Using Business Process Specifications and Agents

to Facilitate Supply Chain Integration

Hyunbo Cho*, Boonserm Kulvatunyou**, Hanil Jeong***, Albert Jones**
*Industrial Engineering Department, Pohang University of Science and Technology
Pohang 790-784, Korea
** MSI Division, National Institute of Standards & Technology
Gaithersburg, MD 20899, USA
*** Computer & Communications Engineering Division, Daejeon University
Daejeon 300-716, Korea

Abstract

In today's increasingly competitive global market, most enterprises place great stress on reducing order fulfillment costs, minimizing time-to-market, and maximizing product quality. The desire of businesses to achieve these goals has seen a shift from a make-to-stock paradigm to a make-to-order paradigm. The success of the make-to-order paradigm requires robust and efficient solutions to the supply-chain-integration problem in the business-to-business (B2B) environment. Recent Internet-based solutions to this problem have enabled efficient and effective information sharing among trading partners (customers, manufacturers, and suppliers). Here we present an integration framework for integrating supply chain operations among such trading partners. A supply chain scenario is constructed, for which an integration framework is proposed based on by concept of business process specifications (BPS) and technology of distributed agents. We model the BPS, which specifies message choreographies among the trading partners, using a modified Unified Modeling Language (UML). We model the behaviors of various enterprise applications within each trading partner as agents using Petri-nets, which depict how the enterprise applications respond to external events specified in the BPS. Finally, we argue that the concepts and models presented in this paper could provide the foundation for a structured approach to supply chain integration.

Keywords: Agents, Business Process Specification, Supply chain Integration, Supply chain Planning, Petri net Supply chain planning

1. Introduction

Making to stock and then shipping immediately upon receipt of customer orders is the most cost-effective approach to manufacturing high-volume, low-variety products. However, the life cycle of products is becoming shorter and products are manufactured increasingly according to specific customer specifications. The lead-time required to complete these customized products can be high due to a variable product mix and the uncertainty of process routings. To survive in this dynamic environment, manufacturing firms must change to a make-to-order rather than a make-to-stock paradigm, where design and manufacturing are not initiated until a customer order is received [7, 13]. Firms that have been successful in making this change typically do not hold finished inventory and are characterized by a functional shop layout with flexible production equipment [1]. In this new manufacturing paradigm, the ability to provide real-time order status and accurate delivery dates becomes imperative for both customer satisfaction and business growth.

The success of the make-to-order paradigm, particularly in emerging the Business-to-Business (B2B) environment, requires a highly integrated supply chain [8]. A supply chain is a network through which products, materials, information, and funds flow. That network contains suppliers, manufacturers, retailers, warehouses, and customers that implement a number of business processes [5]. Among the roles within this chain, the manufacturers must be able to manage their suppliers efficiently and effectively and control their own activities to meet customer orders in a timely fashion. The objective of this management and control is to maximize the overall value, which is defined to be the difference between the value to the customers and the cost incurred by filling the customers' requests.

Traditionally, the value of this maximum has been constrained by the sequential nature of supply chain operations and the inability to share both business and engineering information easily and concurrently [16]. Increasingly, supply chain performance depends on trading partners not only sharing information, but sharing it electronically over the Internet [11]. Over the last decade, massive investments in information technology have been made in the hope of meeting this requirement [22]. Electronic Data Interchange (EDI) [11], which enables information sharing between heterogeneous systems to support business transactions, was one of the first fruits of that investment. However, traditional EDI has several problems including fixed transaction sets, high fixed cost, and fixed business rules. Recent emergence of Internet-based, E-business technologies, which are described below, promise to address these problems, thereby enabling coordinated supply chain planning, scheduling, and operation. Chopra and Meindl [8] describe several companies that used these new technologies to achieve this coordination.

The Internet is the most important of these new technologies. It facilitates the physical transfer of business and manufacturing data quickly and cheaply across the entire supply chain [16]. The eXtensible Markup Language (XML) (http://www.w3.org/xml/) provides a standard way of representing that data on computer system that has software needed to encode and decode the data. Several organizations have proposed XML-based specifications aimed at standardizing the structure, content, and choreography of E-business transactions that use that data. A number of vendors have built commercial products that claim to implement these specifications. The National Institute of Standards and Technology has launched a project to tools and methods for testing conformance of these applications to thos specifications (www.mel.nist.gov/msid/oagnisttestbed). Nevertheless, the problem of integration with enterprise applications to support the B2B activities remains a challenge and an active area of research [20].

We view a business system that facilitates B2B integration as consisting of two functional components: a **private** and a **public** business component, as shown in Figure 1. The private business component, often referred to as a legacy system, includes data stores and enterprise applications (e.g., planning and scheduling modules). The public system, also called a business process executor, keeps track of business collaboration states and handles communications among trading partners. Adaptors (or interfaces) may be needed to exchange information between the private and public components.

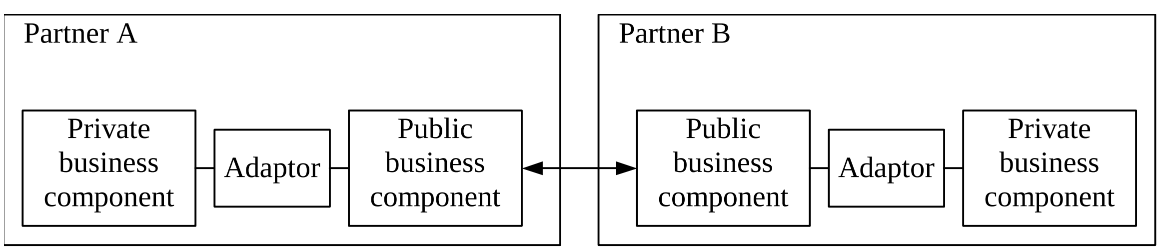

Figure 1. Relationship between private and public business components

In this paper, the private components of interest are those involved in integrated supply-chain planning; the public components are those involved in the exchange of business transactions needed to implement that planning. **FIX THIS!!!!** The detailed objectives are as follows. First, a supply-chain scenario is formed and its integration framework is proposed. Second, the behavior of the enterprise applications within each trading partner is modeled as agent concepts by using Petri-nets, in order to show how the applications respond to external events. The synchronization between two partners is also modeled by using Petri-nets. Third, BPSs and message choreographies among the trading partners are detailed by using a modified Unified Modeling Language (UML) (www.omg.org/uml/). This paper focuses on Request for Quote (RFQ) interactions; all other activities such as order fulfillment, contract validation, payment, and security, which are related to RFQ, are beyond the scope of this paper.

2. Related Work

2.1. Heterogeneous Supply Chain Integration Environments

In general, a supply chain includes many different trading partners. These trading partners will deploy different business practices and processes, and uses different means for exchanging information. Much of this information is exchanged electronically using a variety of communication protocols, message formats, security levels, and enterprise applications [4]. This situation requires each trading partner to implement a subset of the protocols and formats used by other trading partners in the chain. A number of B2B Integration framework have been proposed to address this situation [4,5].

2.2. B2B Integration Framework

Traditionally, business integration has been based on a set of specific translators to exchange information between a fixed set of software applications. To address the obvious deficiencies of this approach, B2B modeling constructs has been proposed to aid integration. These constructs, together with a process binding mechanism, help to transform business documents to and from different B2B protocols [4]. A message broker, which converts the formats of the messages exchanged across applications [5], can then be used to integrate the trading partners and enterprise applications. A methodology to integrate the internal information flows of the organization with these external B2B interactions has been proposed [20].

These flows and interactions have used with an agent-based architecture to select trading partners dynamically [6] and to coordinate production decisions across the supply chain [19]. These efforts rely on a Petri-net model for coordinating the distributed agents in this architecture. An agent-based approach, in conjunction with a workflow management system, has also been proposed to automate supply chain execution and monitoring [5, 14]. Typically, each agent fills the transactions extracted from the UML interaction diagram (www.omg.org/uml/).

