n-1}$) or (N,M $\in L_{n-1} \cup L_{n-2} \ldots \cup L_1$)) and ($\|N\| \geq 2$) and ($\exists$BR | ORCS(N,BR)-> P$\in L_n$) and ($L_1 \equiv$ ARR)

where P_i are the L_{n-1} level processes and ORCS(N, BR) represents the orchestration of L_{n-1} processes into a process P on L_n level based on specific business rules BR.

3.2 Process Algebra for model description and verification

The processes on each architectural layer are described, modeled and verified using PA CSS [2]. In [2] the authors present a way of describing web service using CSS Process Algebra formalism and how CWB-NC tool is used to ensure some properties, like the correctness of a web service composition. This verification can be done during the design phase. All the processes on each layer are formalized and verified using CSS Process Algebra and CWB-NC [11].

In our situation all processes communicate with simple request/reply messages. Using CSS algebra we can define the following actions: receive message (indicated by the message name) or emit message (indicated by message name prefixed by the quote symbol). The first step in modeling a process is to agree on a set of action names which represent the messages used in the system, such as send, receive, confirm, etc.

In PA, processes are represented as follows. A process which is terminated is written 0: “do nothing”. A process can execute a sequence of actions of the form send.P, where ‘send’ is an action and P is a process with the meaning: “first execute send and then execute P”. This way, a process can perform a nondeterministic choice like P + Q: “execute P or execute Q”. The coexistence of several processes Pi with i = 1..n , whose execution is interleaved, is written: P1| . . . |Pn (“run in parallel P1, . . . , Pn”).

3.3 Layers construction and verification for a sausage production line

In the context of FOOD-TRACE project, for a sausage production line, using the proposed methodology, we have identified four layers and we have developed the system model workflow which follows the food industry business rules. The four specific architectural layers are described below.

The ARR (Atomic Request/Reply) layer, specifies the atomic services that use a request/reply message exchange pattern. The services on this layer interact with the physical level (real or simulated sensors or simple machines), such as those responsible for acquiring the production line parameters (temperature, humidity, etc.).

Using process algebra we can describe the ARR layer processes into the following formal definition:

```
proc getTemperature=  ‘send. receive.getTemperature.0
proc execCutting     =  ‘send.receive.execCutting.0
```

The SP (Simple Processes) layer is generated on top of the ARR layer. This layer contains simple processes that are obtained by composing or orchestrating the atomic processes from the ARR layer using specific business rules. A process is part of the SP layer if the following holds:

(2) $(P \in SP) \Leftrightarrow$
$(\exists\ M=\{P_1, P_2\ldots, P_k \mid k>1\} M \in ARR)$, $(\|M\| \geq 2)$ and
$(\exists\, BR \mid ORCS(M, BR) \rightarrow P \in SP)$

where ORCS(M, BR) represents the orchestration of the set M of ARR layer processes into a process P on SP layer based on business rules BR. The processes which correspond to a single machine of the production line are included in this layer. For example, the process of “meat cutting” corresponds to the meat cutting machine. According to the business rules, the “meat cutting” process orchestrates temperature acquisition and machine starting from the ARR layer.

Figure 3 shows a state diagram of the process algebra representation of “meatCutting” process. Using this state diagram we can identify set of action names which represent the messages exchanged in the “meatCutting” process.

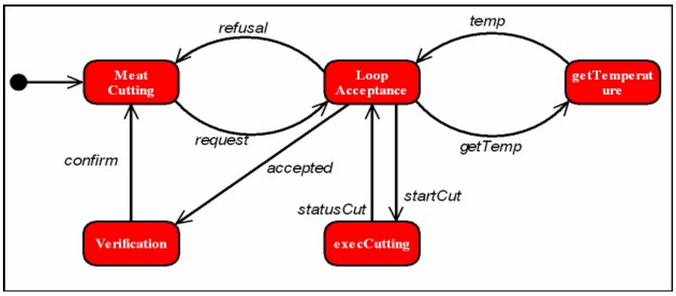

Fig. 3 Meat cutting process

After the set of action names has been identified, the process is described in CSS PA and its correctness is verified. The SP layer “meatCutting” process is defined below:

```
proc meatCutting = request.loopAceptance
proc loopAceptance =
       ‘sendGetTemp.receiveTemp.loopAceptance
           + ( ‘refusual.meatCutting
             + ‘sendStart.receiveStart.loopAceptance
               +( ‘refusual.meatCutting
                 + acceptance.confirm.meatCutting ))
```

The next level, CP (Complex Process) layer, defines complex processes that involve a set of machines working together for achieving a complex task. The definition for this level processes is given below:

$(3)\ (P \in CP) \Leftrightarrow$
$((\exists\ M=\{P_1, P_2\ldots, P_k \mid k>1\}\ M \in SP)$ or
$(M \in SP \cup ARR), (\|M\| \geq 2)\ )$ and
$(\exists BR \mid ORCS(M, BR) \rightarrow P \in CP)$

where ORCS(M, BR) represents the orchestration of the set M of SP and ARR layer processes into a process P on CP layer based on a specific business rules BR.

