Managing Business Relationships in E-Services Using Business Commitments

Haifei Li, Jun-jang Jeng, and Henry Chang

IBM Thomas J. Watson Research Center
1101 Kitchawan Road, Route 134, Yorktown Heights, NY 10598, USA
{haifeili, jjjeng, hychang}@us.ibm.com

Abstract. With the rapid advancement of e-service technology, there is a need to manage business relationships among business entities such as service providers, service consumers, and internal departments. In this paper, we have proposed a novel approach to business relationship management using business commitments and associated business commitment hubs. Business commitments are commitments related to business issues such as service levels in service agreements, and terms and conditions in procurement contracts. The concept of business commitments has captured the essence of business relationships in e-services. Based on case studies, we have envisioned the need of establishing a business commitment hub to centrally manage the external relationships with trading partners and the internal relationships with internal departments. A language called Business Commitment Language (BCL) has been proposed to specify business commitments. These business commitments are used to monitor and control the execution status of e-services. Conceptually, the business commitment hub has two related subsystems: the active subsystem and the dashboard subsystem. The active subsystem responds to business events received from business entities. The dashboard subsystem visually displays the key data and the execution status of business commitments.

1 Introduction

Recently there have been many interests in studying e-services in e-commerce research community. An e-service [3] is usually established between two business entities that are either trading partners or internal departments. A relationship is said to be setup once an e-service has been established. There are many aspects of relationships between business entities, such as legal aspect, social aspect and economic aspect, and each of them requires serious academic research work. One approach to model the economic aspect is through business commitment. In this paper, business commitment is defined as commitment related to business issues. A language called BCL is proposed to specify business commitments from a viewpoint of a single business entity. Since a business entity may utilize multiple e-services to run its business, there is a need to manage them in a uniform way. We call the system managing business commitments "business commitment hub."

During the execution of a typical e-service, there are many data exchanged between a service provider and a service consumer. In order to isolate the business commitment monitoring from the low-level details of e-service implementation, a concept called "KSI" (Key Status Indicator) is proposed. A KSI is an important parameter of the e-service that can manifest its status. An e-service may have multiple KSIs, which are the subject that business analysts want to monitor. The design of BCL follows the ECA (Event Condition Action) paradigm. The ECA is a common design pattern used in active database/system research community. In an ECA system, once an event has occurred, the condition is evaluated. If the condition is evaluated to be true, actions are to be taken. BCL expands the traditional ECA system in several ways. First, the concept of KSIs is introduced. The condition part is a logical expression based on KSIs and commitment variables. Commitment variables are threshold values that can be dynamically set by business analysts. Second, the concept of commitment profiles is introduced. A commitment profile provides values for condition matching variables and commitment variables. A profile provides a way to separate the logic part of a commitment from the data part of the commitment. Third, the action part of ECA is expanded to action set where a collection of related actions can be executed either sequentially or in parallel. In order to support the language, an architecture has been proposed for "business commitment hub." During the build-time, the business commitment hub accepts a BCL document and configures various components. During the run-time, the hub receives events, evaluates conditions, and takes corresponding actions. In the meantime, the KSI and execution status of business commitment are visually displayed in a dashboard.

Most work on SLAs/contracts is solely for external parties (e.g., trading partners). WSLA [1] and tpaML [2] are two specifications related to one-to-one service provider/service consumer relationship. The approach proposed in this paper is applicable to both external and internal parties. Therefore, it is possible to have an integrated view of business relationships to be managed, thus leading to an optimal solution of business relationship management. Traditional contract management or service level management [4] deals with trading partners individually; therefore a global view is missing. The final result is a sub-optimal solution. There is a need to collect the relationships among trading partners and interactions among internal parties, and to manage them globally. In this paper, we introduce a way to build a business commitment hub, which centrally manages business commitments from/to multiple parties. In particular, the concept of business commitments is presented, its corresponding language called BCL (Business Commitment Language) is explained, and an architecture based on BCL is proposed.

