A SYNTHESIS OF BUSINESS ROLE MODELS
Conceptual Tool for Business Process Innovations

Pavel Balabko, Alain Wegmann
Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne, EPFL-IC-LAMS, Lausanne, Switzerland
Email: pavel.balabko@epfl.ch, alain.wegmann@epfl.ch

Keywords: Business process, BPR, innovation, role, role model, synthesis, synthesis constraints.

Abstract: Modern Information and Communication Technology open a door for innovations that can improve the functioning of companies. Many innovations can come from the analysis of business processes. Today modeling is widely used for the analysis of business processes. In these work we propose a process modeling technique based on role modeling. To specify a process where one business object may play several roles, a synthesis operation (the composition of two base roles in a third role) has to be specified. All role-based techniques have difficulties specifying role synthesis: synthesis is never specified without the description of actual messages passing between business roles. Such implementation details complicate the understanding of the model and semantics of synthesis become implicit. To specify a business process of a complex system at a higher level of abstraction requires the proper understanding of relationships between roles, when they are put together in one common context. In this paper we define the concept of "synthesis constraints" that shows relations between roles. Using "synthesis constraints" allows a business modeler to make explicit his decisions about how the synthesis is done in an abstract and implementation independent way. This approach can be used for building a BPR case tool that enables the discovery of new business processes by means of different disassembling and assembling of roles.

1 INTRODUCTION

To stay at the competitive edge, the modern market requires companies to adopt new business strategies that allow them to take advantages from modern Information and Communication Technology (ICT). "In the emerging era of the Web, companies that don't have a Web-based strategy - don't have a strategy" (Miers, 2001). Rapid development of Web-based technologies provides an inexpensive way to communicate with customers, partners and suppliers. A cheap, easy and fast way of the information exchange is the main condition that enables the distribution of activities between different partners. In the "pre-internet era", companies were limited in their communication facilities and therefore the distribution of activities was difficult. Today due to the development of ICT, internal activities of companies often become outsourced to partners and some internal activities may become parts of customer's processes. ICT allows companies to outsource their non-core activities and better define their competence. "… Within the past five to eight years, outsourcing has evolved from a purely tactical option - often of last resort - to an ongoing, standard business practice and strategic management tool" (Berger, 2002).

If we agree that it makes sense to outsource, we have to consider internal activities that can be outsourced. Many outsourcing solutions can be possible. They should be analyzed in two steps. First, a business analyst has to build a model of each solution. Second, solutions should be evaluated and the best one should be taken. In our work we are not going to discuss the second step that determines which outsourcing solution should be taken.

We focus on the question: "How different outsourcing solutions can be found based on the analysis of the business process of a company and existing service providers?" In a company that has a complex business process it can be difficult to identify internal activities that can be outsourced. Internal activities can be so closely interrelated that it can be difficult even to distinguish between them: an internal activity may have many internal constraints (behavioral, structural etc.) with other

internal activities. Outsourcing of an activity requires that some internal constraints should be changed in the context of the business process of a company and some internal constraints should be transformed into external constraints (or contracts) between a company and its partners. Also new constraints should be defined between an outsourced activity and activities in the context of the business process of partners. This is a typical Business Process Reengineering (BPR) problem.

Before describing our solution to the mentioned problem we begin with the overview of the business process concept. Business process is only a part of the whole model that shows how the company does its business. Other parts are the business model that have "a description what values the business creates for its stakeholders" (Stähler, 2002) and the implementation that shows how the business process is implemented. Innovations can come from any of these three parts.

In this work, we consider innovations that come from Business Process Part. We take the definition of the business process from the Workflow Reference Model (WMC, 1995):

Business Process *is "a set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships".*

This definition extends the definition proposed by Davenport & Short: a *business process* is "a set of logically related tasks performed to achieve a defined business outcome" (Davenport, 1990). It is extended with the notion of roles that are used to specify the responsibilities of objects. *Object* (or business object) is yet another concept that should be considered when the business process has to be implemented. An object is the model of an entity in the Universe of Discourse. It plays roles in a business process by means of participating in different activities. Totally we have to address four key elements: *Goals, Activities*, *Roles* and *Objects*. (Kueng, 1996).

To model these four business process elements some Process Modeling Techniques (PMTs) should be used. Carlsen in (Carlsen, 1997) refers to the following PMTs types: Traditional Input Process Output (IPO) techniques; Conversation Based techniques; PMTs based on role modeling; System thinking and system dynamics techniques; Constrained-based representations techniques. In this work, we use PMT based on role modeling. Its main advantage is that it supports the separation of concerns: the best way to deal with the business process of a complex system is to specify concerns separately and show how these concerns are synthesized together in the context of a business system. Each concern can be specified as a separate activity and a business process is a synthesis of different concerns.