In order to establish dynamic trading partners and execute transactions across heterogeneous organizations, a B2B protocol standard is required. Such a protocol would include a description of the syntax and semantics of the exchanged messages, message choreographies, business rules, transport mechanisms (e.g., HTTP, SMTP), and the security (e.g., 128-bit encryption) [3]. Examples of B2B protocol standards include RosettaNet (http://www.rosettanet.org) and ebXML (http://www.ebxml.org). Rosettanet is a standards initiative working on a broad set of supply chain scenarios, mainly focusing on the IT, electronic component and semiconductor industries. ebXML is a standardization effort established by UN/CEFACT and OASIS to provide an open XML-based infrastructure that can be employed in an interoperable, secure and consistent manner by all partners. RosettaNet predefines the interactions between business processes across enterprises called Partner Interface Processes (PIPs) whereas ebXML allows the construction of arbitrary business processes through trading partner agreement. In particular, the ebXML project aims to create a global electronic marketplace in which enterprises can find each other, agree to become trading partners, and conduct business.

2.3. Supply Chain Optimization

Recently, numerous researchers have used optimization techniques to study supply chain coordination and optimization [2] [12] [23] [22]. In particular, agent systems have been known for promising techniques of optimizing the entire supply chain as a networked system of independent trading partners, each of which optimizes its own local problems. A multi-agent approach has been applied to obtain control strategies for tactical decision-making beyond the boundaries of a single enterprise [12] [19].

3. Supply Chain Planning Scenario and Integration Framework

All supply chain operations begin with the arrival of an RFQ from each customer, which is followed by the manufacturer sending an RFQ to the suppliers to obtain the capacities and prices of the required materials and/or parts to respond to the customer with the quote, as shown in Figure 2. This paper describes the modeling issues associated with the manipulation of RFQs from a single manufacturer's viewpoint.

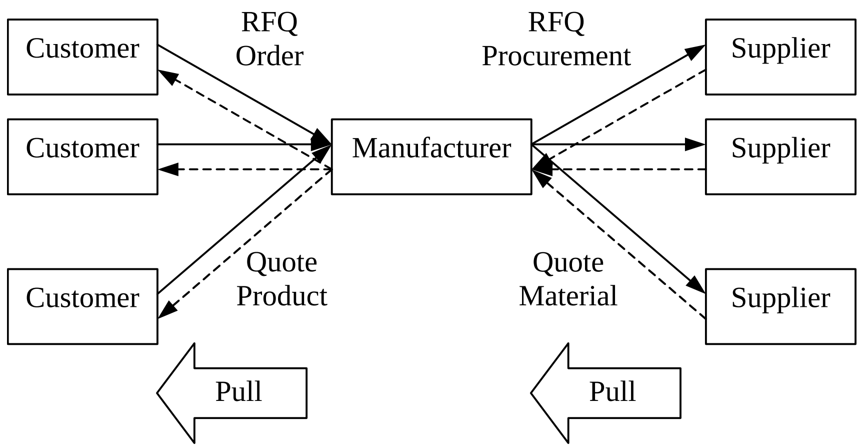

Figure 2. Three-stage SCP environment

3.1. Supply Chain Planning Process

A prototype transaction flow in the B2B SCP process is shown in Figure 3. The manufacturer receives an RFQ from each customer with information consisting of quantities, expected delivery dates, and the priorities. An issue arising in this situation is how many RFQs the manufacturer should wait for before their evaluation and response. This decision depends on the flexibility of the production facility, market demand, and the trade-off between retaining customers and maximizing profit. Immediately responding to RFQs may result in low production performance. After receiving a batch of RFQs, the manufacturer may perform material requirements planning and make-buy analysis based on resource availability and existing production schedule constraints. Based on the analysis results, suppliers for outsourced parts/materials must be discovered from the business registry (e.g., yellow page, UDDI web service registry (http://www.uddi.org)). The manufacturer interacts with the discovered suppliers to receive the quotes on their potential capacities, delivery dates, and prices. Using the information provided by customers and suppliers, the manufacturer solves SCP to select the suppliers that can provide the outsourced parts/materials with respect to manufacturer's performance and profit goals.

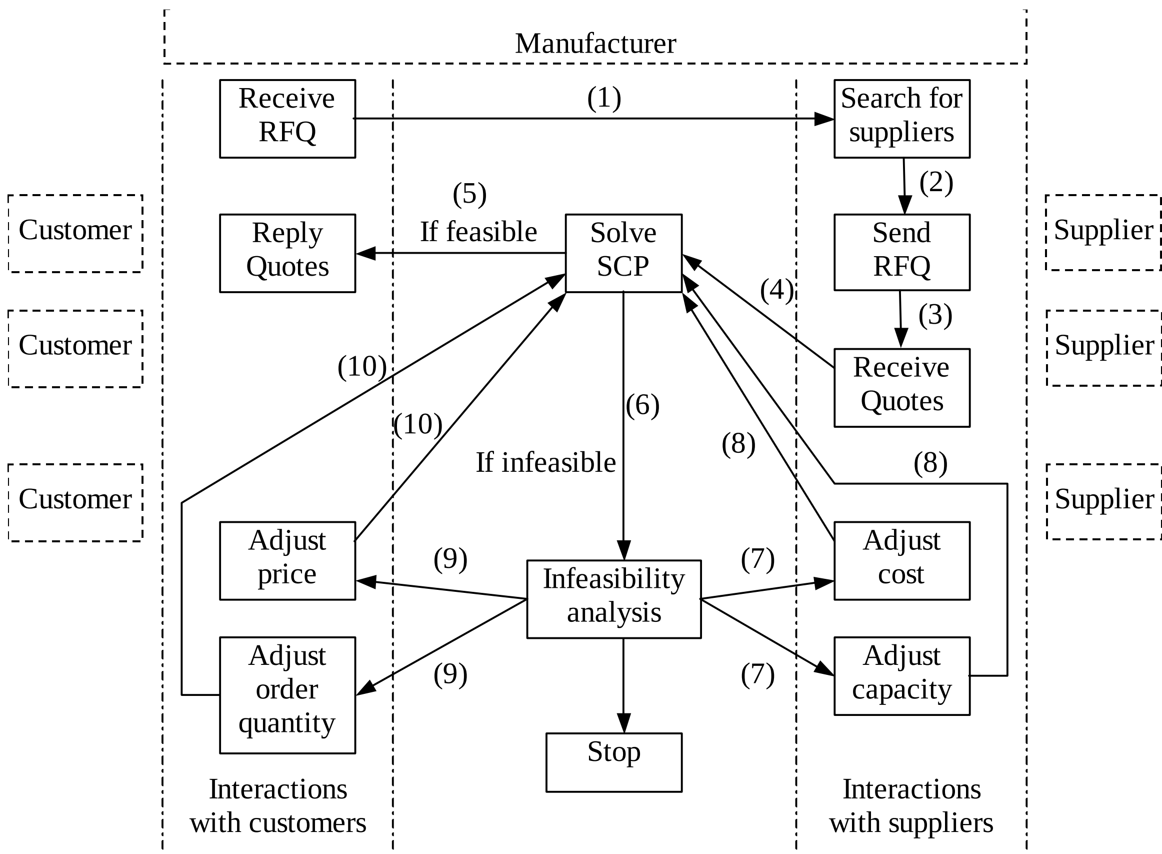

Figure 3. Work flows among the trading partners in the SCP environment

If the quantities and due-dates requested by the customers cannot be met (which can be determined based on the SCP solution), the manufacturer may negotiate with the suppliers to improve their capacities and/or due-dates. The suppliers may then want to raise the supply prices, which may be accepted by the manufacturer after solving the SCP problem again. Alternatively, the manufacturer may negotiate with customers to adjust requested quantities and/or due-dates according to the results of an infeasibility analysis. At this stage, some customers and/or suppliers dissatisfied with the adjustments may drop off.