2 Business Commitments

According to Merriam-Webster's collegiate dictionary, commitment is "*an agreement or pledge to do something in the future.*" Business commitments are broadly defined as commitments related to business issues. Commitments can be between trading partners (called external commitments), or between internal parties within a business (called internal commitments). The definition of business commitments captures not only the current stable states ("agreement") but also the future actions ("to do something") and constraints (not to do something); therefore it is an appropriate concept to describe certain types of business relationships and interactions that may require both agreements and actions from participating parties. A set of business commitments establishes the agreement of a business commitment hub to its customers (both external and internal) regarding how their artifacts are to be managed. In our opinion, the concept of "business commitments" nicely fits into the business commitment hub that likely manages multiple e-services and multiple parties.

2.1 Business Commitments, Contracts and SLAs

It is critical to discuss the distinctions among Business Commitments, Contracts, and Service Level Agreements (SLAs). Informally, contracts are legal documents specifying the duties and obligations of parties involved in a deal. Contract is a generic term for agreements with a legal flavor. For example, the implication of signing a contract is that "if party A breaks the contract, party B may take party A to a court." An SLA is a contract between a service requestor and a service provider that specifies the minimal acceptable levels for the service. SLA is one type of contracts with a business and quantitative flavor (reflected in the term "level" in SLA). The concepts of business commitments, contracts, SLAs are related, but with different focus. Business commitment is the best concept to fit our needs because of its explicit focus on actions. These actions will be taken to configure, control, and monitor the execution of e-services.

2.2 BPCL Creation

Apparently, most business commitments are derived from contracts and/or SLAs. Some business commitments come from contracts and their relationships that are influenced by the perspectives of the owner of a business commitment hub. It implies that a business commitment hub should monitor and control not only the execution of individual contracts, but also the relationships among these contracts. In BCL, the relationships among contracts are captured as *inter-contract clauses*.

Figure 1 shows procedures for creating a BCL. We assume that Party1 is the owner of a business commitment hub and it negotiates the management agreements with multiple (three in this example) parties. The result of the negotiation between Party1 and Party4 is SLA1 (assuming Party1 and Party4 have negotiated a service agreement). The results of the negotiation between Party1 and Party3, and the negotiation between Party1 and Party2 are Contract2 and Contract3 respectively (assuming these two negotiations are about general business contracts). Party1 may have its internal SLAs/Commitments that describe the obligations of various internal departments. Since SLA1, Contract2 and Contract3 are results of separate negotiations, they are fed into a process called Inter-Contract Analysis to generate possible inter-contract clauses. Inter-contract clauses are combined with internal SLAs/Commitments, SLA1, Contract2 and Contract3 (these are all from Party1's perspective) to form the BCL.

The procedures described in Figure 1 are manual. How to automate these procedures is an important topic but beyond the scope of this paper.

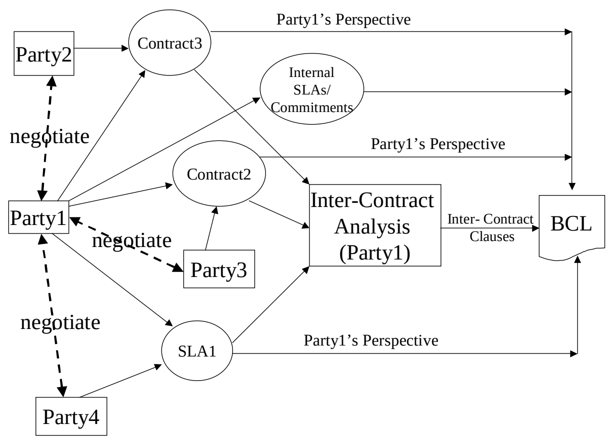

Fig. 1. Steps for Creating a BCL

3 Key Status Indicator and Commitment Profile

One of the major concepts embodied in BCL is *Key Status Indicator* (KSI). A KSI is an important data from an e-service that manifests the execution status of an e-service in the business sense. KSIs are a group of the subjects that a business commitment hub monitors and controls. It is obvious that not all the data within an e-service is a KSI. KSIs should be defined by business analysts who intimately understand the related e-services.

Key Performance Indicator (KPI) is a term used in areas like balanced scorecard [5], business intelligence, and supply chain management to describe the important data/parameters that can be used to measure (i.e., indicate) the performance of an enterprise. One key characteristic of KPI is that an indicator must be measurable. Otherwise, it is very difficult, if not impossible, to manage it. This key characteristic is inherited by the definition of KSI. Our definition of KSI is more specific and is closely tied to e-service.