There are several PMTs based on role modeling. The three of them seem to be the most important: RIN – Role Interaction Networks (Singh, 1992), RAD – Role Activity Diagram (Ould, 1995) and OORAM – the Object-Oriented Role Analysis Method (Reenskaug, 1996). These three approaches are quite similar. Roles are considered as sets of sequentially ordered actions and/or interactions

The main drawback of the PMTs described above is that goals of business processes are difficult to model with these PMTs. To model goals we have to specify states of business objects because "… goals are usually defined as objectives to be achieved by the system and its environment or a state of affairs that is deemed desirable by stakeholders" (Regev, 2001).

Another important requirement for PMT is that it should be business oriented and thus be simple and diagrammatic. This will allow a business analyst "…to discuss and validate process models with both users and owners of the process, many of whom are not prepared to invest their time in understanding more complex representations" (Phalp, 1997).

PMT that allows for the simulation of a business process can help a business analyst to evaluate a process. Making the simulation possible requires that PMT supports the formalization of a business process. Such formal and rigorous PMT can be used in a BPR case tool for the development and analysis of business processes. We propose a role modeling PMT that satisfies the three mentioned requirements.

As we said above, the role modeling PMT allows for the separation of concerns: a business model is specified as a set of concerns synthesized together. Therefore synthesis is a necessary operation of the role modeling PMT. In section 2 we show how synthesis can be specified and implemented. First, we consider the semantics of a synthesis operation. Then, we show how the specification of a synthesis can be done. To specify it we define a new concept, called *synthesis constraints*. The definition of synthesis constraints is the main contribution of this paper. We show how synthesis constraints can be used to specify decisions taken by a business analyst during synthesis. The specification of synthesis is done in an implementation independent way that is convenient for the business reasoning. In the end of section 2 we explain how the implementation

independent specification of synthesis can be implemented: we introduce two patters for building implementation models.

Role modeling PMT presented in this paper can be used to build a case tool for the specification and discovery of new business processes by means of different disassembling and reassembling of roles. In section 3 we give two practical examples. In the first example we show how synthesis constraints are used to specify a business process as a synthesis of separate concerns of a system. In the second example we show how reassembling of roles in a business process can help do discover new business processes and models.

2 SYNTHESIS OF ROLE MODELS

In this section we define the concept of *synthesis constraints* that is used to specify how different concerns of a business system can be synthesized together into one model. We define synthesis constraints in the context of our PMT. We begin this section with the description of our PMT.

2.1 Our Approach for Business Process Modeling

In this subsection we present concepts that we use in our PMT One of the requirements that we mentioned in the previous section is that PMT should support rigorous and formal modeling of a business process. In order to give rigorous definitions for concepts that we use in our PMT we have to choose a consistent semantic framework. We use the ISO/ITU standard "Reference Model for Open Distributed Processing" – part 2 (ISO, 1996) as a framework.

Based on RM-ODP, modeling consists of identifying entities in the universe of discourse and representing them in a model. The *universe of discourse* corresponds to what is perceived as being reality by a business analyst and *entity* is "any concrete or abstract thing of interest" (ISO, 1996) in the universe of discourse. Identified entities are modeled as *model elements* in a *model*. Model elements are different modeling concepts (object, action, behavior etc). We give definitions of some modeling concepts necessary for the understanding of our paper (other definitions see in the RM-ODP). We begin with the definition of an object. If in the universe of discourse we have entities that can be modeled with state and behavior, we model this entities as objects:

Object: *"An object is characterized by its behavior and dually by its state"* (ISO, 1996).

The duality of state and behavior means that the state of an object determines the subsequent behavior of this object. The definition of an object is based on the definition of behavior and state:

Behavior: *A collection of actions and a set of (sequential) relations between actions.*

State: *A collection of attributes, attribute values and relations between attributes.*

Attributes can change their values; relations between attributes can also change. To specify these changes we use pre- and postconditions. For any action we will specify "*an interval of arbitrary size in time at which action can occur"* (ISO, 1996). Based on the definition of behavior we define a role[1]:

Role: "*An abstraction of the behavior of an object" defined in a given context.*

To represent the context where a role is defined, we will use the *role model* term, inspired by OOram (Reenskaug, 1996):

Role Model *specifies the set of collaborating roles along with their state and behavior.*

From this definition we can see that a role model can be used to specify all four elements of a business process: roles, activities (or collaborations), goals (postconditions for all roles in a role model) and objects (that play roles in the implementation). Let's give some comments about how business process goals are modeled with postconditions. In our RM-ODP role based approach, the goal of any collaboration is specified as a set of postconditions for each role that participate in this collaboration. A business process can include several collaborations (each one represents a separate concern of a system). Therefore the goal of this business process is a set of goals for each collaboration or the set of postconditions for all roles in a role model.