If the manufacturer's profits are less than predetermined thresholds, the customers may be quoted with a higher price, or, alternatively, the suppliers may be asked to reduce the supply prices for the materials/parts. The manufacturer may suggest preliminary prices according to the results of an infeasibility analysis. During this process, the manufacturer's negotiations continue running in a cyclical manner: solving SCP, analyzing infeasibility, adjusting the manufacturer's capacity and/or profit, adjusting the customer's capacity and/or price, and adjusting the supplier's capacity and/or price.

3.2. Integration Framework

In order to perform B2B electronic transactions among trading partners, each partner should publish its business profile (e.g., types of products/materials it supports, messages exchanged, transport mechanism for the messages, etc.) in an open registry for other partners to discover and retrieve. When a company discovers new trading partners, all of the partners negotiate and agree on a trading partner agreement (e.g., business contracts and message exchanges) before conducting any business. The detailed procedures for trading partners discovery and agreement, which can be performed either automatically or manually, are beyond the scope of this paper.

In order to integrate and implement the proposed SCP in the B2B environment, agent technologies and business process specifications are employed as shown in Figure 4. Various enterprise applications within each trading partner are coordinated and activated by the agent to respond to the business transactions among the trading partners. In other words, each enterprise application is controlled via an agent, whose behavior is modeled using a Petri-net. The synchronization properties of Petri-nets enable the modeling and validation of interactions between trading partners before the actual implementation begins. These are detailed in Chapter 4. Business transactions and choreographies among the trading partners are modeled using a modified UML sequence diagram. This is detailed in Chapter 5.

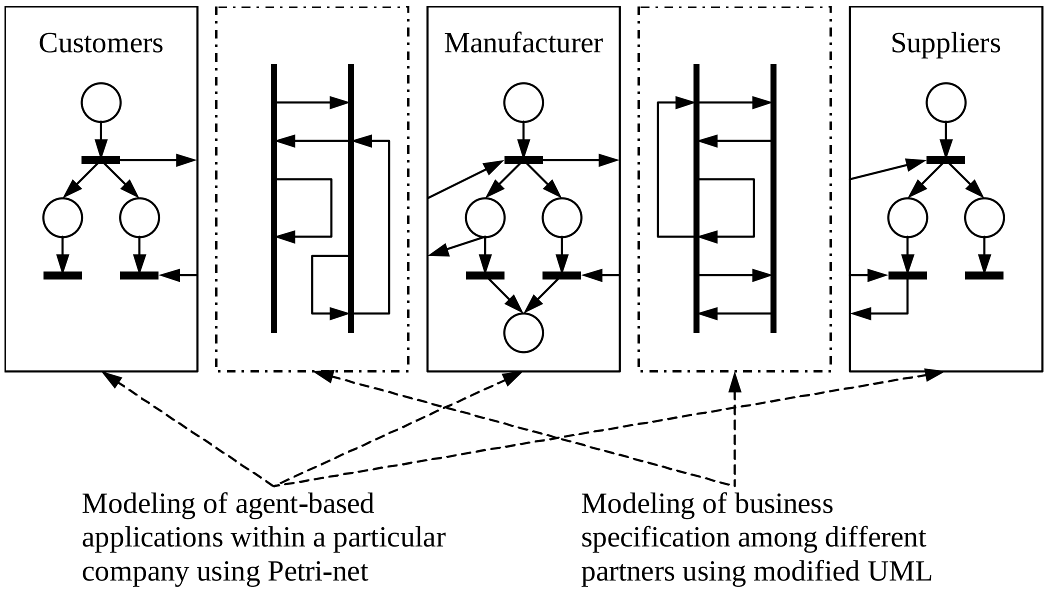

Figure 4. Implementation framework for integration of SCP

4. Behavioral Modeling of Agents using Petri-nets

An agent-based collaboration framework among trading partners is illustrated in Figure 5. The control agent of each partner coordinates the collaborations via interfaces with the enterprise applications. The control agent of each customer first dispatches a discovery agent to find manufacturers whose manufacturing capabilities match the product requirements. The control agent also dispatches an execution agent to construct a trading partner agreement and then starts a business collaboration by sending RFQs to the discovered manufacturers. It is noted once again that the supply chain activities described in this paper are defined and modeled from the viewpoint of a single manufacturer.

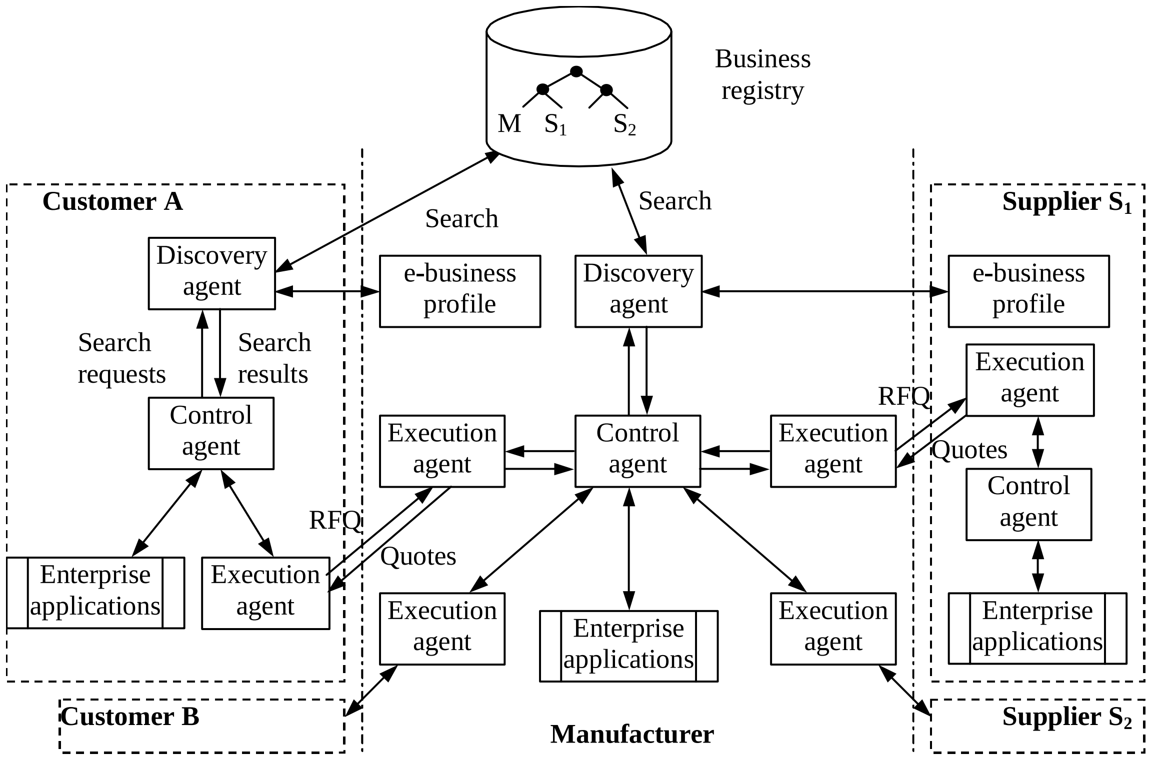

Figure 5. Agent-based collaboration framework for SCP

Once a batch of RFQs are received from the execution agents of the customers, the control agent of the manufacturer commands the discovery agent to search for suppliers that can provide the materials/parts required to satisfy the customers' RFQs. The discovery agent of the manufacturer uses e-business profiles of suppliers to accomplish the search. The control agent of the manufacturer then interacts with its enterprise applications to make decisions and generate RFQs for the discovered suppliers. Based on the quotes and delivery dates from the suppliers, the execution agents of the manufacturer may negotiate with the execution agents of the customers and suppliers to optimize the number of satisfied customers and the profits of the manufacturer.

