Abstract

Faced with the challenges of current e-business powered competitive environments, enterprises now find that it is not adequate to manage their own businesses but also the supply chain. They must be involved in managing the network of the upstream that provides the input, as well as the network of the downstream responsible for the output to the customer. Managing the supply chain has become one of the important issues in today’s e-business as the participants span across the network among which the processes and services are independently designed and configured. It is required that a rich variety of interactions must be made available to enable and support the information flows. While several network protocols are currently operating, however, a supply chain protocol owns its specific properties to the business operation that is not supported by the current network protocols. In this paper, we investigate the needs of the supply chain protocol, and propose a novel approach for designing the protocol that brings the business processes and services together to support the supply chain management.

1 Introduction

E-commerce, more generally e-business, involves multiple business entities spanning across the network that are based on a rich variety of interactions among software components of business processes and services. These software components often are independently designed and configured representing independent business interests, which require communication protocols for them that enable business integration and process inter-operation. A supply chain can be regarded as such a business integration that encompasses all organisations and activities associated with the flow and transformation of goods from the raw material stage, through to the end user, as well as the associated information flows. Under a modern dynamic e-business environment, the supply chain has shifted the views from that the processes among participants are considered as rigorous and closely coupled task flows, to that of distributed, flexible and complex network processes. These views require a notion that could deal with the complexity in business process engineering.

Web services currently provide a basis for realising such processes by enabling businesses to inter-operate in a standardised manner. However, Web services are at a lower level of abstraction that is not adequate for encoding all business scenarios. This has been leading to the interests in technologies such as coordination, conversation, orchestration and choreography. These approaches have given the benefits for the supply chain management, but suffer from the shortcomings that a centralised perspective of processes is viewed. It needs a reliable modelling method and a rigid enactment, which result in the tremendous difficulties for the supply chain management. The needs of protocols are apparent in order to bring distributed business processes and services together. This paper relates to all of these efforts, but focuses on supply chain protocols that are viewed as distributed process and service integration.

The supply chain is not only about the material and information flows, but business process integration and supply chain management through cooperative organisational relationships, effective business processes and high level information sharing to create highly performing value systems. The effective supply chain management requires that the contents of interactions must be made clear to supply chain participants before the operation of the supply chain could start off. For example, the automobile industry would have a variety of interaction protocols that support various aspects of its supply chain network. However, such process protocols are restricted to the industrial specification, not generally available to others without operational commitments. With the advanced Internet technologies, it is
inevitable that the dynamic characteristics of supply chain will become the bottleneck for its management, which requires flexible protocols that reflect the issues of its management, including business and operation. Protocols would be needed for every business aspect of interaction within a supply chain, for example, negotiation, production, product ordering, payment, only named a few business scenarios, through two or more participants’ interaction. Clearly, developing sophisticated protocols would not be a trivial task. However, by recognising these scenarios, it is possible that a high level of abstraction can be captured, further used for designing the protocols. Our approach is based on the motivation of business scenarios in practice, that simply define the protocols of the supply chain as the specification of a logically related set of interaction processes characterised by business scenarios. It reflects supply chain management requirements in addition to the algebraical interaction mechanisms.

This paper is organised as follows. In Section 2, we briefly introduce the process algebra that defines interaction processes and their evolution as the technical background, and demonstrate the abstracted notation with some Internet purchasing examples. In Section 3, we discuss the semantics of the interaction processes and process commitments, and formally introduce our designing approach for protocolling that supports a natural structure of a supply chain. An example is given to show the steps of designing a protocol for a new interaction scenario. In Section 4, the relevant literature is reviewed and compared with our proposed protocolling approach. Section 5 gives our conclusion and further work related to the proposed protocolling approach.

2 Technical Background

We represent the supply chain protocols as interaction process series. The protocols generate computations that represent the sequences of interactions between the participants involved in a supply chain. These interactions could be thus regarded as process units that compose of the processes and services that the participants have offered. We outline a formal treatment of the interaction and interaction process based on process algebra, which gives the basis for the discussion of the protocol design in Section 3.

2.1 Process algebra notation

For formally representing the interaction process, we consider a mathematical formalism based on $\pi$ calculus, a type of process algebra with the notation of names and processes specified for the distributed and communication computation, that describes the communication behaviours of process systems. We introduce some basic notation of process algebra which is sufficient for the discussion of technical development in the rest of the paper.