To represent visually the structure of collaborating objects in a business process, we use the notation inspired by UML. We represent a role model (see figure 1) by collaboration (a dashed oval), roles by stick men or boxes and role names by names below stick men or at the end of a line connecting collaborations with roles. If we want to hide details about roles we represent them with stick men (see figure 1.c from the right). Otherwise we represent roles by boxes with three panes. This notation is similar to the notation used in a UML Collaboration Diagram. The difference is that

[1] This definition is inspired by the definition of Role in RM-ODP and [Genilloud00].

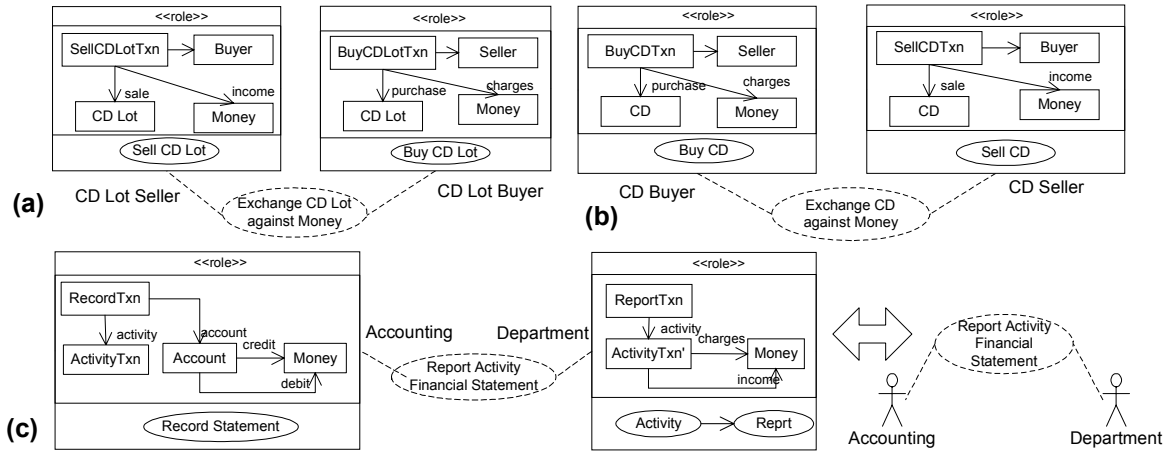

Figure 1: Three base role models 1.a: "Exchange CD Lot against Money", 1.b: "Exchange CD against Money" and 1.c: "Report Activity Financial Statement"

instead of an attribute compartment in UML (middle pane in each box) we use graphical notation inspired by a UML class diagram to represent a state as a set of attributes and relations between them. Instead of the compartment that holds a list of operations in UML (lower pane in each box) we use the graphical notation inspired by a UML activity diagram to represent a role behavior.

We use this notation to model the three role models that we use as examples in the following sections. They are taken from the StreamCom project supported by the Swiss National Science Foundation (project No. 5003-05755). This project studied issues related to the commercialization of streamed information, such as video, audio and news-feeds. The first two role models (figure 1.a and 1.b) describe CDs Lot selling/buying activity. The third model (figure 1.c) specifies how accounting has to be done for a core company department: first the department accomplishes its core business activity and then sends a financial report about this activity to the accounting department. Then the accounting department credits/debits an amount corresponding to the performed activity to/from the company account. In the following sections of this paper we show how these role models can be combined to make a composite role model.

Each role model in figure 1 includes one collaboration and two roles represented by stick men. The goal of each role model is specified as postconditions for each role participating in this activity. Postconditions can be specified formally using the OCL language. However due to the limit of pages in this work we do not shown postconditions explicitly. We suggest using names of the role models such that a reader of the diagram can «guess» them. For example, the "Exchange CD Lot against Money" role model assumes the following postconditions: "CDLotBuyer got CD Lot from CDLotSeller" and "CDLotSeller got money from CDLotBuyer".

2.2 Synthesis Semantics

If a business process includes several concerns (or activities), then we represent these concerns with several role models. In the context of a business process these role models (and corresponding roles) are mutually dependent. What does a mutual dependence between roles mean? To explain the meaning of the mutual dependence of roles we use figure 2. The upper part of figure 2 shows the universe of discourse, the lower part represents the model of the universe of discourse. Let's suppose that we have a business analyst who modeled separately two concerns of a system with two role models from the previous subsection: "Exchange CD Lot against Money" and "Report Activity Financial Statement". We can see that the business analyst decided to model Entity6 as the Department role (where he modeled Entity2, Entity3, and Entity4 as attributes and Entity1 as the Activity action). He also decided to model Entity7 as the CDLot Buyer role (where he modeled Entity2, Entity3, and Entity5 as attributes and Entity1 as an the Buy CD action). To make a bigger model he wants to synthesize these two role models. If we consider the semantics of role synthesis we can see that roles are mutually dependant, i.e. behavior of different roles can influence behavior of other roles involved in synthesis. By looking in the universe of discourse we can see that the Money attribute of the CDLot Buyer role and the Money attribute of the