4.1. Modeling of Task Allocation and Distribution

Tasks requiring more specialized knowledge (e.g., negotiation of price or order quantity) can be decomposed into subtasks which can then be allocated to skilled execution agents based on their cognitive and communication skills as well as environmental constraints. The allocation of tasks is processed in a hierarchical manner. The control agent acts as a supervisor that coordinates enterprise applications as well as governs execution agents. The behavior of the control agent can be modeled and visualized by using a Petri-net, a tool that has been previously used in applications involving concurrency, synchronization, and resource sharing [15].

A Petri-net is defined as a direct graph consisting of places, transitions, and arcs, where a circle represents a place, a bar represents a transition, and an arc links a place to a transition or a transition to a place. A marking indicates the distribution of an integer number (zero or positive) of tokens in every place. The movements of tokens from place to place represent the dynamic aspects of processes. A transition is enabled only if each of the input places towards the transition contains at least one token. Firing a transition by removing a token from each input place results in adding a token at each output place. When alternative firing transitions exist, a rule can be applied to choose which of them is to be fired first.

As a formal definition, a Petri net is four-tuple $PN=(P, T, I, O)$ where

1) P is the finite set of places, i.e., $P=\{p_1, p_2, ..., p_n\}$ for $n>0$,

2) T is the finite set of transitions, i.e., $T=\{t_1, t_2, ..., t_m\}$ for $m>0$,

3) $P \cap T = \phi$,

4) I: $P \times T \rightarrow \{0, 1\}$ is the input function that specifies the arcs directed from places to transitions,

5) O: $P \times T \rightarrow \{0, 1\}$ is the output function that specifies the arcs directed from transitions to places.

Among the various types of Petri-nets, an interpreted Petri-net model can be used to enable systems to be synchronized with external events [9]. It has the following three characteristics: 1) transitions can be synchronized with external events; 2) the tokens at a place may remain there for a certain time; and 3) places can contain a set of conditions, which are modified by associated actions. The modification of conditions can also contribute to transitions. A transition is fired if it is enabled, that is, when conditions are set to true and associated external events occur.

The coordination and allocation of tasks to the execution agents can be modeled using a type of interpreted Petri-net. A place contains the waiting conditions for completion of the execution agent’s negotiation and enterprise application’s execution. A transition represents occurrences of external events flowing along the business process specifications (e.g., failure of the search for suppliers, success of negotiation). For example, a token stays in a place when a negotiation process is in progress. When a negotiation process associated with the place has been finished, the token in the place will be transitioned with respect to the results of the negotiation.

An interpreted Petri-net model for specifying the behavior of a control agent of the manufacturer is illustrated in Figure 6. Detailed descriptions of the actions in the places and events in the transitions are given in Table 1. Initially, one token is deposited in place O_1 and another in O_2. The place O_1 represents the waiting condition for receiving RFQs from the execution agents of customers. The place O_2 represents a RFQ batching condition.

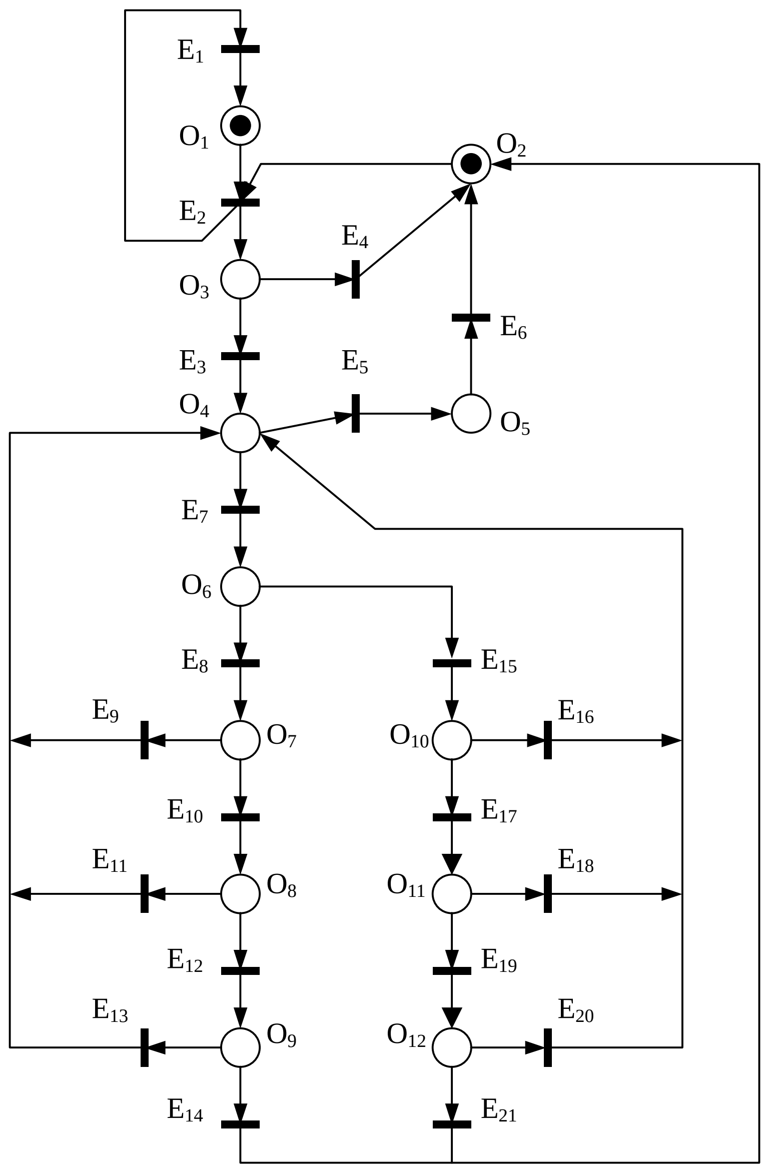

Figure 6. Interpreted Petri-net model for the behavior of a control agent of the manufacturer

Table 1. Actions and events for the interpreted Petri net of a control agent

Action	Description of an action in the place	Event	Description of an event in the transition
O_1	Receiving RFQs from customers (by agents)	E_1	Order arrival
O_2	Ready for processing RFQs	E_2	Processing start
O_3	Searching for suppliers (by an agent)	E_3	Searching success
O_4	Solving a SCP problem (by an enterprise application)	E_4	Searching fail
		E_5	Feasible SCP
O_5	Sending the quotes back to the customers (by agents)	E_6	RFQs replied
		E_7	Infeasible SCP
O_6	Analyzing infeasibility (by an enterprise application or manually)	E_8	Infeasible customer
		E_9	Capacity change
O_7	Modifying manufacturer’s capacity (by an enterprise application or manually)	E_{10}	No capacity change
		E_{11}	Capacity change accepted
O_8	Negotiating capacity with suppliers (by agents)	E_{12}	Capacity change rejected
O_9	Negotiating quantity with customers (by agents)	E_{13}	Quantity change accepted
O_{10}	Modifying manufacturer’s profit (by an enterprise application or manually)	E_{14}	Quantity change rejected
		E_{15}	Infeasible profit
O_{11}	Negotiating price with suppliers (by agents)	E_{16}	Profit change
O_{12}	Negotiating price with customers (by agents)	E_{17}	No profit change
		E_{18}	Price change accepted
		E_{19}	Price change rejected
		E_{20}	Price change accepted
		E_{21}	Price change rejected

When the transition E_2 is fired, one token is transitioned back to the place O_1 for collecting another batch of RFQs and the other is transitioned to the place O_3 in order to search for suppliers. If no supplier is found, the transition E_4 is enabled and the cycle ends without fulfilling the customers’ RFQs. If one or more suppliers are found, the transition E_3 is fired. The token is then deposited to the place O_4, implying that a SCP problem is solving. If there exists a feasible solution satisfying all customers, the transition E_5 is enabled to deposit a token the place O_5, implying that the fulfilled RFQs are sent back to the customers.