Two primitive entities in $\pi$-calculus are a process, $P$, that is the unit of behaviour, and an action, $\pi$, that is an action prefix of a process. Two basic forms of the action prefix are $\alpha(x)$ and $\beta(y)$, which define a pair of complementary actions between processes in communication, where $x, y, \ldots \in X$ are names without a structure and $\alpha, \beta, \ldots \in C$ are names channels, which is distinguished from the rest of the family of process algebras. The processes, $P, Q, \ldots \in P$, are built up from names by the syntactical form of

$$ P ::= \pi.P | 0 | P + Q | P||Q||P|!r(x)P $$

with the semantics of the process being defined as follows:

- $\alpha(y).P$: The process inputs/receives name $y$ on channel $\alpha$, then behaves as $P$.
- $\alpha(y).P$: The process outputs/sends name $y$ on channel $\alpha$, then behaves as $P$.
- $0$: The termination process that stops and does nothing.
- $P + Q$: The choice process behaves either like $P$ or $Q$.
- $P||Q$: The composition process that $P$ and $Q$ proceed concurrently and independently. The synchronisation can occur if the pair of complementary actions is taken between $P$ and $Q$.
- $!P$: The replication process is the copies of $P$, $P|P|P|\ldots$.
- $r(x)P$: The process behaves like $P$; but name $x$ is restricted by $r$ to $P$, which can not be used for communication.

In [5], a full treatment is given on process algebra. The algebra has been widely used as a formal formalism complementary to Petri Nets in distributed and communication systems, which is familiar to most of system modelers. Since the protocols for a supply chain are at a high level of abstraction of process modelling, the language of the algebra gives the advantage of more expressive power in modelling the behaviours of communication systems.

Several aspects of the algebra are worth to be mentioned for our discussion on supply chain protocols as follows:

1. Reduction relation: the only reduction relation $\rightarrow$ over processes, $P \rightarrow P'$, is used to define the transformation among the processes for a computation step, for example

$$ (\cdots + \bar{x}(\bar{y}).P)|(\cdots + x(\bar{z}).Q) \rightarrow P|Q\{\bar{z}/\bar{y}\} $$
means that process \(((\cdots + \vec{y}.P))((\cdots + x(\vec{z})).Q)\) is transformed into \(P[Q\{\vec{z}/\vec{y}\}]\) that \(\vec{z}\) is substituted by \(\vec{y}\) after the communication occurs, where \(\vec{y}, \vec{z}\) are of a equal length of the name vector. The reduction rules can be derived by process commutative and associative properties based on the reduction relation.

2. Mobility: it is an interesting property in process algebra that the mobility is achieved by name and process passing. A simple example of the process mobility is demonstrated by defining a process

\[
P|Q|R = \bar{\alpha}(\beta, x).P'|\alpha(x, y).\bar{y}(z).Q'|\beta(z_1).R'
\]

then after a communication step, the process is evolved into

\[
P'|\bar{\beta}(z).Q'|\beta(z_1).R'
\]

where the communication between processes \(Q\) and \(R\) is established though the original process defined does not have the linkage between \(Q\) and \(R\).

Since we are only interested in designing protocols at high level of abstraction, a fragment of the algebra will be sufficient for modelling the sophisticated interaction behaviours of a supply chain. The detailed processes in a supply chain will remain for our further investigation.

2.2 Process interaction

Much of our technical development is based on the established concepts of distributed systems, where the interaction between software components is inspired. Since the distributed computing is one of well established theories, we do not intend to present them exhaustively. Rather, a concept of process interaction is introduced that brings the relevant processes together for fulfilling a specified task.

**Definition:** An interaction process is an enactment process, denoted as \(P_I\), that defines at least two independent processes that interact via a channel for a name to be sent and received on the channel.

Using the \(\pi\)-calculus notation, we may easily express the interaction process, \(P_I = \bar{\alpha}(x).P|\alpha(y).Q\), where \(\{\alpha, \bar{\alpha}\}\) is a pair of dual ports that are considered as the channel between the processes; \(x\) and \(y\) are the name variables that can be instantiated as a specified object for transmission; \(\alpha(x).P\) and \(\bar{\alpha}(y).Q\) are two processes. Note that \(\alpha(x).P|\bar{\alpha}(y).Q\) is not an interaction process since there is no channels for processes to interact. The interaction process is a specified process that the properties of process algebra are retained.

