A Layered Workflow Model Enhanced with Process Algebra Verification for Industrial 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
Ioan.Salomie@cs.utcluj.ro   http://dsrl.coned.utcluj.ro

Abstract

This paper addresses the problem of modeling and verification of complex industrial production lines that include physical machines. 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. For workflow verification purposes, the model is represented in Process Algebra (PA) formalism which allows for reasoning and checking the correctness of the business process model and early identification of any logical faults in the model design phase. The resulted model, validated using with the PA formalism can be translated, deployed and executed by any workflow execution engine. The proposed methodology was used for modeling and verifying a sausage processing line, developed in the context of the Food Trace [4] research project.

1. Introduction

Business processes recently became the main elements of large scale industrial systems. Therefore, the need for their 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 intra-business processes and inter-business processes. The intra-business processes are modeling complex company-specific activities such as industrial workflows that drive industrial production lines. For product manufacturing traceability the intra-business process model should also contain traceability features. The model should allow for both upstream and downstream product traceability. Upstream traceability starts from row materials and concludes to the final product while downstream traceability takes the product and decomposes it into sub-products and traces them down until the row materials.

Inter-business processes, or business to business cooperation, imply an active collaboration of a set of business partners based on specific business rules.

This paper focuses on modeling intra-business processes representing industrial production lines consisting of a set of cooperating physical machines. The final result of industrial line modeling should be a workflow corresponding to manufacturing technology of a certain product.

The direct execution of industrial workflows on production line machines before the integration testing can be very expensive and may lead to improper machine operation and even to unrecoverable faults. For integration testing purposes, in order to simulate workflow execution on production lines physical machines we have proposed a physical machine simulation model based on probabilistic state machines [1]. The proposed simulated environment uses the layered workflow model presented in the current paper.

Relevant simulation results are obtained only if the model accurately reproduces the real physical system. Due to the complexity level, modeling real systems is not always a simple task, being difficult to be

185

described with precise deterministic mathematic models.

In the context of business process modeling and execution, many approaches have been proposed. In [2], the authors propose the SQMA model (Situation-based Qualitative Modeling and Analysis) for representing and simulating industrial systems using Rough Set Intervals. The proposed model uses interval-based representation for qualitative models in order to implement the behavior of real systems. The SQMA model hierarchically structures the whole system and decomposes system levels into components. Component variables are modeled using intervals and characteristic values represented as a one-value interval. The description of each component is completed by defining interval arithmetic based constraint rules for model verification. Using Rough Set Intervals and constraint rules, a transition matrix between components is constructed and used in simulation. The main disadvantages of this approach consist of inaccurate representation of machine's business logic on one hand and difficulties in model management on another hand. Also it proves difficult to model complex business scenarios involving more cooperating machines.

Another modeling approach uses formal description techniques for specifying industrial business processes, especially when concurrent and communicating components represented as web services (WSs) are involved. WSs and their interaction are best described using process description languages like Process Algebra (PA) [3]. 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 time. The main advantage of using PA for the description and modeling of simple and complex business processes is that it allows for the verification of the correctness of the obtained business process model on one hand and the identification of the logical faults on the other hand.

Our approach on modeling and simulating industrial business processes is presented in the context of the Food Trace research project [4]. 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 industrial production lines and their physical machines by using business processes.

The objective of this paper is to define a method for capturing and modeling business processes involving production lines and physical machines, targeting their workflow integration and simulation.

This objective was achieved by: (i) defining a methodology for the construction of a workflow model corresponding to a product manufacturing technology according to specific business rules; (ii) verifying model correctness and consistency using the Process Algebra formalism; (iii) translating the verified model into executable workflows.

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 and consistency. In section 4, a business scenario for food processing and traceability is modeled using the defined methodology. Section 5 gives conclusions and promising future work.

2. Industrial workflow model - design and execution requirements

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

The business level contains organization entities that compete or interact to achieve their goals according to organization's policies and domain rules.

The 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.

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

The next section focuses on the 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, such as BPMN [5], which is then converted into an executable language such as BPEL [6]. The second approach describes the processes directly in BPEL.

In the context of the Food Trace project, we have identified the following requirements that should be addressed during industrial workflow model design:

1. The resulted workflow model should correspond to a specific product manufacturing technology.

2. The model should capture product traceability features.

3. The model should allow both upstream and downstream traceability.

The resulted workflow can be executed by different BPEL Servers such as Oracle BPEL [7], Microsoft BizTalk [8] or IBM Web Sphere [9].

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 (see workflow element Transform for example) can be converted into BPEL elements thus leading to incomplete workflow-BPEL mapping.