If some customers are dissatisfied with the solution, the transition E_7 is enabled and fired to deposit the token in the place O_6 to perform an infeasibility analysis. Satisfied customers may be excluded at this stage. Either the transition E_8 or E_{15} is fired based on the analysis results and the manufacturer’s planning strategy. If the infeasibility derives from a capacity shortage, the transition E_8 is fired and then the place O_7 is activated with a token to determine whether the manufacturer can improve capacity. If the manufacturer's capacity cannot be improved, the token is deposited in the place O_8 to invoke the execution agents for a capacity negotiation with the suppliers. Depending on the negotiation status, the manufacturer may negotiate with the customers to decrease the order quantity. The places O_8, O_9, O_{11}, and O_{12} are related to the waiting conditions of the execution agents' negotiation.

4.2. Modeling of Collaboration of Execution Agents

On receipt of tasks allocated by the control agent, the execution agents of two trading partners collaborate their interactions to accomplish the tasks. The execution agent of one trading partner makes progress based on information provided by the execution agent of the other partner. The interactions of the execution agents are synchronized in order to guarantee some degree of coherence and to prevent interference between the two trading partners. The sequencing of interactions must be articulated to achieve appropriate synchronization.

The interpreted Petri-net for representing synchronization between customer and manufacturer with respect to cost negotiation is illustrated in Figure 7. The actions and events in this Petri-net are listed in Table 2. Tokens are initially located at the places O_2 and O_{13}, which represent the waiting conditions of the agents before the negotiation process commences. On receipt of a token to the place O_1 from the control agent, the execution agent of the manufacturer invokes the control agent of the customer and then moves the token to the place O_3 to wait for a message from the execution agent of the customer. In other words, the place O_7 initiates the collaboration process by sending an RFQ to the control agent of the customer. After a token is placed at the place O_{11} (implying that the execution agent of the customer is invoked by the control agent), the transition E_6 is enabled and fired. The token stays temporarily in the place O_{12} while the execution agent determines the response to the query. The response may be to reject the query by sending a token through the place O_8, to accept the query by sending a token through the place O_9, or to accept with a modified cost by sending a token through the place O_{10}. The execution agent of the manufacturer sends a message to its control agent based on the response from the execution agent of the customer. It is noted that a timeout mechanism can be inserted, but its detailed representation is shown in a UML sequence diagram.

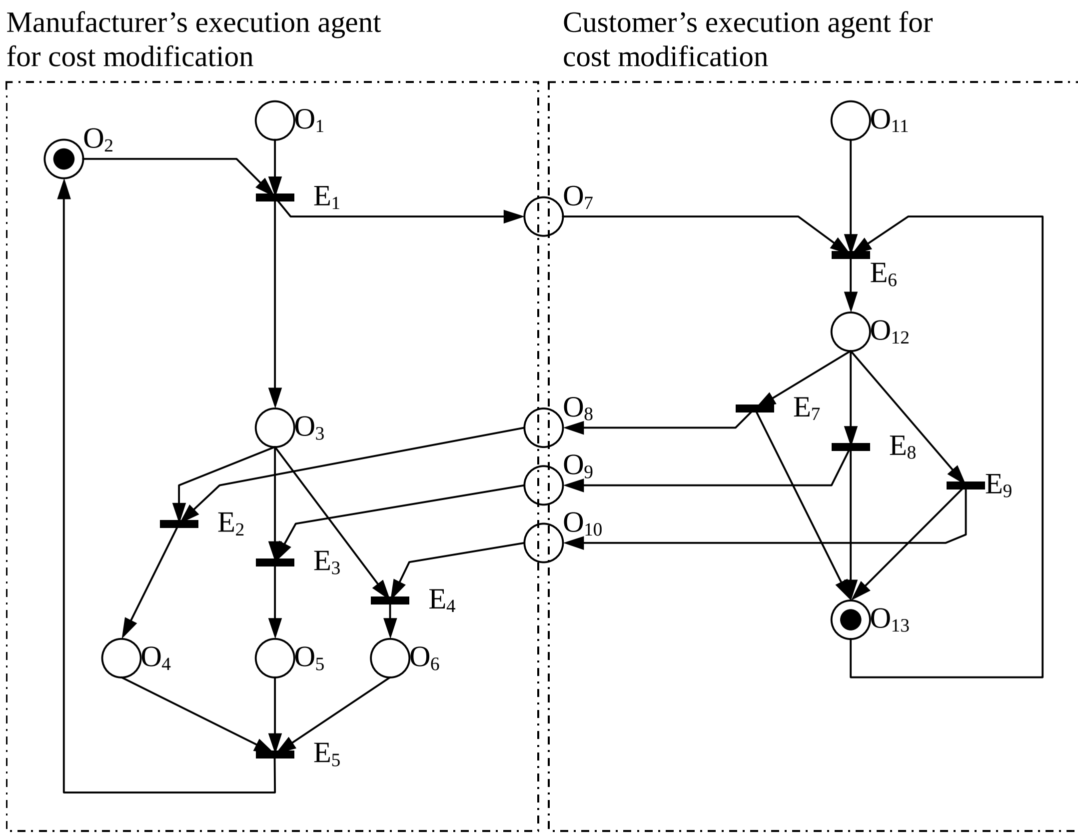

Figure 7. Interpreted Petri-net model for collaboration of cost negotiation

Table 2. Actions and events for the interpreted Petri-net of two execution agents

Action	Description of action	Event	Description of event
O_1	Receiving an invocation message	E_1	Negotiation initialization
O_2	Ready for negotiation	E_2	Negotiation failure
O_3	Waiting for a message	E_3	Negotiation success
O_4	Sending a failure message	E_4	Modified cost success
O_5	Sending a success message	E_5	Negotiation over
O_6	Sending a success message with modified cost	E_6	Negotiation initialization
O_7	Invoking the partner’s control agent	E_7	Negotiation failure
O_8	Receiving a negotiation-failure message	E_8	Negotiation success
O_9	Receiving a negotiation-success message	E_9	Modified capacity offer
O_{10}	Receiving a cost-modification message		
O_{11}	Receiving an invocation message		
O_{12}	Decision-making of cost modification		
O_{13}	Ready for negotiation		

Interpreted Petri-nets can be used to model all types of concurrent activities between the two trading partners. For example, the interpreted Petri-nets representing synchronization between any two trading partners with respect to capacity and price negotiation can be constructed using an approach similar to that shown in Figure 7, with only a few messages needing to be changed. In addition, there should also be differences in the related enterprise applications. When necessary, messages can be easily added to or deleted from the interpreted Petri net models. The formal modeling of agent collaboration using a Petri-net produces more clearly defined semantics and facilitates empirical and theoretical analysis before actual implementation.

5. Business Process Representation

A business system that facilitates B2B integration is viewed as consisting of a public and a private component as illustrated in the Introduction chapter. Subcomponents of the public component, such as the message security module, enable the cross-enterprise functionality of the enterprise application. Section 5.1 describes the business process specification, which specifies the public business process governing the public components of each trading partner. Section 5.2 then illustrates the connection between the public and private components described in Chapter 4.

5.1. Business Process Specification (BPS)

The public components of each business partner need to agree on a business process before they begin to collaborate (they must also agree on a communication mechanism, but that is beyond the scope of this paper.) A business process is described by a set of message choreographies (business states and transitions between them), which must be followed by the public components of each business partner to complete the collaboration.

A BPS represents a business collaboration, which prescribes a set of business activities between trading partners. Each partner's business system follows the message choreography and the business states defined in a BPS. This allows the business systems to keep track of the business state in the collaboration. This traceability is especially necessary in the B2B environment, in which collaborations occur over long periods. A business collaboration can be represented graphically as a sequence diagram. Each collaboration consists of a start state, a success state, a fail state, and at least one business state. The start state, success state, and fail state are shown in the sequence diagram with solid-filled, circular-hatched, and crosshatched circle, respectively. The business state is shown in a grayed box with its detail inside. Each business state has a request action and an optional response action, which are shown as solid lines labeled with messages. Each request and response action may require business signals to acknowledge the action status. Business signals are represented with dotted lines in the diagram.