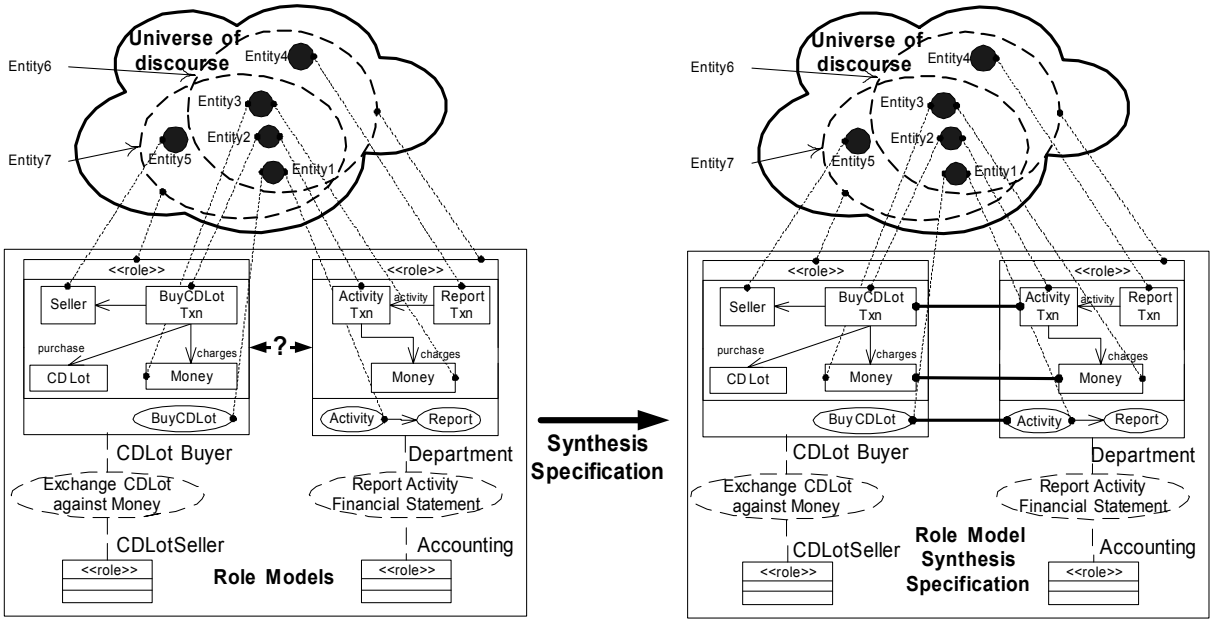

Figure 2: The semantics of synthesis constraints; a: Two base role models and their semantic meaning; b: Role Model Synthesis Specification: base role models and synthesis constraints.

Department role have the same semantic meaning since they model the same Entity3 (the same thing we can say about BuyCDLotTxn and ActivityTxn attributes and the "Buy CD" and "Activity" actions). How this can be specified?

The dependence between roles can be specified in several ways. Examples of different syntax for expressing role's mutual influence can be seen in (Riehle, 1998), (Fiadeiro, 2001), (Reenskaug, 1996). But the syntax proposed in these works depends on implementation details: synthesis is specified as a communication through given role's interfaces. These technical details complicate understanding of synthesis semantics. That is why we believe that the syntax abstract from the implementation details can be extremely useful in the modeling of business systems. This can help a business analyst to concentrate his efforts on design decisions. Decisions taken by a business analyst have to be based on the semantics of the synthesis rather than on implementation. In the next subsection we define a *Role Models Synthesis Specification* that reflects the semantics of the synthesis and makes abstraction from the implementation details.

2.3 Role Model Synthesis Specification

Our Role Models Synthesis Specification is based on the concept of *synthesis constraints*. As we have seen in the previous section, we have to define a new concept that will specify how base roles influence each other within a new synthesized model. Synthesis constraints are used for this purpose:

Synthesis constraints: *Constraints implied on the behavior of base roles*[2] *from different base role models.*

In general, synthesis constraints can be specified as a mapping of model elements used in the specifications of the basic roles. As an example we give definitions of three synthesis constraints that we have used in our research work[3] (more formal definitions we give in our technical report (Balabko, 2002)):

- **Attribute Equality** ($Role_1.attr_1$ ⊷ $Role_1.attr_2$)

It specifies that values of $attr_1$ and $attr_2$ should be equal at any time moments specified by $Role_1$ and $Role_2$ correspondingly.

- **Constraints of Sequentiality** ($Role_1.action_1 \rightarrow Role_2.action_2$).

It specifies that the second action can start at any time after the completion of the first action. In details about constraints of sequentiality you can read in (Balabko, 2001).

- **Action Equality** ($Role_1.action_1$ ⊷ $Role_2.action_2$).

[2] We call role models before synthesis *base role models* and correspondingly roles, *base roles*. We call role models after synthesis *synthesized role model* and correspondingly roles, *synthesized roles*.