More complicated interaction processes can be constructed via the channels and processes involving multiple participants, for example

\[
Q_I = \bar{\alpha}(x).P|\alpha(y).Q|\alpha(z).S \ldots |\alpha(v).W
\]

means one sends the message to multiple recipients, and

\[
R_I = \alpha(x, y, \ldots, z).P|\bar{\alpha}(a).Q|\bar{\alpha}(p).S \ldots |\bar{\alpha}(q).W
\]

means one recipient and multiple senders.

The interaction processes like

\[
S_I = \bar{\alpha}(x).P|\alpha(y).\bar{\beta}(o).Q|\beta(p).S
\]

can be split into two interaction processes, \(T_I = \bar{\alpha}(x).P|\alpha(y).Q_1\) and \(U_I = \beta(o).Q_2|\beta(p).S\). We will develop a treatment to connect the two interaction processes later.

As an example, simple interaction processes like product quotation and cancellation, can be expressed by the notation of the interaction process. Suppose a customer requests a quotation for a product from a retailer, one would send the request for a list of products and prices. This interaction process, then, is defined as

\[
\text{Request}_{\text{quotation}} = \bar{\alpha}(x).P_1|\alpha(y).Q_1
\]

where \(x\) is the assumed quotation. This interaction process defines process \(\bar{\alpha}(x).P\) which sends quotation \(x\) to process \(\alpha(y).Q\). For the cancellation process, it is defined as

\[
\text{Cancellation}_{\text{order}} = \bar{\alpha}(n).P_2|\alpha(m).Q_2
\]

where \(n\) is a message for order cancellation.

The interaction between two processes is at a high level of abstraction that only indicates what the content of the interaction is, which gives the implementation of the interaction even more flexible to be configured at run time.

Now we consider the transformation of two interaction processes. In process algebra, the reduction rule determines the transformation of two processes. However, we can not deploy the reduction rule for interaction processes due to the logical order of the interaction process because the transformation of interaction processes does not obey the process order. For example, \(\bar{\alpha}(x).P|\alpha(y).Q \rightarrow P|Q\{y/x\}\) that follows the process order, while \(\bar{\alpha}(x).P|\alpha(y).Q\) and \(\bar{\alpha}(x).R|\alpha(y).S\) are two interaction processes that do not obey the process order. However, these interaction processes may have a logical relation i.e. logical order. The logical order determines the transformation of the interaction processes. We use “stepping” to represent the logical order of the transformation of interaction processes.

**Definition:** For \(P_I, Q_I, \ldots \in P_I\), a stepping of interaction processes, denoted as \(\backslash\), transforms interaction process \(P_I\) into \(Q_I\) in the form of \(P_I \backslash Q_I\).
Though the stepping does not directly show the relationship between two interaction processes, the information is actually hidden for the purpose of abstraction. Indeed, with $P_I = \bar{\alpha}(x).P|\alpha(y).Q$ and $Q_I = \alpha(m).R|\bar{\alpha}(n).S$, after a computation step, $\bar{\alpha}(x).P|\alpha(y).Q \rightarrow P|Q\{x/y\}$. Because of the stepping of interaction processes, the next interaction process will be $Q_I$ after $P_I$ implying that $Q\{x/y\} \Rightarrow \bar{\alpha}(n).S$, and $P \Rightarrow \alpha(m).O$, which means that eventually $Q\{x/y\}$ will become $\bar{\alpha}(n).S$ and $P$ will transform into $\alpha(m).O$. A stepping of interaction processes states the logical relationship of transformation. The actual transformation is occurred behind the process interfaces that depends on individual participant’s process implementation.

Finally we introduce an interaction processes series, which will be the abstracted notation for supply chain protocols.

**Definition:** For $P_I, Q_I, R_I \ldots W_I \in \mathcal{P}_I$ and a stepping, an interaction processes series, named IPS, takes the form

$$IPS :: P_I \backslash Q_I \backslash R_I \backslash \ldots$$

Note that an IPS starts with an interaction process originated from a process task, and ends with an interaction process whenever the task is completed.