Figure 8 uses a sequence diagram to graphically illustrate an exemplary business process of price negotiation collaboration between a manufacturer and a customer. In the requesting action, the manufacturer sends a Change Quote message, in which the offers (e.g., product item, unit price and promising date) are different from those in the previous quote. It should be noted that this price negotiation collaboration is a repeatable part of the larger RFQ collaboration. This requesting action requires that the customer send back an ‘acknowledgement of receipt’ signal and then an ‘acknowledgement of acceptance’ signal. The ‘acknowledgement of acceptance’ signal is not a business acceptance; it is simply an indication that the message is legible and can be processed.

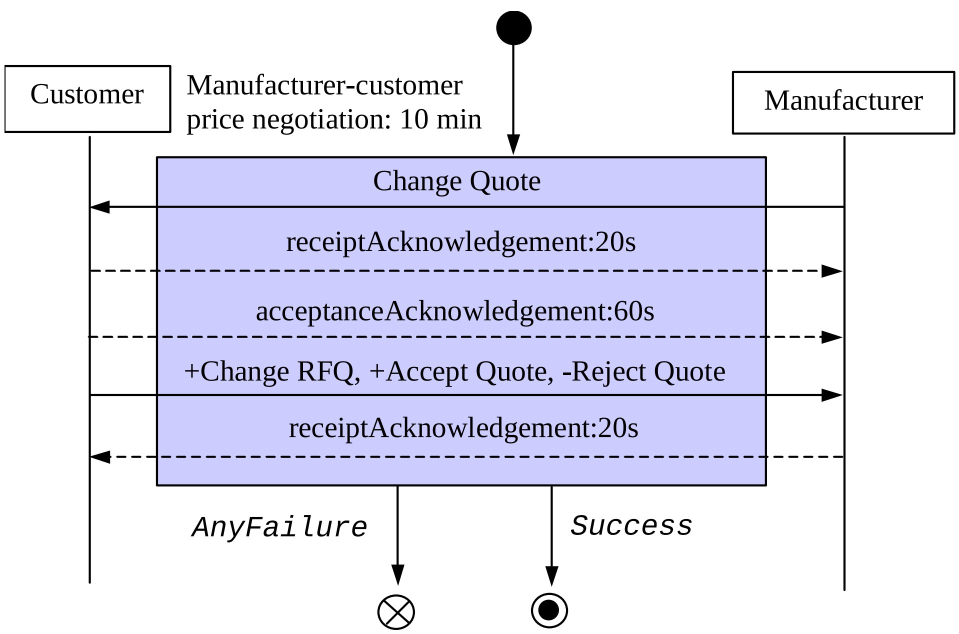

Figure 8. Manufacturer to customer price negotiation

The transaction requires that the customer reply to the manufacturer in the response action with a Change RFQ, Accept Quote, or Reject Quote message. The Change RFQ message is a positive response indicating that the customer would like to continue the negotiation with a counter offer (e.g., a slightly different price). The Accept Quote message is also a positive response indicating that the customer is satisfied with the offer from the manufacturer; in this case, the manufacturer should lock-in the quote for the customer. The Reject Quote indicates a negative response from the customer and results in ending of the collaboration. A negative response may arise in circumstances where the manufacturer's offer is too far from the result desired by the customer, or where the customer has a better offer from another manufacturer. In this collaboration, a positive response causes a transition into the success state and a negative response causes a transition into the failure state.

It should be noted that a number of timeouts apply throughout the collaboration to prevent either business partner from spending an infinite time in the waiting state. Additional legal and security parameters may also be included in the diagram to indicate the IT requirements for sending and saving business data in a secure manner as well as the requirements for business/legal traceability when a business argument arises; however, the discussion of these parameters are beyond the scope of this paper. A prototype XML encoding of the price negotiation is illustrated in Figure 9. It should be noted that the Accept Quote and Reject Quote messages point to the same business document specification; therefore, XML Path Language (XPath) expressions associated with the value of the AcknowledgeCode element are provided to enable the business application to differentiate the message semantics.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ProcessSpecification SYSTEM "http://www.ebXML.org/specs/ebBPSS.dtd">
<ProcessSpecification name="PriceNegotiation" version="1.0" uuid="">
    <BusinessDocument name="Change Quote" nameID="ALNBOLOAKNOIANUI"
    specificationLocation=http://www.openapplications.org/oagis/BODs/ChangeQuote.xsd
    specificationElement="ChangeQuote"/>
    <BusinessDocument name="Change RFQ" nameID="ZKLJUNHUKLIUIOUL"
    specificationLocation="http://www.openapplications.org/oagis/BODs/ChangeRequestForQuote.
    xsd" specificationElement="ChangeRequestForQuote"/>
    <BusinessDocument name="Accept Quote" nameID="UIOMCJUIOMCAUYMD"
    specificationLocation="http://www.postech.ac.kr/ie/oagis/BODs/AcknowledgeQuote.xsd"
    specificationElement="AcknowledgeQuote">
        <ConditionExpression expressionLanguage="XPATH" expression="AcknolwedgeCode =
        'Accepted'"/>
    </BusinessDocument>
    <BusinessDocument name="Reject Quote" nameID="QONBIOQUNDIJHDMJ"
    specificationLocation="http://www.postech.ac.kr/ie/oagis/BODs/AcknowledgeQuote.xsd"
```

```xml
specificationElement="AcknowledgeQuote">
    <ConditionExpression expressionLanguage="XPATH" expression="AcknolwedgeCode =
    'Rejected'"/>
</BusinessDocument>
<Package name="PriceNegotiation" nameID="FIEOANAONDAOKJAO">
    <BinaryCollaboration name="ManufacturerToCustomerPriceNegotiationBC"
    nameID="IUYIBIJGTDVHOPLN" timeToPerform="PT1H">
        <InitiatingRole name="Manufacturer" nameID="HBKJGJHGKKHYFCSR"/>
        <RespondingRole name="Customer" nameID="TYUBVRTTYVDROPMJ"/>
        <BusinessTransactionActivity name="ManufacturerToCustomerPriceNegotiationBTA"
        nameID="LKMHYBGFTRSDCXZA"
        businessTransaction="ManufacturerToCustomerPriceNegotiationBT"
        businessTransactionIDRef=" UJMNHYTGBVFREDCX " fromAuthorizedRole="Manufacturer"
        fromAuthorizedRoleIDRef=" HBKJGJHGKKHYFCSR" toAuthorizedRole="Customer"
        toAuthorizedRoleIDRef=" TYUBVRTTYVDROPMJ" isConcurrent="true"
        isLegallyBinding="false" timeToPerform="PT1H"/>
        <Start toBusinessState="ManufacturerToCustomerPriceNegotiationBTA"
        toBusinessStateIDRef=" LKMHYBGFTRSDCXZA"/>
        <Failure fromBusinessState="ManufacturerToCustomerPriceNegotiationBTA"
        fromBusinessStateIDRef=" LKMHYBGFTRSDCXZA" conditionGuard="AnyFailure"/>
        <Success fromBusinessState="ManufacturerToCustomerPriceNegotiationBTA"
        fromBusinessStateIDRef=" LKMHYBGFTRSDCXZA" conditionGuard="Success"/>
    </BinaryCollaboration>
    <BusinessTransaction name="ManufacturerToCustomerPriceNegotiationBT"
    nameID="UJMNHYTGBVFREDCX" isGuaranteedDeliveryRequired="true">
        <RequestingBusinessActivity name="Manufacturer" nameID="XCNJIONDUIEOABDJIU"
        isAuthorizationRequired="false" isIntelligibleCheckRequired="true"
        isNonRepudiationReceiptRequired="false" isNonRepudiationRequired="false"
        timeToAcknowledgeReceipt="PT60S" timeToAcknowledgeAcceptance="PT120S">
            <DocumentEnvelope isAuthenticated="false" isConfidential="false"
            isTamperProof="false" businessDocument="Change Quote" businessDocumentIDRef="
            ALNBOLOAKNOIANUI "/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="Customer" nameID="DANBUDIBDJBNEJDF"
        isAuthorizationRequired="false" isIntelligibleCheckRequired="true"
        isNonRepudiationReceiptRequired="true" isNonRepudiationRequired="false"
        timeToAcknowledgeReceipt="PT60S">
            <DocumentEnvelope isPositiveResponse="true" isAuthenticated="false"
            isConfidential="false" isTamperProof="false" businessDocument="Change RFQ"
            businessDocumentIDRef=" ZKLJUNHUKLIUIOUL"/>
```