Once KSIs extracted from e-services have been determined, business analysts can build useful relationships among e-services through KSIs. When an enterprise involves more than two parties and more than one e-services, the monitoring and controlling of the relationship are critical. One hypothetical business commitment could be: "if KSI1 from e-service 1 is greater than KSI2 from e-service 2, notify the business commitment hub."

Commitment profile is introduced to separate the logic part of a commitment from the data part of the commitment. The purpose of separation is to be scalable in composing a large BCL document. The logic part is a logic expression over KSIs and parameters. The data part provides values for these parameters. The logic part should be easy to understand, and the data part is simply a collection of data called commitment profiles. The data part can reside in a database table or other separate XML documents. Therefore, the core part of BCL can be small and easy to understand. Each profile has two components: identifying information and threshold information. Identifying information is formally called ConditionMatchingVariable, and the threshold information is formally called CommitmentVariable. As suggested by the names, the value for ConditionMatchingVariable is used as conditions to retrieve the value for CommitmentVariable. These CommitmentVariables participate in the evaluation of the logic part of the commitment. The evaluation results determine whether actions should be taken.

Let's consider an example from a widely studied domain in e-commerce. In a supply chain, a buyer needs to monitor the customer lead time (CustomerLeadTime) when it purchases a particular product from a particular customer. The buyer usually sets an upper limit for the threshold value of CustomerLeadTime. The symbol $CLT is used to denote that particular upper limit, and the symbol #Supplier and #Product are used to denote the supplier name and the product name respectively. If products are delivered to the buyer, the buyer needs to monitor the CustomerLeadTime. If the CustomerLeadTime exceeds $CLT, the dashboard should be informed and updated, and an email should be sent to the purchasing manager at the buyer side.

The above example illustrates several important ideas. At first, the commitment could be associated with an e-service like productDelivery. Even though the details of the e-service are not provided, it is possible to extract associated data from the e-service through wrappers. CustomerLeadTime is a KSI and can be calculated based on a formula. Second, a commitment is directional. This commitment is from the supplier to the buyer, and the buyer needs to monitor the execution status of the commitment. However, commitments always exist in pairs. Another (although implicit) commitment from the buyer to the supplier is that the buyer needs to pay money within a certain time period. The supplier monitors the status of the payment, and takes certain actions if the commitments are violated. Third, a commitment comes with certain actions. In the above case, there are two associated actions: dashboard notification and email notification. These actions are taken when the commitment is violated, but some actions may be taken regardless of the circumstances, such as a status report of the average CustomerLeadTime at the end of each month. The given actions are relatively simple, but more sophisticated actions are needed to handle a complex situation. Forth, information for commitment profile is easy to identify and understand. In the above example, #Supplier and #Product are condition matching variables and $CLT is commitment variable.

4 Structure of BCL

A BCL document contains a set of inter-related parts.
1. Party: party information. The descriptive information about parties participating in the business commitment hub.

```
    <xsd:complexType name="PartyType">
      <xsd:sequence>
        <xsd:element name="PartyIdentifier"
          type="bcl:PartyIdentifierType"
            maxOccurs="unbounded"/>
      <xsd:element name="Contact"
type="bcl:ContactInformationType"/>
        <xsd:element name="RolePlayer" type="xsd:string"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string"/>
    </xsd:complexType>
```

The party information contains identifier information, contact information, zero to many role players. There is one primary party that owns the business commitment hub. There are one to many parties that are participating in the activities of business commitment hub.
2. KSI: key status indicator. These are important parameters that indicate the execution status of e-service.

```
    <xsd:complexType name="KSIType">
      <xsd:sequence>
        <xsd:element name="KSIName" type="xsd:string"/>
        <xsd:element name="KSIType" type="xsd:string"/>
        <xsd:element name="KSICategory"
          type="bcl:KSICategoryType"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:choice>
          <!--got the value from a business process  -->
          <xsd:element name="ProcessAssociation"
            type="bcl:ProcessAssociationType"/>
          <!-- computing value based on other KSIs -->
          <xsd:element name="Computation"
type="bcl:FunctionType"/>
          <!-- deriving value from a basic KSI -->
          <xsd:element name="ValueDerivation"
          type="bcl:ValueDerivationType"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
```