[3] This list is not exhaustive and can be extended with other useful synthesis constraints.

It specifies that two actions happen at the same time and preconditions should be satisfied for both actions before these actions can occur.

To specify the synthesis of role models we have to specify a set of base role models and synthesis constraints that relate them. We call such specification as a *Role Model Synthesis Specification*. An example of a role model synthesis specification you can see in Figure 3.a. It represents two role models: "Exchange CDLot against Money" and "Report Activity Financial Statement" and the synthesis constraints.

A Role Model Synthesis Specification allows a business analyst to make explicit his decisions about how synthesis is done. These decisions are made based on what a business analyst observes in the universe of discourse. The main advantage of a Role Model Synthesis Specification is that it allows a specification to be abstract from an implementation: a business analyst does not need to decide on how and by which object the synthesis constraints would be implemented.

2.4 Role Model Synthesis Implementation

In this work we use a role model to specify a business process. The specification of a business process should not include implementation details. That is way we do not consider how the role model should be implemented in details. We would like to explain a general strategy for the implementation of a Role Model Synthesis Specification. To implement it, we have to take decisions on objects that would be responsible for the implementation of synthesis constraints. Here, two situations are possible: when one object plays both base roles and the same object is responsible for the implementation of the synthesis constraints. In the second situation there are two objects playing base roles and the responsibility for the implementation of synthesis constraints is distributed between them. This brings us to two possible implementation patterns of the Role Synthesis Specification: an implementation by merging base roles into one synthesized role and implementation by extending the behavior of the base roles.

Let's consider an example that shows how a Role Model Synthesis Specification can be implemented using two implementation patterns.

The first implementation, *"Merging Base Roles*", merges base roles by putting them in the common context of a new role ("Supply Department" in figure 3.b). In this case the "Supply Department" role is entirely responsible for carrying out the synthesis constraints. However this way of specification seems easy, but it has a significant drawback: it does not allow us to separate base roles to implement them with different business objects (to keep them separate, for example for the purpose of distribution).

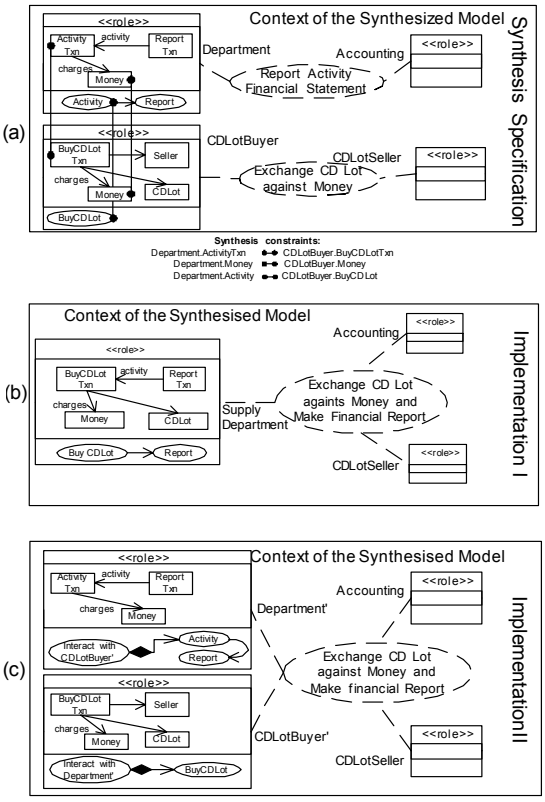

Figure 3: Two implementation patterns: (a) Role Model Synthesis Specification; implementations: (b) Merging Base Roles and (c) Extending Base Roles.

The second implementation, "*Extending Base Roles*", "implements" synthesis constraints by means of extending the behavior of base roles with some additional behaviors that specify how the base roles can interact (see Fig. 3.c). Additional behaviors are the two actions "Interact with Department´/CDLotBuyer´". These actions allow for the communication between roles: they guaranty the equality of attributes and that the Activity and BuyCDLot actions happen "simultaneously". Using the "Extending Base Roles" Implementation allows us to distribute base roles (Department and CDLotBuyer) between two objects.

3 APPLICATION

In this section, we show practical results of our PMT based on synthesis constraints. We hope that examples form this section can serve as a prototype for the BPR case tool that can be built based on out PMT.

We saw that specifications with synthesis constraints make abstraction from the implementation details. This allows a business analyst to concentrate efforts on design decisions rather than on an implementation. This can be especially useful when specifying business processes as synthesis of separate concerns. Specifications can be detailed and outlined.

A detailed specification represents roles as boxes with state and behavioral parts and includes the specification of synthesis constraints. In subsection 3.1 we show an example of how a detailed specification can be used to specify a business process form the set of base role models.

An outlined specification specifies synthesis at a higher level of abstraction. It represents roles as stick men and hides details of synthesis. In section 3.2 we show an example of how an outlined specification can be used in BPR.