```xml
                <DocumentEnvelope isPositiveResponse="true" isAuthenticated="false"
                isConfidential="false" isTamperProof="false" businessDocument="Accept Quote"
                businessDocumentIDRef=" UIOMCJUIOMCAUYMD"/>
                <DocumentEnvelope isPositiveResponse="false" isAuthenticated="false"
                isConfidential="false" isTamperProof="false" businessDocument="Reject PO"
                businessDocumentIDRef=" QONBIOQUNDIJHDMJ"/>
            </RespondingBusinessActivity>
        </BusinessTransaction>
    </Package>
</ProcessSpecification>
```

Figure 9. Prototype XML encoding for price negotiation

A sequence diagram for the capacity negotiation business process between a manufacturer and a supplier is illustrated in Figure 10. In this collaboration, the manufacturer, after an internal planning decision that is based on the initial quotation given by the supplier, negotiates with the supplier to change its plan. The manufacturer first sends a Change RFQ message to the supplier. The supplier may respond with a new quote, accept the change, or deny the change by responding with a Change Quote, Accept RFQ, or Reject RFQ message, respectively. The messages Change Quote and Accept RFQ are positive responses, and therefore end the collaboration with a Success state. On the other hand, the Reject RFQ message is a negative response that ends the collaboration with a Failure state. This collaboration is part of the complete RFQ collaboration. At the end of this collaboration, the transition within the RFQ collaboration may indicate that the manufacturer cannot negotiate capacity with the supplier again after the failure of this capacity negotiation.

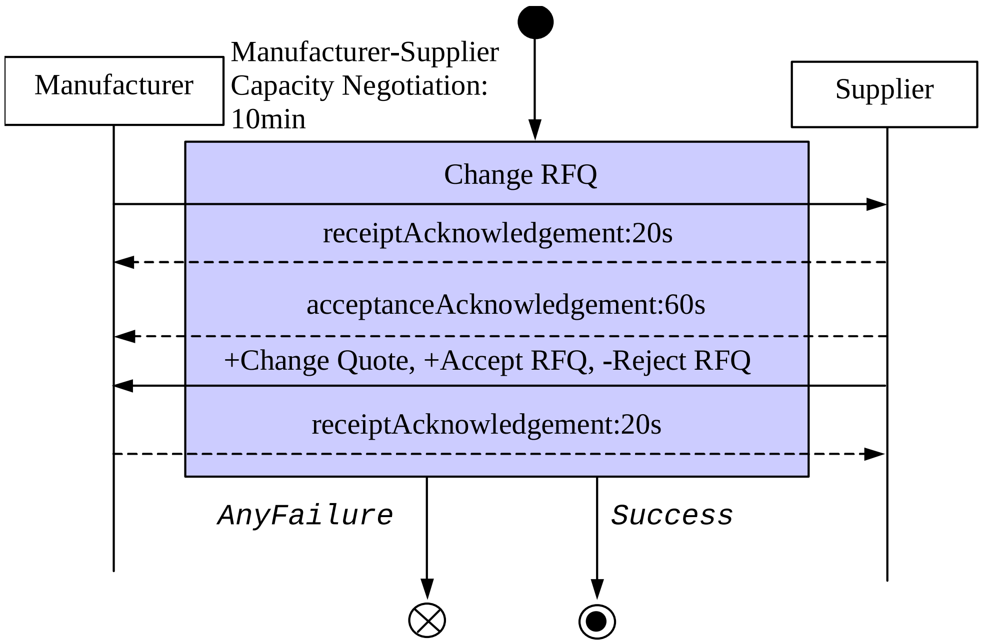

Figure 10. Manufacturer to supplier capacity negotiation

5.2. Connecting the Public and Private Business Components

This section illustrates an approach for connecting the public component's flow of events as specified by the BPS with the private component's flow of events using the petri-net model described in Chapter 4. The ability to connect these two components represents the business capability of the business system as indicated by the associated BPS.

In this distributed Petri-net model, the shared places must be distributed to each partner. Each shared place must be replaced by two places and a transition. Formally, the following modifications must be made to the original Petri-net model of the collaboration.

1) Three sets of places are defined as follows:

a) sP is a set of shared places, $sP \subset P$

b) sP' is a set of places that are split out from places in sP that are associated with the output functions, $O{:}sP \times T$, of the members of sP . Members of sP' are therefore representing message encoding and sending actions.

c) sP'' is a set places that are split out from places in sP that are associated with the input functions, $I{:}sP \times T$, of the members of sP. Members of SP'' are therefore representing message receiving and decoding actions.

The set P is then modified to $P = (P - sP) \cup sP' \cup sP''$

2) A new set of transitions is defined as follows:

sT is a set of transitions between members of sP' and sP''. Members of sT are therefore representing a set of message transmissions.

The set T is then modified to $T = T \cup sT$

3) Input and output functions

a) Input functions associated with the places in sP are input functions associated with the places in sP''. That is, $I{:}\ sP \times T$ is replaced with $I{:}\ sP'' \times T$.
b) Output functions associated with the places in sP are output functions associated with the places in sP'. That is, $O{:}sP \times T$ is replaced with $O{:}sP' \times T$.
c) Input functions of the transitions in sT must be added. That is, $I{:}\ sP' \times sT$ are added.
d) Output functions of the transitions in sT must be added. That is, $O{:}\ sP'' \times sT$ are added.

Figure 11 illustrates the resulting Petri-net model after applying the above modification to the agent negotiation model (Figure 7) in conjunction with the 'Manufacturer to customer price negotiation' BPS (Figure 8). The resulting model represents the flow of events connection between the private (the agent model) and the public (the business process) components. The places in sP' and sP'' become parts of each partners' Business Process Executor (BPE). It should be noted that the business signals and others functionalities (e.g., time out, non-repudiation) required by the BPS are handled entirely by the BPE, keeping the public and private component modular in their functionalities. The model may be interpreted as follows. Instead of directly triggering the event E_6, the place O'_7 represents the action of the public business component, which encodes and sends the Change Quote message to the place O''_7 in the public business component of the other partner. The place O''_7 must decode the message and then feed the information into the place O_{12} for decision-making. Similarly, places O'_8, O'_9, O'_{10} encode the associated messages, and places O''_8, O''_9, O''_{10}

decode the associated messages. Each node follows the message encoding formats and specifications defined in the BPS and communication protocol agreements.

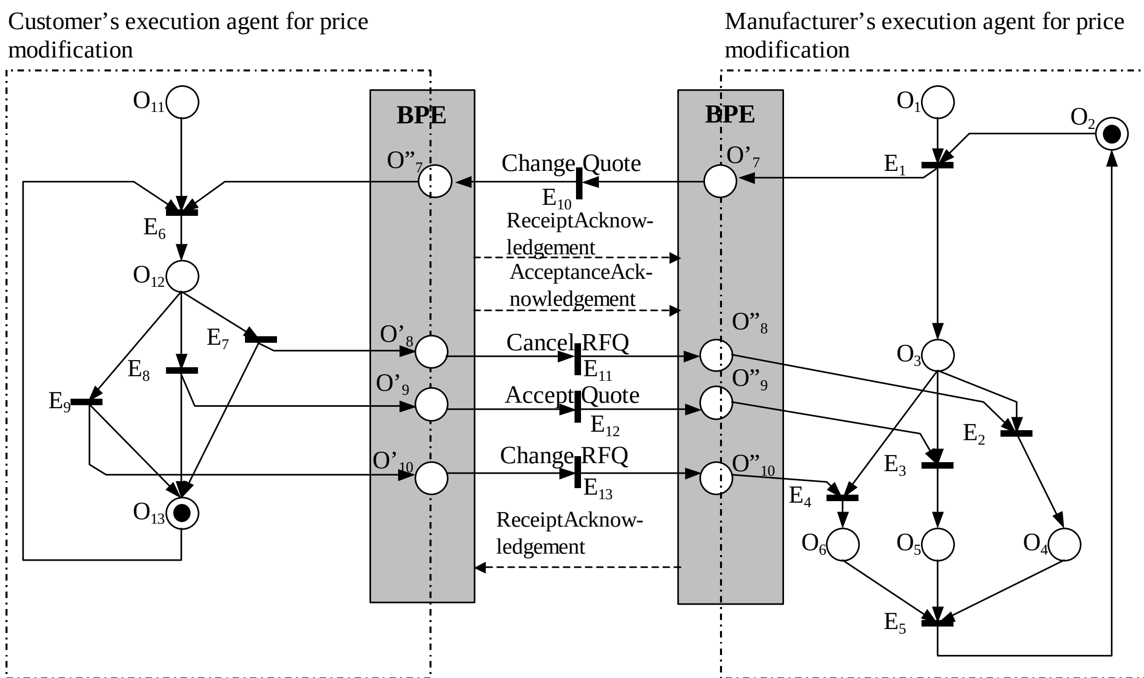

Figure 11. Connecting the Petri-Net model of the execution agent to the BPS

6. Conclusion

In this paper, we present an integration framework for the three-stage SCP model. A system framework for implementing a prototype supply chain scenario is constructed using the concepts of business process specifications and agents. The supply chain operation is supervised by the control agent whose behavior is modeled using an interpreted Petri-net. The control agent coordinates the back-end enterprise applications as well as dispatches the execution agents to conduct the business collaboration. A place contains the waiting conditions for completion of the execution agent's negotiation and enterprise application's execution. A transition represents occurrences of external events flowing along the business process specifications.

