Business integration

Business to business integration with trading partner agreements

How IBM inter-enterprise integration technologies can support tomorrow’s business models

www.ibm.com/software/big/papers
The new economy
As we enter the new millennium, fundamental changes are happening to trade and the way it is organized. There is a growing shift towards an electronically connected world where ideas, information and services are replacing the traditional reliance on physical goods production as the primary generator of wealth and employment. The rules that have defined and governed business since the industrial revolution are changing. Traditional industries are facing non-traditional competition. The vertically integrated corporation, where a single organization spans and controls an entire value chain, is becoming the exception rather than the norm.

In the new economy, market dynamics will dictate a business model that provides for the integration of different partners in a value chain. Using a variety of IT technologies, this model can enable highly co-ordinated trading communities, each with the ability to operate like a ‘virtual enterprise’.

The companion paper ‘business survival in the next millennium’ explores the pressures for change motivating new information technology to help businesses adapt.

Technologies to support the new economy
The emerging web economy requires an agile enterprise, one that can work more directly with suppliers and customers and respond more rapidly and intelligently to change. Technologies such as the Internet are beginning to transform traditional business models. Business pressures – margin erosion, channel proliferation, rising customer expectations, time-based competition and faster product commoditization – are placing increased emphasis on how organizations operate and inter-operate with other enterprises.

These dramatic changes in the business environment demand dynamic and flexible integration between partners in the value chain. While new technologies will be needed to enable such integration, these will need to be able to work seamlessly with existing inter-enterprise business processes (e.g. EDI) and leverage investments in existing Enterprise Application Integration (EAI).

IBM is currently developing new technologies designed precisely to support this type of business-to-business integration. This paper describes these technologies, which are a key component of IBM’s Application Framework for e-business¹, and how they can be used with IBM’s MQSeries and WebSphere middleware to build robust and flexible integration solutions. Using this approach, both traditional enterprises and Internet focused ‘dot.coms’ can fully exploit the exciting business opportunities of the next millennium.

---

Figure 1. Application framework for e-business
**What are the requirements?**

Facilities such as Electronic Data Interchange (EDI) have successfully provided electronic document interchange between companies and their suppliers for a number of years. However, EDI’s high cost and inflexible structure has always proved a barrier to adoption by all but the largest enterprises. While EDI will continue to evolve, utilizing pervasive networks such as the Internet to reduce costs, complementary technologies are emerging, able to provide some of the key capabilities necessary to enable dynamic business process integration. The basis of these technologies is the formulation of:

- A ‘common language’ that can be employed by existing or potential trading partners to specify how they will interact
- An ‘electronic contract’ that employs this common language in order to define and enforce the interaction protocols with which they will do business.

While XML-based business to business protocols such as OBI and RosettaNet are beginning to set a standard for business interactions (albeit currently fragmented), IBM has been expanding on this base with the development of flexible electronic documents termed Trading Partner Agreements (TPAs). These TPAs operate within a Business-to-Business Protocol Framework (BPF) that provides a comprehensive tool set for the specification, configuration, customization and execution of electronic contracts between business partners.

Technologies such as XML and TPAs, coupled with advances in middleware and workflow software, provide the key building blocks needed to underpin an electronic business-to-business integration infrastructure. Supporting the extensible and easy to use TPA format with the BPF framework and middleware from IBM’s MQSeries and Websphere families, enables dynamic business process integration by providing:

- Integration of internal processes, using modifiable ‘business rules’ to route information between the various internal business information systems
- Secure, reliable and auditable electronic document interchange between organizations
- Externalization of appropriate business functions and processes to suppliers, customers and partners.

In addition, this infrastructure can provide support for:

- The use of a broad range of standard message formats, transport and business protocols and network connections with the capability to dynamically connect new trading partners
- Easy to use, business oriented ‘single point of control’ for interactions across an extended or virtual enterprise
- Extensible open interfaces with flexible connectors to link to existing applications.

The BPF framework meets these requirements.

---

1 Business to business integration is a key component of the application integration layer of IBM’s Application Framework for e-business. Additional information on the Application Framework for e-business can be found at [http://www.ibm.com/framework](http://www.ibm.com/framework).
Integration of internal processes
Successful integration within the enterprise is the first step toward business to business integration. Proven technologies for Enterprise Application Integration include connectors for coupling to databases and ERP systems, message handling including flexible message transformation and routing, and workflow management for coordination and control of business processes. TPA documents and supporting BPF infrastructure coupled with the services in IBM’s MQSeries middleware and WebSphere application servers, enable these integration styles to be extended to operate between businesses and across firewalls.

Secure, reliable and auditable document exchange
Security is a critical concern for interactions between business partners, particularly across publicly accessible networks. Security features such as reliable sender authentication, audit trails and secure logging and checking for permissable messages are crucial to trusted business trading. Historically, capabilities such as these have been enabled through Value Added Networks (VANs) often with EDI interactions. To fully exploit pervasive networks such as the Internet, it is important that these types of security features be maintained.
Externalization of business functions
When a business externalizes automated functions to make them available to business partners several new issues must be addressed which were less important or had simple solutions within a single enterprise. If the function becomes accessible via a public network, how will it be reached, who will be allowed to use it, and how will these users identify themselves? What do both parties in the resulting long-running conversation commit to in terms of behavior, responsiveness and recovery from error situations? How can rules of interaction be specified even when the partners have no shared middleware? The TPA is structured to gather the information needed to address these issues and hence provides the basis for middleware which constructs wrappers allowing business functions to be externalized safely.