3.1 Role Model Synthesis: Detailed Specification

In this subsection we give an example of a detailed specification that shows how a business process can be assembled from the set of base role models.

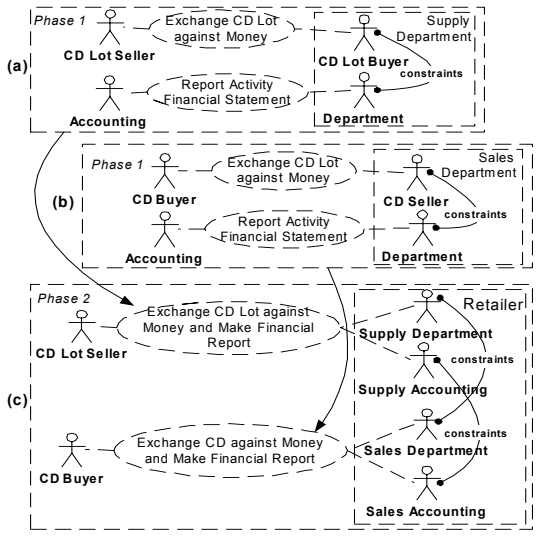

Figure 4: An Example of the synthesis of role models. (a), (b): Phase 1; (c): Phase2.

Due to the limit of pages we give only an overview of the example. The complete example is given in our technical report (Balabko, 2002).

The idea of this example is the following: based on these three base role models form figure 1, the goal is to build a new "Retail CD" business process that will specify a CD retailing business. For this we specify the synthesis of the CDLotBuyer and CDSeller roles in such a way that they share the same account for their business activities. Following this idea, we will synthesize base roles in two phases.

In the first phase we include the accounting activity in the "Exchange CD lot against Money" role model: we synthesize the CDLotBuyer role with the Department role (see figure 4.a) by means of the following synthesis constraints:

Department.Activity⊷CDLotBuyer.BuyCDLot
Department.ActivityTxn⊷
CDLotBuyer.BuyCDLotTxn
Department.Money ⊷CDLotBuyer.Money
Accounting.AttrRels.credit = undef (no credit for buying CD Lot activity)
Department.AttrRels.income = undef (no income for buying CD activity)

The result of this synthesis would be the "Exchange CD Lot against Money and Make Financial Report" role model. This is the synthesized model based on the "Merging Base Roles" implementation pattern (the "Extending Base Roles" implementation pattern is not considered in this example). In a similar way we include the accounting activity in "Exchange CD against Money" (see figure 4.b).

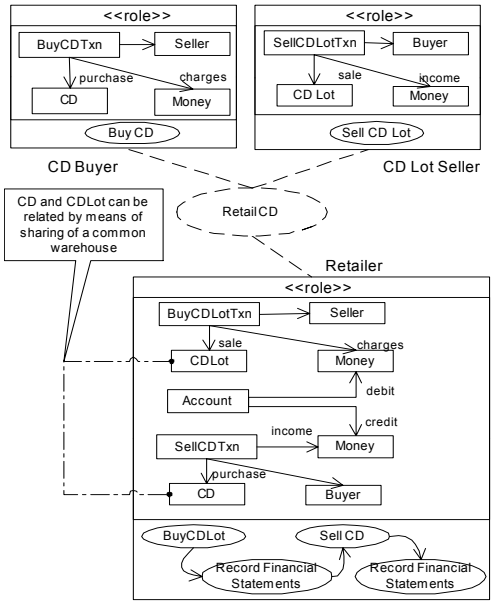

Figure 5: "Retail CD" role model.

In the second phase we synthesize two resulting role models by means of sharing common accounting activities (see figure 4.c). We have to specify synthesis constraints in a way that the Supply Accounting role and the Sales Accounting role share the same account. We also have to guarantee that the Report action (that follows the BuyCDLot action) precedes the SellCD action. To specify this we use the following synthesis constraints:

SupplyAccounting.SupplyAccount•–•
SalesAccounting.SaleAccount,
SupplyDepartment.A.Report→
SalesDepartment.A.SellCD

We show the result of the synthesis in figure 5 that represents the implementation for this role model synthesis specification based on the "merging base roles" implementation pattern. The four roles (Supply Department, Sales Department, Supply Accounting and Sales Accounting) from figure 4.c become one role: Retailer.

Note that in order to define the complete "Retail CD" role model, the "CD Seller" and "CD Buyer" roles have to share not only the common account but also a common warehouse for stocking CDs. The modeling of a common warehouse is done exactly the same way and thus we will not show it in our work.

3.2 Role Model Synthesis: Outlined Specification

In the previous subsection we considered how a business process can be specified from the set of base role models. The goal of the synthesis was clearly defined: to specify the "Retail CD" role model. However we can not always synthesize role models univocally. Role models can be synthesized in different ways that can result in different business processes. The important question is: which roles can be synthesized and how? This decision should be taken by a business analyst based on specifications of the base role models. The experienced business analyst has to rapidly search for meaningful mappings of the model elements in the specifications of roles. Meaningful means that a new synthesized role model does not contain contradictions and makes sense form the business point of view.