As illustrated above, there are three different ways to get the value for KSI: directly from a business process, computing the value based on other KSIs, and deriving value from a basic KSI.
3. BusinessEvent. Events provide an entry point for evaluating the logic expressions inside each individual business commitment.

```
    <xsd:complexType name="BEType">
```

```
      <xsd:sequence>
        <xsd:element name="EventName" type="xsd:string"/>
        <xsd:element name="EventType" type="xsd:string"/>
        <xsd:element name="ProcessID" type="xsd:string"/>
        <!-- event source: either Sender (directly come
          from a sender) or Timer (come from a timer) -->
        <xsd:choice>
          <xsd:element name="Sender" type="xsd:string"/>
          <xsd:element name="Timer"
            type="bcl:TimerType"/>
        </xsd:choice>
        <xsd:element name="Receiver" type="xsd:string"
          minOccurs="0"/>
        <xsd:element name="EventAttributes" type =
        "bcl:EventAttributesType" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
```

There are two event sources in our event model: Sender (directly come from a sender) and Timer (come from a timer). Any information specific to an event is stored within *EventAttributes*.

4. BusinessCommitment: The main parts of BusinessCommitment are BCIdentifier, triggering event, commitment level, validity, (logic) expression, initiator, receiver and actions. Actions are a set of action(s) to be taken when the logical expression is evaluated to be true. A commitment is directional, so it is necessary to indicate the initiator and the receiver. There are two possible values for commitment level: individual level (commitment for each transaction instance) and process level (based on the aggregated result over a certain period of time).

```
  <xsd:complexType name="BCType">
      <xsd:sequence>
        <xsd:element name="BCIdentifier"
          type="xsd:string"/>
        <xsd:element name="TriggeringEvent"
          type="xsd:string"/>
        <xsd:element name="CommitmentLevel"
          type="bcl:CommitmentLevelType"/>
        <xsd:element name="Validity"
          type="bcl:PeriodType"/>
        <xsd:element name="Expression"
          type="bcl:LogicExpressionType"/>
        <xsd:element name="Initiator" type="xsd:string"/>
        <xsd:element name="Receiver" type="xsd:string"/>
      <xsd:element name="Action" type="bcl:ActionType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
```

5. *Actions*: Sequential and parallel execution of actions.

```
    <xsd:complexType name="ActionType">
      <xsd:sequence>
      <xsd:element name="ActionCategory"
```

```
        type="bcl:ActionCategoryType"/>
        <xsd:element name="ProcessID" type="xsd:string"/>
        <xsd:element name="ActivityName"
          type="xsd:string"/>
        <xsd:element name="Parameter"
          type="bcl:NameValueType"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="ExecutionMode"
          type = "bcl:ExecutionModeType"/>
      </xsd:sequence>
    </xsd:complexType>
```

ExecutionModeType is defined as an enumeration. Two valid values are "Sequentially" and "InParallel" in our model.

6. Commitment Profile: condition matching variables and commitment variables.

```
    <xsd:complexType name="CommitmentProfileType">
      <xsd:sequence>
      <xsd:element name="ConditionMatchingVariable"
        type="bcl:NameValueType" minOccurs="0"
          maxOccurs="unbounded"/>
      <xsd:element name="CommitmentVariable"
        type="bcl:NameValueType" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
```

5 Architecture of a Business Commitment Hub

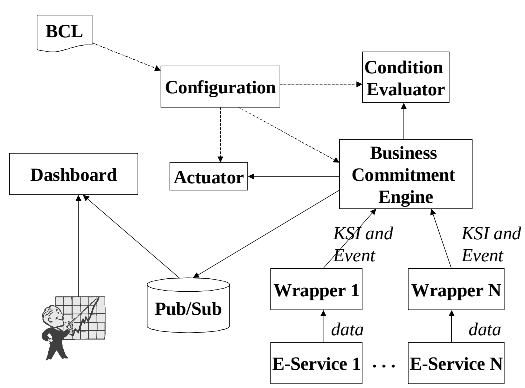

Fig. 2. Architecture of a Business Commitment Hub

Figure 2 shows the architecture of a business commitment hub. During the build-time, the BCL is composed by business analysts, and is passed to the Configuration component. The Configuration component parses the BCL document and configures Actuator, Condition Evaluator, and Business Commitment Engine during the build-time. This part is shown as dotted lines in the figure. During the run-time, e-services send data to wrappers, which convert the data in e-services into KSI data and events. The KSI data and events are inputs to Business Commitment Engine (BCE). The BCE processes received events and calls Condition Evaluator to evaluation condition. Based on the result returned from the Condition Evaluator, the BCE calls Actuator to finish the work. Data generated from Actuator is sent to the Dashboard through pub/sub (publication/subscription) mechanism.