### 2.3 Example scenarios

Purchasing goods or products via the Internet now becomes common activities in e-commerce. Practically, a current supply chain for e-commerce still follows the traditional model, such that the goods are either stored ready for sale or organised according to customers’ orders. However, the complex customer behaviours require the supply chain to be flexible enough to handle the customers’ needs. Such a flexibility of supply chain is due to the appropriate protocols among participants with respect to supply chain planning, coordination and execution. The following examples, that identify four purchasing scenarios related to book purchasing, show the interactions as protocols. We only discuss the purchasing protocols based on the notation developed earlier, and the protocols of the supply chain will be considered in the succeeding sections.

1. For the simplest purchasing on the Internet, the customer would ask the bookstore for a price quote on the book that the customer wishes to buy. Upon receiving a quote from the bookstore, the customer would accept the bookstore’s offer. The bookstore would then send the book, and the customer would send the payment. This process model is commonly implemented as e-business on the Internet, where the interaction are defined in Figure 1 and interaction processes are evolved accordingly.
3. The customer might delegate the payment via a third party or use a credit card. Then the interaction will involve the bank with different protocols for two types of payment. Here the scenario of payment through a bank without refund processes is considered. Figure 2 shows the interaction structure, and the IPS is given as follows:

\[ IPS : \text{reqQuote} \setminus \text{sendQuote} \setminus \text{sendAccept} \setminus \text{sendGoods} \setminus \text{authPay} \setminus \text{sendMoney} \]

4. The bookstore might have to negotiate with and contract out the actual shipping to a shipper. Here the bookstore interacts with the shipper, and gets the books delivered to the customer. The shipper is then paid by the bookstore, but only after the book has been delivered to the customer. Such an interaction scenario is shown in Figure 3 and the IPS is given by

\[ IPS : \text{reqQuote} \setminus \text{sendQuote} \setminus \text{sendAccept} \setminus \text{reqShipQuote} \setminus \text{sendShipAccept} \setminus \text{sendShipGoods} \setminus \text{sendGoods} \setminus \text{authPay} \setminus \text{sendMoney} \setminus \text{sendShipMoney} \]

**Figure 3. Interaction among Customer, Bookstore, Bank and Shipper**

The above scenarios give four differently evolved interaction process patterns, which indicate that the interaction between individual processes has a logical dependency that depends on the transaction scenarios rather than the participant’s processes.

### 3 Interaction Process Protocolling

To function the distributed processes over the network of a formed supply chain, we require protocols that integrate these distributed processes. We have considered an interaction process as a mediator for enacting two other processes. We also see that a series of interaction processes can be used for modelling business scenarios. However, the algebra based interaction processes can not be directly applied to design protocols without the concerns of a business approach. A protocol is about the automation of machines. However, designing a supply chain protocol involves several much more complex issues not only at the data abstraction but the semantics levels. Traditional protocols in a supply chain are static that the members of the supply chain do not dramatically change over the time and each member must make its commitment to such a set of fixed protocols. In modern e-business, however, the heterogeneity of business processes requires a flexible protocolling scheme that integrates them dynamically. Our focus is to design the protocols that address on the interaction of distributed processes. In this section, we discuss some basic concepts of process commitment, interaction evolving, and protocolling that are related to business process integration for supporting the supply chain formation and management.

#### 3.1 Interaction commitment

Prior to the supply chain formation, we assume that a potential participant, in addition to its own business processes, has the processes interfaces that are ready for interaction, for example, processes for ordering, delivering, payment etc. We may regard the participants of a supply chain as process agents that commit themselves to business processes.

**Definition:** A process commitment is a process relation \( \triangleright \) between process \( P \) and commitment \( \alpha.A \), such that \( P \) is committed to \( \alpha.A \), denoted as \( P \triangleright \alpha.A \), where \( \alpha \) is the action that process \( P \) will perform and \( A \) is the continuation of the process.

Recall that in process interaction, the interaction process is defined as \( \alpha(x).P|\alpha(y).Q \). If \( P_1 \triangleright \alpha(x).P \) and \( P_2 \triangleright \alpha(y).Q \), we have \( P_1|P_2 \triangleright \alpha(x).P|\alpha(y).Q \). We see that the commitment is made on the processes as well as the action by both parties of interaction. It will be a successful interaction if both parties agree on the contents of communication. Otherwise either \( P_1 \) or \( P_2 \) might have additional processes within the continuations of \( P \) or \( Q \) to modify the contents of name \( x \) or \( y \). Moreover, in a supply chain, it is required that the participant commits to a set of commitments \( \mathcal{C} \) for process interactions.