3. Workflow-based Modeling of Industrial Processes

One way to present collaboration and interaction among physical machines, part of an industrial production line, is by using workflows. Mapping and modeling real processes onto workflows is an open research problem. Usually, this mapping is achieved in two steps. In the first step, the real processes are divided into simple atomic processes while in the second step the atomic processes are represented as web services interconnected by a workflow model.

For modeling the workflow associated with an industrial production line we propose a layered architecture which is presented in Figure 1. The model is based on service orchestration [10] 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.

3.1. Model Construction

This paper proposes an incremental methodology for each architectural layer construction. We start from the physical machines (or from their simulated model) of the production line on which simple atomic services based on request/reply paradigm are mapped. The

target is to obtain the workflow model corresponding to the product manufacturing line on the uppermost level of the hierarchy. The intermediary layers are incrementally generated, each increment generating a new layer. The new layer is created if both of the following two conditions hold: (i) at least two processes could be identified on top of the existing layers and (ii) 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 [11]. Using the business rules and process orchestration, new business processes can be created. Workflow construction methodology uses an orchestration operator for generating new processes.
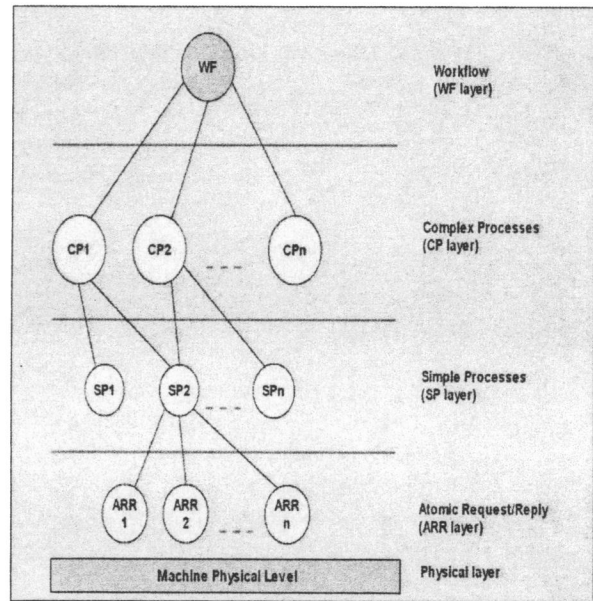
Figure 1. Workflow Model Architecture

A layer n process is generated by using the orchestration operator defined as ORCS (N, BR) -> P
The operator takes the set of processes N from inferior architectural layers (L_1 ... L_{n-1}) and generates a new composite process P based on a subset of domain business rules BR.

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

$((\exists \; N, M \; | \; N,M \in L_{n-1})$ or

$(N, M \in L_{n-1} \cup L_{n-2} ... \cup L_1))$, $(\|N\| \geq 2)$ and

$(\exists \; BR \; | \; ORCS(N,BR)\text{->} P \in L_n)$ and $(L_1 \equiv ARR)$

where P_i are the L_{n-1} .. L_1 level processes and ORCS (N, BR) represents the orchestration of L_{n-1} .. L_1 processes into a process P on L_n level based on specific business rules BR.

3.2. Model layers

In the context of FOOD-TRACE project, the hierarchical workflow model for food manufacturing products is organized on four layers (Figure 1).

The bottom layer, ARR (Atomic Request/Reply), 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.).

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 domain specific rules. A process is part of the SP layer if the following holds:

(2) $(P \in SP) \Leftrightarrow$

    $(\exists\ M = \{P_1, P_2..., P_k \mid k>1\}, M \in ARR)$,

    $(\|M\| \geq 2)$ and $(\exists\ BR \mid ORCS\ (M, BR)\text{->}P)$

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

The 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) $(P \in CP) \Leftrightarrow$

    $((\exists\ M = \{P_1, P_2..., P_k \mid k>1\}, M \in SP)$ or

    $(M \in SP \cup ARR), (\|M\| \geq 2))$ and

    $(\exists\ BR \mid ORCS\ (M, BR)\text{->}P)$

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

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

(4) $(W \in WF) \Leftrightarrow$

    $((\exists\ M = \{P_1, P_2..., P_k \mid k>1\}, M \in CP)$ or

    $(M \in CP \cup SP \cup ARR), (\|M\| \geq 2))$ and

    $(\exists\ BR \mid ORCS\ (M, BR)\text{->}W)$

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

3.3. Process Algebra for Model Description and Verification

Executing workflow modeled business processes that involve physical machines before verifying the model correctness and consistency may cause errors that could lead to improper operation of the industrial system. Workflow model correctness and consistency can be verified during the design phase using the PA formalism for describing and reasoning on business process.