For example, in the FOOD-TRACE project, consider the process of mixing the meat with ingredients. This is a complex process, which is executed by two machines: the “add-ingredients” machine and the “mixing” machine.

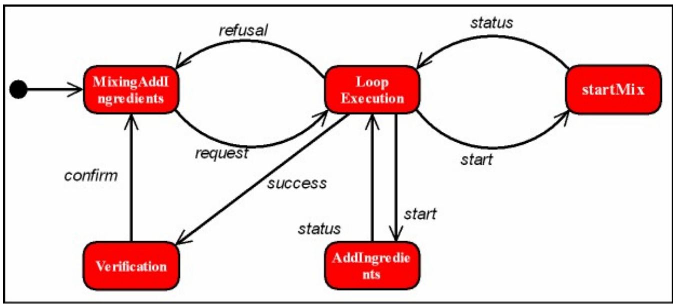

Fig. 4 MixingAddIngr process

Using the state diagram presented in Figure 4, the process can be described in process algebra in the following way:

```
proc mixingAddIngr= request.loopExec
proc loopExec=
    ‘sendStartMix.receiveStartMix.loopExec
     +( ‘refusual.mixingAddIngredients
       +‘sendStartAddIng.recStatusAddIng.loopExec
        +( ‘refusual. mixingAddIngredients
          +acceptance.confirm. mixingAddIngr))
```

The topmost level, the WF (Workflow) layer, represents the workflow which models a specific production line. The workflow is defined as follows:

$(4)\ (W \in WF) \Leftrightarrow$
$((\exists\ M=\{P_1, P_2\ldots, P_k \mid k>1\} M \in CP)$ or
$(M \in CP \cup SP \cup ARR), (\|M\| \geq 2))$ and
$(\exists BR \mid ORCS(M, BR) \rightarrow W)$

where ORCS(M, BR) represents the orchestration of the set M of CP,SP and ARR layer processes into a process P on the CP layer, based on specific business rules BR

The results of the workflow model execution are stored in an internal repository and exposed through web services to organization business partners such as the Organization of Consumer Protection.

4 Modeling and Verification Scenario

Our approach on modeling of business processes is presented in the context of the FOOD-TRACE research project. In this case, at the highest level of modeling abstraction, are food-industry organizations. For example, for a specific sausage recipe, based on company internal procedures, the sausage preparing scenario shown in Figure 5 can be constructed.

The workflow model of the proposed scenario was build by using the layered construction methodology presented in Section 3. We have identified the following atomic request/reply processes: getTemperature, getTime, getHumidity, getOxidation, getWeight and machineStart/Stop. These processes are represented as web services based on the request/reply paradigm and interact directly with the simulated (or real) machines. The simple processes of the SP layer such as “meat-cutting”, “mixing” or “filling” are constructed by orchestrating the atomic request/reply web services.

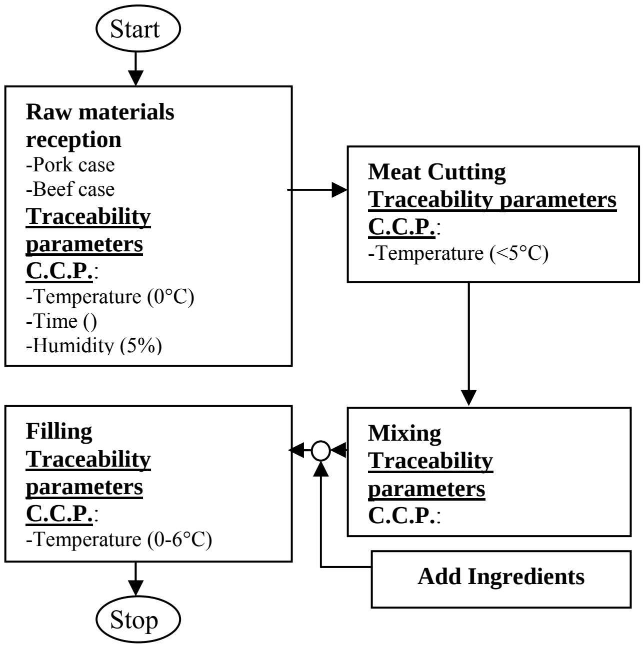

Fig. 5 Workflow scenario

The proposed scenario is represented in Process Algebra CSS and is verified using CWB-NC. This permits the logical faults removal from the workflow model before translating it to BPEL.

Using Microsoft BizTalk Server Orchestrator, the simple services are represented as BizTalk workflows, exported to BPEL processes and saved in a database for a later use. For the complex processes level in the layered architecture, in the proposed scenario, we have identified the process of “Mixing and Add-Ingredients”. The simulated execution is conducted by the BizTalk representation of the workflow model.