**Definition:** Agent commitment is a relation \( \triangleright \) between the agent’s processes \( P_i \) and a set of commitments \( \{ \alpha_i.A_i \} \in \mathcal{C}, i = 1,2,... \) such that the agent will perform commitments to enable interaction processes.

\[ \{ \alpha_i.A_i \} \in \mathcal{C}, i = 1,2,... \] is said a logical dependent commitment if \( \{ \alpha_i.A_i \}, \{ \alpha_i.A_i \} \in \mathcal{C} \) that can be transformed to another in a logical order for given finite compu-
ation steps, \( \{\alpha_{i1}, A_{i1}\} \rightarrow ... \rightarrow \{\alpha_{i2}, A_{i2}\} \), otherwise they are independent. The dependent commitments are common scenarios in business processes. As an example based on Scenario 1 in Section 2.3, we can see the commitments in the IPS of purchasing below:

\[
P_1 \triangleright \alpha.A_1 \quad reqQuote \quad Q_1 \triangleright \alpha.B_1
\]

\[
P_2 \triangleright \alpha.A_2 \quad sendQuote \quad Q_2 \triangleright \alpha.B_2
\]

\[
P_3 \triangleright \alpha.A_3 \quad sendGoods \quad Q_3 \triangleright \alpha.B_3
\]

\[
P_4 \triangleright \alpha.A_4 \quad sendMoney \quad Q_4 \triangleright \alpha.B_4
\]

Both the customer and bookstore need the commitments for each of interaction processes. The commitment is an important concept for supply chain protocolling, which needs a good engineering treatment. For an established protocol standard, the commitments will give the flexibility in re-engineering softwares for process integration.

3.2 Interaction process evolving

The interaction process discussed earlier only enables the enactment of the individual processes owned by different participants. The transformation from one interaction process to another requires the operation that determines how the interaction processes run logically.

Let us consider scenario 1 in Section 2.3 again as we have the set of interaction processes required to complete the transaction of the purchase process. Assume the interaction process set, \( P_i \in P\), \( i = 1, ... , 4 \), prior to the interaction completion, the interaction process series is given by

\[
IPS : P_{i1} \setminus P_{i2} \setminus P_{i3} \setminus P_{i4}
\]

By interaction process commitments of agents, each interaction process in the series is defined as

\[
P_{i1} = \alpha.A_i | \beta.B_i, i = 1 ... 4.
\]

where \( \alpha \) and \( \beta \) are communication actions including the channel and contents, and \( A_i \) and \( B_i \) are two processes performed by individuals. Then the stepping of the interaction processes gives an interaction process series in the form of

\[
IPS : \alpha.A_1 | \beta.B_1 \setminus \alpha.A_2 | \beta.B_2 \setminus ... \setminus \alpha.A_4 | \beta.B_4
\]

Because \( P_{i1} \in P \) and \( P_{i1} = \alpha.A_i | \beta.B_i \), let \( \alpha.A_i | \beta.B_i \rightarrow A_i | B_i (\alpha/\beta) \), if \( A_i \) implies \( \alpha.A_{i+1} \), denoted as \( A_i \Rightarrow \alpha.A_{i+1} \), and \( B_i (\alpha/\beta) \Rightarrow \beta.B_{i+1} \), then we bring out a new interaction process \( \alpha.A_{i+1} | \beta.B_{i+1} \).

Proposition: For sequential interaction process \( P_T \), there must be two sets of independent processes such that the process involved has the same order as the interaction process. Furthermore the evolving operators define the additional processes between interface processes that transform from one interface process to another.