On receipt of tasks allocated by the control agent, the execution agents of the two trading partners interact to accomplish the tasks. Synchronization between the two trading partners is modeled using interpreted Petri-nets. The synchronization is required to guarantee some coherence and prevent interference between the two trading partners.

The BPS among the trading partners is modeled in the form of a modified UML sequence diagram. This diagram represents a business process, which embraces the choreography of business transactions between trading partners. The public component's flow of events specified by the BPS is expressed through the modification of the Petri-net model of agent synchronization.

The concepts and models proposed in this paper should provide the starting point for developing a structured approach to B2B supply chain integration and implementation. In the near future, we plan to implement the scenario described in the paper using technologies and software components available in the open literature. A public mechanism for registering and storing each partner's e-business profile will be used (e.g., the UDDI registry or ebXML registry). As the agent development platform, JADE (Java Agent DEvelopment Framework) will be employed, which is a software framework fully implemented in Java language to simplify the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA (Foundation for Intelligent Physical Agents) message specifications. If any logical inference is required, for example in the infeasibility analysis, the XSB system will be used. This system is an open source logic programming system that extends PROLOG with new semantic and functional features, mostly based on the use of Tabled Logic Programming or tabling.

Acknowledgement

The authors wish to acknowledge the partial financial support of the NIST BK21 project.

References

1. Babu, A. S., "Strategies for enhancing agility of make-to-order manufacturing systems," *International Journal of Agile Management Systems*, Vol. 1, No. 1, pp. 23-29, 1999.

2. Barbarosoglu, G., "An integrated supplier-buyer model for improving supply chain coordination," *Production Planning and Control*, Vol. 11, No. 8, pp. 732-741, 2000.

3. Bussler, C., "B2B protocol standards and their role in semantic B2B integration engines," *Bulletin of the Technical Committee on Data Engineering*, Vol. 24, No. 1, pp. 1-9, 2001.

4. Bussler, C., "Modeling and executing semantic B2B integration," *Proceedings of the 12th International Workshop on Research Issues in Data Engineering* (RIDE '02), pp. 69-74, San Jose, CA, February 25-26, 2002.

5. Casati, F., Dayal, U., and Shan, M., "E-business applications for SCM: Challenges and solutions," *Proceedings of the 17th International Conference on Data Engineering*, pp. 71-78, Heidelberg, Germany, April 2-6, 2001.

6. Chen, Y., *et al.*, "A negotiation-based multi-agent system for supply chain management," *Proceedings of the 3rd International Conference on Autonomous Agents,* pp. 15-20, Seattle, WA, May 1, 1999.

7. Cho, H., Jung, M., and Kim, M., "Enabling technologies of agile manufacturing and its related activities in Korea," *Computers and Industrial Engineering*, Vol. 30, No. 3, pp. 323-334, 1996.

8. Chopra, S. and Meindl, P., *Supply Chain Management: Strategy, Planning, and Operation*, Prentice Hall, Upper Saddle River, NJ, 2001.

9. David, R. and Alla, H., "Petri nets for modeling of dynamic systems-a survey", *Automatica*, Vol. 30, No. 2, pp. 175-202, 1994.

10. Ferber, J., *Multi-agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison Wesley, Essex, England, 1999.

11. Furst, K. and Schmidt, T., "Turbulent markets need flexible supply chain communication," *Production Planning and Control*, Vol. 12, No. 5, pp. 525-533, 2001.

12. Gjerdrum, J., Shah, N., and Papageorgiou, L. G., "A combined optimization and agent-based approach to supply chain modeling and performance assessment," *Production Planning and Control*, Vol. 12, No. 1, pp. 81-88, 2001.

13. Hendry, L. C., "Applying world class manufacturing to make-to-order companies: problems and solutions," *International Journal of Operations and Production Management*, Vol. 18, No. 11, pp. 1086-1100, 1998.

14. Huhns, M. N. and Stephens, L. M., "Automating supply chains," *IEEE Internet Computing*, Vol. 5, No. 4, pp. 90-93, 2001.

15. Jensen, K., "Colored Petri nets and the invariant-method", *Theoretical Computer Science*, Vol. 14, pp. 317-336, 1981.

16. Kehoe, D. and Boughton, N., "Internet based supply chain management: A classification of approaches to manufacturing planning and control," *International Journal of Operations and Production Management*, Vol. 21, No. 4, pp. 516-525, 2001.

17. McIlraith, S. A., Son, T. C., and Zeng, H., "The semantic web - semantic web services," *IEEE Intelligent Systems and Their Applications*, Vol. 16, No. 2, pp. 46-53, 2001.

18. Qinghe, H., Kumar, A., and Shuang, Z., "A bidding decision model in multiagent supply chain planning," *International Journal of Production Research*, Vol. 39, No. 15, pp. 3291-3301, 2001.

19. Sadeh, N. M., Hildum, D. W., Kjenstad, D., and Tseng, A., "MASCOT: an agent-based architecture for dynamic supply chain creation and coordination in the Internet economy," *Production Planning and Control*, Vol. 12, No. 3, pp. 212-223, 2001.

20. Sayal, M., Casati, F., Dayal, U., and Shan, M., "Integrating workflow management systems with business-to-business interaction standards," *Proceedings of the 18th International Conference on Data Engineering*, pp. 287-296, San Jose, CA, February 26-March 1, 2002.

21. Sundaram, M. and Shim, S. S. Y., "Infrastructure for B2B exchanges with RosettaNet," *Proceedings of the 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems* (WECWIS '2001), pp. 110-119, Milpitas, CA, June 21-22, 2001.

22. Tayur, S.,Ganeshar, R., and Magazine, M., *Quantitative Models for Supply Chain Management*, Kluwer Academic Publishers, Dordrecht, Holland, 1999.

23. Zhao, W. and Wang, Y. "Coordination of joint pricing-production decisions in a supply chain," *IIE Transactions*, Vol. 34, No. 8, pp. 701-715, 2002.