6 Case Studies

6.1 Insurance Hub

In the insurance industry, small businesses may buy insurance policies through independent agents. These independent agents then contact with insurance carriers that actually issue the insurance policies. Since it is time consuming for an independent agent to deal with many insurance carrier that potentially have different policies, it is cost effective if an insurance hub can aggregate the result returned from multiple insurance carriers, and provide a uniform interface to independent agents.

During the business interactions between an insurance hub and insurance carriers, it is possible that the insurance hub may find that the policy rules provided by insurance carriers do not match the real world situation. The insurance hub may send a rule set revision request to insurance carriers. Since policy rule changes require human interventions, insurance carrier may take hours, even days, to process the rule set, and send the results back to the hub. From the viewpoint of insurance hub, it is beneficial if each insurance carrier can return the result with an agreed upon period of time. The value for the time period could be a part of SLA/contract between the insurance hub and carriers.

Two commitments can be derived from the above description. One is from carrier to insurance hub: the carrier must return the result within a specific time period and the insurance hub monitors the result. The other is from the insurance hub to carrier: at the end of each reporting period, the hub reports the average turnaround time to the carrier.

6.2 Supply Chain Management

In a large manufacturing enterprise, the manufacturing facility may be separated from the stocking center for the efficiency reason. The manufacturing facility deals with channels, which are actual customers of the manufacturing enterprise. The stocking center orders raw materials from suppliers. The interactions among these four entities are modeled as e-services. The business relationships among them are modeled as business commitments on top of these e-services. There are potentially many commitments, but three commitments are picked for the illustrative purposes.

The first commitment is called customer serviceability. It is a commitment from the manufacturing facility to channels. Depending on the channel/customer class, the on-time percentage should be 95%. Delivery is said to be on-time if it is finished within the pre-defined delivery time, such as 3 or 4 days. The second commitment is called supplier replenishment, which is from supplier to the stocking center. Depending on the supplier name, part name or part family name, the stocking center may require that cycle time should be less than 2 days, standard deviation should be less than 4 hours, and error tolerance should be less than 2 hours. The third commitment is called forecast accuracy. Depending on part name or part family name, the forecast accuracy should be greater than 80%.

It is easy to identify what KSIs, condition matching variables, and commitment variables are. For example, in the third commitment, ForecastAccuracy is a KSI, PartName or PartFamilyName is a condition matching variable, and 80% is the value for a commitment variable like $FA. During the run-time, the value for $FA can be dynamically modified, thus effectively change the commitment on the fly.

7 Conclusion

In this paper, we have presented an approach to manage business relationships in e-services using business commitments. We argue that business commitment is the appropriate way to managing business relationships. We have proposed a language, and architecture to build a business commitment hub. Two case studies, one from the insurance industry, and the other from the supply chain management, are provided to validate our design.

References

1. Ludwig, H., Keller, A., Dan, A., and King R.: A Service Level Agreement Language for Electronic Services, Proceedings of the 4th International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems (WECWIS 2002), Newport Beach, CA, (2002)
2. Dan, A., Dias, D. M., Kearney, R., Lau, T. C., Nguyen, T. N., Parr, F. N., Sachs, M. W., and Shaikh, H. H.: Business-to-business integration with tpaML and a business-to-business protocol framework, IBM Systems Journal Vol. 40, No. 1, (2001)
3. Kuno, H., “Surveying the E-Services Technical Landscape”, Proceedings of the Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000), Milpitas, CA, (2000)
4. Sturm, R., Morris, W., and Jander, M., “Foundations of Service Level Management,” SAMS publishing, Indianapolis, IN, (2000)
5. Kaplan, R. S., Norton, D. P., “The Balanced Scorecard: Translating Strategy into Action,” Harvard Business School Press, Boston, MA, (1996)