Proof: Suppose that interaction processes \( P_1, P_2, ... , P_n \) in \( P_T \), ISP : \( P_1 \setminus P_2 \setminus ... \setminus P_n \), by the definition of the interaction process, each interaction process \( P_i = \alpha.A_i | \beta.B_i, i = 1, ... , n \) leads the process stepping \( \alpha.A_1 | \beta.B_1 \setminus \alpha.A_2 | \beta.B_2 \setminus ... \setminus \alpha.A_n | \beta.B_n \), where \( \alpha.A_i \) and \( \beta.B_i \) are communication processes independently performed by different agents. Now \( \alpha.A_i | \beta.B_i \rightarrow A_i | B_i (\alpha/\beta) \), if \( \alpha.A_i | \beta.B_i \rightarrow \alpha.A_{i+1} | \beta.B_{i+1} \), the dependent property of the communication processes gives \( \alpha.A_i \rightarrow \alpha.A_{i+1} \) and \( B_i (\alpha/\beta) \rightarrow \beta.B_{i+1} \). We can simply define \( A_i = \alpha.A_{i+1} \), and refine \( B_i (\alpha/\beta) \) such that \( B_i (\alpha/\beta) \rightarrow ... \rightarrow \beta.B_{i+1} \), an evolving consequence leads to \( \beta.B_{i+1} \). Then \( P_1 \setminus P_{i+1} = \alpha.A_i | \beta.B_i \rightarrow A_i | B_i (\alpha/\beta) \Rightarrow \alpha.A_{i+1} | \beta.B_{i+1} \rightarrow \alpha.A_i | \beta.B_i \).

Based on the analysis above, it is concluded that for a given logically ordered interaction process series among the participants, there are internal processes of each participant such that if the participant commits to the interaction process series then in addition to incorporating processes required, the internal process can be constructed in such a way that its evolution could follow the stepping of interaction processes for communication. It says that as long as the agent commits to the interaction process, the internal process can be constructed or re-constructed to match the interaction process series. Such a series is designed in a logical order based on the business scenarios, which is regarded as a protocol in business communication.

3.3 Protocolling

Given the agent commitments and related processes, in order to bring them together at the process interaction level, we define an protocol schema based on the interaction scenario that describes a logical order of the interaction process as well as agents commitments that enables the interaction processes. The protocol schema defines the interaction of the commitments such that a pair of complementary commitments is matched in the sense of interaction process semantics. For example, assume that \( \alpha.A_i \in C_1 \) and \( \beta_j.B_j \in C_2 \), where \( i, j = 1, 2, ... \in N \), we might match these processes by looking at these actions that are complementary.

Definition: A protocol schema, \( S \), consists of a defined interaction process series, \( IPS \), and interaction commitment \( A_i | B_i, i = 1, 2, ... \) such that the series follows the logical
order of interaction scenarios and the interaction commitments in which enable an interaction process.

In order to establish the interaction among participants’ processes, we consider a procedure for the protocol schema so that processes can handshake with each other:

1. Investigating the interaction scenario among participants within a potential supply chain;
2. Logically collecting the interaction processes required for the scenario, and defining the evolving sequence for the interaction processes;
3. Matching the individual’s processes with the interaction processes based on the individual commitment, and constructing additional processes, e.g. data formatting if required;
4. Semantically formatting the interaction process and the communication process.

### 3.4 A new example scenario

In scenario 4 of Section 2.3 shown in Figure 3, the protocol is given for a model with four participants, namely Customer, Bookstore, Insurer, Bank and Shipper. Here we investigate a scenario of five participants, the additional participant is an insurer that performs the insurance task. It might happen if a customer wants insured shipping, and the bookstore’s existing shipper does not insure goods, then the shipper has to interact with an insurer. That is, the shipping insurance requested by the customer, the shipper carries out the task of interaction with the insurer. The interaction structure among participants may be constructed in Figure 4. The protocol of scenario 4 does not change except the additional interaction processes for a shipping insurance are given below:

\[
\text{IPS} : \begin{align*}
\text{reqQuote} \rightarrow & \text{sendQuote} \\
\text{sendQuote} \rightarrow & \text{sendAccept} \\
\text{reqShipQuote} \rightarrow & \text{sendShipAccept} \\
\text{reqInsure} \rightarrow & \text{sendInsureQuote} \\
\text{sendInsureQuote} \rightarrow & \text{sendCustomerInsureQuote} \\
\text{sendCustomerInsureQuote} \rightarrow & \text{sendInsureAccept} \\
\text{sendInsureAccept} \rightarrow & \text{forwardInsureAccept} \\
\text{sendShipGoods} \rightarrow & \text{sendGoods} \\
\text{sendGoods} \rightarrow & \text{authPay} \\
\text{authPay} \rightarrow & \text{sendMoney} \\
\text{sendMoney} \rightarrow & \text{sendShipAndInsureMoney} \\
\text{sendShipAndInsureMoney} \rightarrow & \text{sendInsureMoney}.
\end{align*}
\]