Following the scenario, the layered architecture presented in Section 3.1 is instantiated in Figure 6. The resulting model has a tree-like shape which permits traceability operations for every node and its sub-tree. Also, the tree-like shape permits both types of traceability: upstream and downstream. Upstream traceability starts from tree leaves towards the root node of the tree which in this case is the process that generates the product. Downstream traceability takes the root node that represents a process which generates a product and decomposes it into sub-processes tracing them until the leaf nodes.

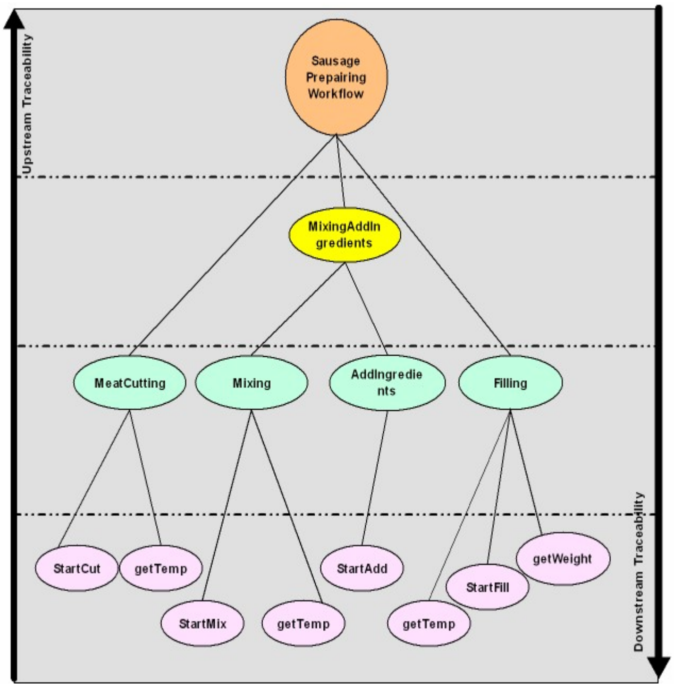

Fig. 6 Instantiated layered architecture

5 Conclusion and Future Development

The paper proposes a methodology for workflow based modeling and verification of complex industrial production lines. The business is modeled by a set of layers of increasing complexity from production line elementary machines and sensors to complex business workflows. The resulted model can be translated and executed by any workflow execution engine.

The proposed methodology was used for modeling of a sausage processing line developed in the context of Food Trace [3] research project. The resulted model, enhanced with traceability elements was translated into BPEL and executed using the Microsoft BizTalk server. The resulted model was also represented using the Process Algebra formalism and verified using CSS Process Algebra and CWB-NC.

For future development, we intend to extend the model by considering the collaboration among business partners and the possibility of dynamic binding of web services to workflow elements.

Acknowledgements

This work was supported by the Food Trace project within the framework of the “Research of Excellence” program initiated by the Romanian Ministry of Education and Research.

References:

[1] M. Rebolledo, *Rough intervals—enhancing intervals for qualitative modeling of technical systems*, Artificial Intelligence 170, 667–685, 2006.

[2] Gwen Sala¨un, Lucas Bordeaux, Marco Schaerf, *Describing and Reasoning on Web Services using Process Algebra*, ICWS 2004

[3] FOOD-TRACE project, *http://www.coned.utcluj.ro/FoodTrace/*.

[4] BPMN (Business Process Modeling Notation), *http://www.bpmn.org/*.

[5] BPEL4WS Specifications, *http://www-128.ibm.com/developerworks/library/specification/ws-bpel/*.

[6] Oracle BPEL, *www.oracle.com/technology/bpel*

[7] Microsoft Biztalk Server 2006, *http://www.microsoft.com/biztalk/default.mspx*.

[8] Web Sphere, *www.ibm.com/software/ websphere*.

[9] Service Orchestration, www.serviceoriented.org.

[10] Defining business rules http://www.businessrulesgroup.org/first_paper/

[11] CWB-NC http://www.cs.sunysb.edu/~cwb/1.11/install.html

[12] Moscato, F., Mazzocca, N., Vittorini, *Workflow Pattern Analysis in Web Services Orchestration: The BPEL4WS Example*, 1st International Conference on High Performance Computing and Communications 2005, LNCS 3726, 395-400.

[13] Li, H., Lu, Z, *Decentralized Workflow Modeling and Execution in Service-Oriented Computing Environment*, IEEE International Workshop on Service-Oriented System Engineering, 2005.

[14] Mate Kovacs, Laszlo Gonczy, *Simulation and Formal Analysis of Workflow Models*, Electronic Notes in Theoretical Computer Science, *www.elsevier.nl/locate/entcs*