To illustrate how the set of base roles can be used to discover business process models, we have used an example from the audio-streaming industry. In this example we assume that readers are generally familiar with this industry and we do not show detailed specifications of each role. We believe that the names of roles are self-explanatory. The set of base role models was identified by means of analysis of the mp3.com business model (see figure 6).

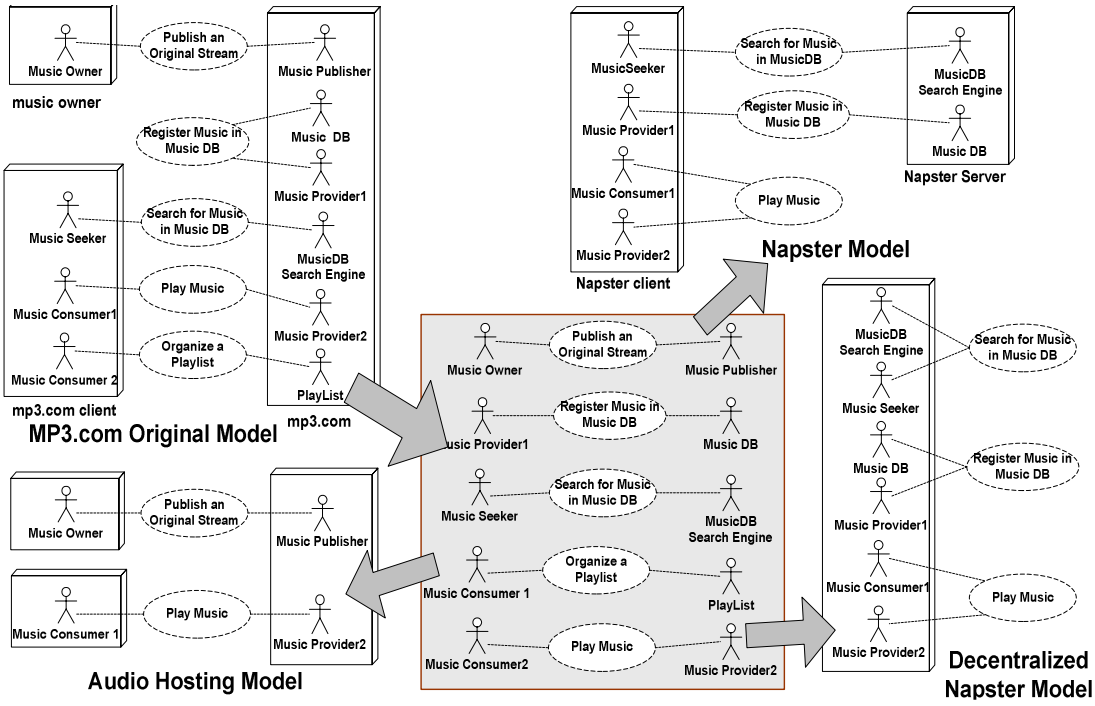

Figure 6: Original MP3.com model and different business processes that can be derived from it.

The original mp3.com model is shown in the upper left corner of figure 6. The set of base role models used to specify the mp3.com model is shown in the rectangle in the center of the figure. Around this rectangle we show business processes that can be derived from the set of based role models by means of assigning roles differently to business objects. Here we do not show how roles are synthesized in the context of business objects. Instead we appeal to our reader's intuition about how synthesis should be specified with synthesis constraints.

Role models in figure 6 can be used as a basis to define new business solutions (like outsourcing solutions). However these role models are not sufficient to define business models. Some additional decisions should be taken by a business analyst. For example, the Napster model from figure 6 does not have a customer (one that purchases a commodities or services). Thus a business analyst has to define a customer and define how the Napster server earns money. Several solutions are possible. For example: collecting money from Napster clients; making advertisements to Napster clients and collecting money from advertisement's providers. In the actual Napster business model, the Napster server earned money from music owners that were disappointed working with major record companies (like Columbia or Sony). These music owners sponsored the free distribution of music done by Napster to make their protest against big record companies (Gauze, 2002).

4 DISCUSSION AND FUTURE WORK

The idea about restructuring the elements of a business process in order to get a new one (probably better process) is not new. In 1994 Davenport & Stoddard identified seven reengineering myths. One of them is *the Myth of Reengineering Novelty*: "Reengineering, although about familiar concepts, is new in that these concepts are combined in a new synthesis. These key components have never been together before (Malhotra, 1998)". This shows that synthesis of independent concepts is one the most important operations in BPR. Different models of synthesis have been proposed by (Riehle, 1998), (Fiadeiro, 2001), (Reenskaug, 1996), (Bernstein, 1999). Some of them consider synthesis at a very detailed level where the semantics of synthesis becomes hidden behind technical details (like message passing); others do not consider the explicit modeling of state that makes the goal modeling impossible.