Each interaction process in the series above is defined as

\[
\begin{align*}
\text{reqQuote} &= C_1 | B_1; \\
\text{sendQuote} &= C_2 | B_2; \\
\text{sendAccept} &= C_3 | B_3; \\
\text{reqShipQuote} &= B_4 | S_1; \\
\text{sendShipAccept} &= B_5 | S_2; \\
\text{reqInsure} &= C_4 | S_3; \\
\text{reqInsureQuote} &= S_4 | I_1; \\
\text{sendInsureQuote} &= S_5 | I_2; \\
\text{sendCustomerInsureQuote} &= C_5 | S_6; \\
\text{sendInsureAccept} &= C_6 | S_7; \\
\text{forwardInsureAccept} &= S_8 | I_3; \\
\text{sendShipGoods} &= B_6 | S_9; \\
\text{sendGoods} &= S_{10} | C_7; \\
\text{authPay} &= C_8 | B_1'; \\
\text{sendMoney} &= B_2' | B_7; \\
\text{sendShipAndInsureMoney} &= B_8 | S_{11}; \\
\text{sendInsureMoney} &= S_{12} | I_4.
\end{align*}
\]

where \(C_i, B_j, I_k, B'_l, \text{and} S_M\) are the process commitments of Customer, Bookstore, Insurer, Bank and Shipper respectively. In this example the additional insurance process scenario is first investigated. The shipper receives the insurance request from the customer, and consults with the insurer and passes the quotation back to the customer for acknowledgment. If the interaction process of the insurance is fine then the insurance payment is made by the shipper to the insurer. The insurance scenario is protocollled and integrated with the existing protocol of example scenario 4. Furthermore, the commitments of the interaction processes among the participants are indicated that the participants’ processes must commit to interaction processes.

We see that at this level of abstraction the communication protocol can be flexible. The protocols can be refined, for example substituting or alternating the protocol, and aggregated by combining sub-protocols for the complicated supply chain network management.
4 Related Work

Related work can be broadly classified under the research areas: e-business models and e-business/supply chain protocols. We briefly describe the relevant literature in these areas to put our work into the context.

Many publications have been made on e-business models, which can not executively listed here due to the space limit. E-business models clearly define value systems that give the value architectures showing who offers what to whom in business approaches [2, 4, 9, 10]. To some extend, protocolling a supply chain is relied on the value architecture - a kind of interaction architecture that provides the basis of possible interaction scenarios - which is certainly beneficial to the protocol design. Several approaches on supply chain protocols have been carried out by some leading researchers. Noticeably, process integration, commitment and Web services based protocols have been investigated. In [1, 3], the protocols are considered as interfaces of business processes, which simplify stake-holders’ process integration and configuration. The use of commitments in modelling agent interaction protocols [7, 8, 11], highlights the benefits of commitment-based approaches to interaction protocol design. The need for process composition and interoperability has led to the development of standards for orchestration and choreography of Web services [6], which has the similar intention as the protocol design.

We have definitely benefited and been inspired from the existing work in our investigation of supply chain protocolling. However, our approach is a scenario based protocolling methodology that the protocol can be altered and refined for the requirements of the supply chain management. More importantly it concerns about how the participants’ existing processes within a supply chain can be integrated with respect to the protocol, which has not been addressed in the previous work.

5 Conclusion and Further Work

We have introduced a technical approach for protocolling that reflects the requirements of the supply chain management. The main contribution lies in the formalisation of protocolling that dynamically generates the communication protocols based on the business scenarios. It proposes an interaction process series that logically sequentialise a collection of abstracted interacting processes, and integrates participants’ existing processes by examining the processes required in each interaction process, which offers a flexible protocolling for the dynamic supply chain. Several examples have been given that demonstrate the protocol designing method for different business scenarios at the high level of abstraction, and can be easily elaborated into a low level of implementation which will be our further work.

The protocolling approach is at the high level of abstraction. The semantics of the abstraction needs to be investigated for the needs purposes of automation. Our current investigation on ontology based integration may help to resolve the stated problem. A protocolling tool that supports the proposed approach would considerably enhance the approach’s capability of our approach. The prototype of such a tool is ongoing. A natural challenge is to develop the interaction semantics and reasoning geared towards protocols.

Acknowledgement

This work is supported in part by Swinburne Vice Chancellor’s Strategic Initiative Fund 2002-2004 on “Internet-based e-business ventures”.

References