For a specific workflow, instance of the model, the processes on each architectural layer are described, modeled and verified using PA CSS [3]. In [3] the authors present a way of describing web services using CSS Process Algebra formalism and how CWB-NC tool is used to ensure the correctness of web service composition. This verification can be done during the workflow design phase. All the processes on each layer are formalized and verified using CSS Process Algebra and CWB-NC [12].

In our model, all processes communicate with simple request/reply messages. Using CSS algebra the following actions can be defined: "receive message" (indicated by the message name) and "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, the processes are represented as follows.

A process which is terminated is symbolized as: 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 P_i with $i = 1..n$ whose execution is interleaved, is represented as:

$P_1 \mid \ldots \mid P_n$ "run in parallel $P_1, \ldots, P_n$"

4. Case Study - Model construction and verification for a sausage production line

We have used the methodology for hierarchical model construction described in the previous section and the sausage preparing scenario illustrated in Figure 2 for building the workflow model for a sausage production line which is illustrated in Figure 3.

In Figure 3, the processes on each layer are instances of generic processes of the model represented in Figure 1. The processes on each layer are described in Process Algebra CSS and verified using CWB-NC.

This permits the removal of logical faults from the workflow model before translating it to BPEL. On the ARR (Atomic Request/Reply) layer we have identified the following atomic request/reply processes: getTemperature, getTime, getHumidity, getOxidation, getWeight and machineStart/Stop.
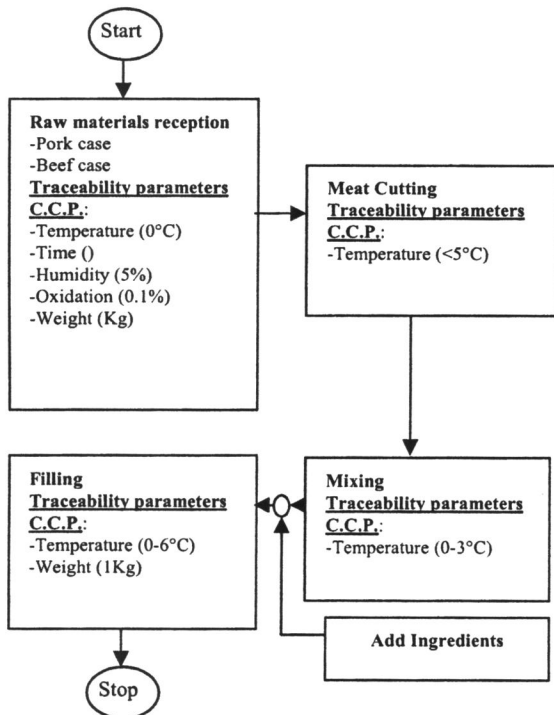

Figure 3. Instantiated layered architecture

Using process algebra we represent the ARR layer processes by the following formal definition:

proc getTemperature= 'send. receive.getTemperature.0
proc execCutting= 'send.receive.execCutting.0

The SP identified processes, such as "meatCutting", "mixing" or "filling" are constructed by orchestrating the atomic request/reply web services of the ARR layer. For example, according to the business rules, the "meatCutting" process orchestrates temperature acquisition and machine starting processes from the ARR layer. Figure 4 shows the state diagram of the "meatCutting" process which is mapped on a meat cutting physical machine.

Figure 2. Sausage preparing scenario

These processes which are directly interacting with the production line real or simulated physical machines are represented as web services based on the request/reply paradigm.
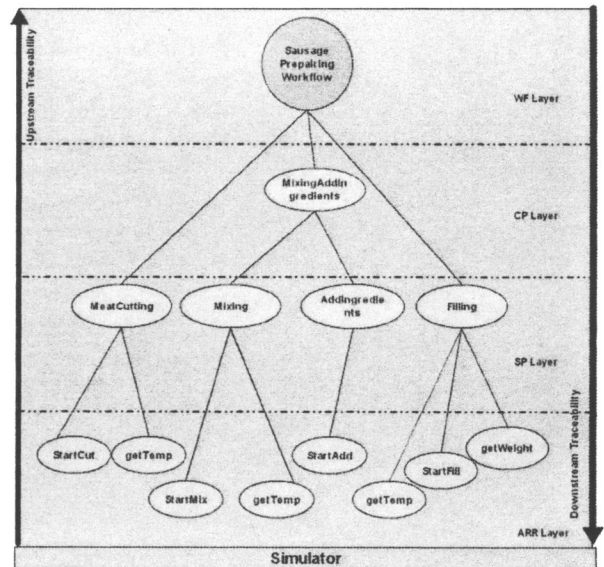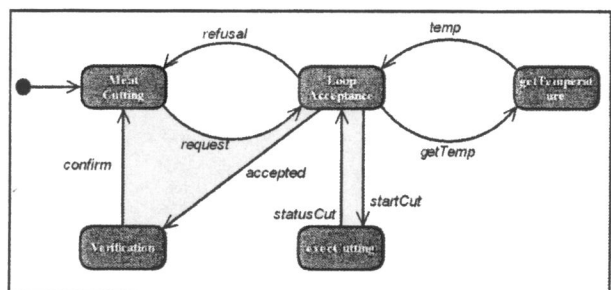