In our approach we propose using synthesis constraints to specify synthesis independently from implementation and allow for mapping structural and behavioral model elements. We believe that these two main features of our method can accelerate the work of a business analyst and improve the comprehensibility of business process models. To validate our approach we are planning in future to build metrics that can be used to compare existing synthesis methods, like OOram (Reenskaug, 1996), with our approach.

5 CONCLUSION

In this paper we addressed issues and questions related to the synthesis of business process models. To specify business processes we used role modeling technique. Role modeling allows us considering separately different concerns of a system. The main contribution of this work is the definition of a new concept of *synthesis constraints* that can be used for the specification of the synthesis of concerns. Synthesis constraints allow a business analyst to make explicit his decisions about how the synthesis is done. Making synthesis constraints explicit between behavioral and state model elements distinguishes this work from similar works.

We considered two implementation patterns for a role model with synthesis constraints: "Merging Base Roles" where one business object plays a synthesized role and "Extending Base Role" where several business objects play a synthesized role.

Our approach for the specification of business systems based on synthesis of role models has strong practical impact. Explicit design decisions on the synthesis of base roles, specified with synthesis constraints, allow a business analyst to disassemble and reassemble roles in many different ways and thus create new business solutions. We considered an example that shows how base roles can be synthesized in order to create a new business process model. Note that this process can not be automatic. It has to be accompanied by a business analyst who has to make decisions on how the synthesis of base role models should be done. Our approach can facilitate the work of a business analyst by means of using a BPR case tool that can be built based on role based PMT proposed in our work.

REFERENCES

Balabko, P., Wegmann, A., (2002). *The Role of Synthesis Constraints in Role Modeling*, Retrieved from EPFL, IC Web site: http://ic2.epfl.ch/publications/list.php

Balabko, P., Wegmann, A., (2001). From RM-ODP to the formal behavior representation, *Tenth OOPSLA workshop on Behavioral Semantics*, September 2001

Berger, J., (2002). The value of outsourcing. Whitepaper, *DecisionOne Corporation*, retrieved from: http://www.decisionone.com/d1m/news/white_papers

Bernstein, A., Klein, M., & Malone, T.W., (1999). The Process Recombinator: a Tool for Generating New Business Process Ideas. *International Conference on Information Systems*

Carlsen, S., (1997). Conceptual Modeling and Composition of Flexible Workflow Models. *Norwegian University of Science and Technology*, PhD Thesis

Davenport, T.H., & Short, J.E., (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*

Fiadeiro, J., Lopes, A., & Wermelinger, M., (2001). A Mathematical Semantics of Architectural Connectors. *ATXSOFTWARE*, retrieved from: http://www.atxsoftware.com/publications.html

Gauze, C. F., (2002, April). Fanning unveils failed Napster business model! *Ink 19,* Retrieved from http://www.ink19.com/issues/april2002/features/

Genilloud, G., & Wegmann. A., (2000). A Foundation for the Concept of Role in Object Modeling, *EDOC*.

ISO/IEC, (1996). 10746-1, 3,4 | ITU-T Recommendation X.902, Open Distributed Processing - Basic Reference Model - Part 2: Foundations. *ISO*

Kueng, P., et al. (1996). How to compose an Object-Oriented Business Process Model? *IFIP Working Conference on Method Engineering*

Malhotra, Y., (1998). Business Process Redesign: An Overview, *IEEE Engineering Management Review*, vol. 26, no. 3

Miers, D., (2001). Modern Business Strategies and Process Support, white paper, *Enix Consulting Ltd, England*

Ould M. A., (1995). Business Processes: Modeling and analysis for re-engineering and improvement, *John Wiley & Sons*

Phalp, K.T., et al. (1997). Using Counts as Heuristics for the Analysis of Static Models. *ICSE Workshop on Process Modeling and Empirical Studies of Software Engineering*.

Reenskaug, T., et al., (1996). *Working With Objects: The OOram Software Engineering Method*. ed: Manning Publication Co,

Regev, G. & Wegmann, A., (2001). Goals, Interpretations, and Policies in Information Systems Design. *EPFL-IC-LAMS*, Technical report No. DSC/2001/043, retrieved from http://icawww.epfl.ch/

Riehle, D. & Gross, T., (1998). Role Model Based Framework Design and Integration. *Conference on Object-Oriented Programming Systems, Languages, and Applications,* ACM Press

Stähler, P., (2002) Business Models as a Unit of Analysis for Strategizing. *International workshop on business models*, Lausanne, UNIL

Singh, B. & Rein, G.L., (1992). *Role Interaction Nets (RINs): A Process Description Formalism*, MCC

WMC, (1995). The Workflow Reference Model. *Workflow Management Coalition*, retrieved from: http://www.wfmc.org/standards/docs.htm.