Figure 4. "meatCutting" process

The state diagram shows machine states and state transition conditions. Each machine state represents an associated machine action while each transition condition represents a message. Using this state

189

diagram, the set of "action names" representing the messages exchanged in the "meatCutting" process, can be identified. The identified "action names" are used for process description in CSS PA followed by process correctness verification. The SP layer "meatCutting" process is defined below:

proc meatCutting = request.loopAceptance
proc loopAceptance =
     'sendGetTemp.receiveTemp.loopAceptance
   + ('refusal.meatCutting
    + 'sendStart.receiveStart.loopAceptance
     + ('refusal.meatCutting
      + acceptance.confirm.meatCutting))

The CP layer defines complex processes that involve a set of machines working together for achieving a complex task. The CP type process identified for the sausage manufacturing production line consists of the complex process "mixingAddIngredients". This process is mapped onto two physical machines: the "addIngredients" machine and the "mixing" machine

Using the state diagram presented in Figure 5, the process "mixingAddIngredients" is described below in process algebra formalism:

proc mixingAddIngr = request.loopExec
  proc loopExec =
    'sendStartMix.receiveStartMix.loopExec
    + ('refusal.mixingAddIngredients
    +'sendStartAddIng.recStatusAddIng.loopExec
     + ('refusal. mixingAddIngredients
     + acceptance.confirm. mixingAddIngr))
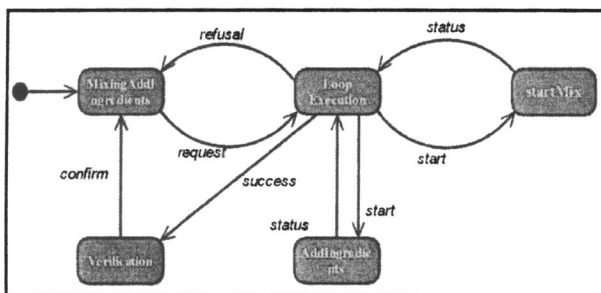
Figure 5. "mixingAddIngredients" process

The topmost level, defines the "sausagePreparingWorkflow" corresponding to a certain sausage manufacturing recipe.

The tree-like representation of the hierarchical workflow model, corresponding to a product manufacturing technology (see Figure 3 for the sausage production line), permits both upstream and downstream traceability operations.

Using Microsoft BizTalk Server Orchestrator, different designed workflow models corresponding to different product manufacturing technologies and recipes, are represented as BizTalk workflows, exported to BPEL or saved in a product workflow repository for a later use.

The results of the workflow model execution, including traceability parameter values captured during product manufacturing processes are stored in repositories and exposed through web services to business partners or control bodies such as the Organization of Consumer Protection

5. Conclusion and Future Development

The paper proposes a methodology for workflow based modeling and verification of complex industrial production lines. The industrial production lines are modeled using a set of layers of increasing complexity from elementary machines and sensors to complex business workflows. The resulted model can be instantiated to specific product manufacturing technology, verified for correctness by using process algebra formalism, translated to BPEL and executed by any workflow execution engine.

The proposed methodology was used for modeling a sausage production line in the context of Food Trace [4] research project. The resulted model, enhanced with traceability elements was verified using CSS Process Algebra and CWB-NC and executed using the Microsoft BizTalk server.

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

References:
[1] I. Salomie, T. Cioara, I. Anghel, M. Dinsoreanu, T. I. Salomie, Machine Simulation for Workflow Integration Testing, submitted for ICCP 2007
[2] M. Rebolledo, Rough intervals—enhancing intervals for qualitative modeling of technical systems, Artificial Intelligence 170, 667–685, 2006.
[3] G. Salaun, L. Bordeaux, M. Schaerf, Describing and Reasoning on Web Services using Process Algebra, ICWS 2004

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

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

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

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

[8] Microsoft Biztalk Server 2006, http://www.micro
soft.com/biztalk/default.mspx.

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

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

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

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

[13] F. Moscato, N. Mazzocca, Vittorini, Workflow Pattern
Analysis in Web Services Orchestration: The BPEL4WS
Example, 1st Internat. Conf. on High Performance Computing
and Comm. 2005, LNCS 3726, 395-400.

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