Broad connectivity
Communications support that enables interaction between potential trading partners—no matter what transport protocol and what network they use to connect—is a key requirement for participation in dynamic business value chains. In practice, this means that a business-to-business infrastructure must support:

- A wide set of common transport protocols including HTTPS (for secure Internet interactions), SMTP (for mail interactions) and reliable messaging such as that provided by IBM's MQSeries
- A wide range of message formats including those based on industry standards for exchanging XML documents
- Existing and emerging business-to-business protocols including EDI, OBI and RosettaNet
- Transcoding from one message format to another.

It must also be possible to merge additional libraries into the middleware base, to support any specialized message formats and protocols that may be needed for specific interactions.

As more dynamic business chains emerge, it will become increasingly important to be able to implement new electronic trading arrangements easily and quickly. This is enabled by providing dynamic connectivity and by leveraging trading partner information in publicly accessible registries and LDAP directories to help identify and locate new potential business partners.

Single point of control and extensibility
Having a consolidated, business oriented 'single point of control' for an extended or virtual enterprise will become a necessity. This facility can help in gathering together templates for e-business interactions, assisting in accelerating the connection to, and start of trading interactions with, new partners.

Open and extensible interfaces
As dynamic business value chains evolve, the protocol interaction styles in general use will be extended and specialized to encourage increasingly sophisticated interactions between partner businesses, each with specialized capabilities, but able to act together as a virtual enterprise to meet the needs of specific industries and business contexts. By providing open interfaces, extensibility points and services to enable the use of XML technology in messages and interfaces, IBM's BPF provides support for these types of extension.
Trading partner agreements
TPAs are XML based documents used to specify the business interaction between trading partners. A TPA defines how trading partners will interact at the transport, document exchange and business protocol layers. Declarative documents, TPAs specify these roles and responsibilities as a set of attribute value pairs. The document is completed by providing specific attribute information for a particular pair of interacting trading partners.

How are TPAs used?
A business uses a TPA document to define the agreed model of interaction with a specific trading partner. The TPA represents a single long-running conversation, consisting of a set of related interaction steps, distributed over time but comprising a single Unit of Business (UOB). Take the example of the conversation between a traveller and travel agent to arrange an itinerary. A TPA is used to define the allowable interactions i.e. making reservations, modifying or cancelling them, issuing tickets and confirmations, making payments. In handling this UOB the travel agent process will start its own conversations with partner hotel and airline businesses. Each of these subordinate business-to-business conversations is governed by its own TPA.

One TPA can be used for many independent UOBs between the same trading partners – either serially or concurrently. A single TPA can include definitions for multiple interaction sequences and multiple message formats, any of which can occur in a UOB instantiated from it. Effectively, a TPA acts as the 'control centre' for all system-mediated interactions with an external entity.

Over time, the accumulation of TPAs can become an effective repository of enterprise inter-business process descriptions, providing a major tool for enabling processes enhancements.

How are TPA specified interactions organized?
The TPA simplifies the specification of a business-to-business interaction by organizing the information into separate functional layers. Data flow through the runtime layers is governed by specifications in the TPA. This layering provides appropriate abstractions for business data flow and minimizes the need for specialized coding. The functional layers specified in a TPA and supported in underlying BPF runtime processing are:

- Transport – which handles the selected transport protocol, network connection and basic security
- Document exchange – which provides document abstraction, including message data mapping, non-repudiation, time-stamping, logging and audit
- Business protocol rules and interface – which provides message sequence and responsiveness checks, document type and trading partner specific data handling, together with interface logic to connect to specific local business applications.

---

3 IBM intends to propose this format to XML ORG for consideration as an open standard.
4 A Unit of Business is a set of actions grouped by the user as associated with achieving a specific business result such as a sale and fulfillment.
How are TPA’s structured?
A TPA document contains the following information:

- **Overall properties** – includes TPA name, starting and possibly ending dates for validity of the agreement and other global parameters

- **Role and identification** – identifies the parties to the TPA as logical roles such as ‘buyer’, ‘seller’, ‘airline’, ‘hotel’ etc. Specific organization names and contact information such as e-mail and postal service addresses are then provided for each role. The allowable actions under a TPA are organized by role making it easy to modify an existing TPA to specify identical interaction rules but with a different partner. An optional role is that of external arbiter for use in resolving disputes

- **Communication and security** – these attributes specify the communication and interaction protocols to be used. The specification is layered into transport, message handling and higher level conversational sections. Protocol choices and parameters for addressing security including authentication, certificate handling and non repudiation are included

- **Action and sequencing** – for each identified role, this is a menu of the actions that can be performed on request from the partner. A signature is specified for each action defining the parameters and their data types. Sequencing rules specify constraints on the order in which actions can be requested. Example actions in the case of the travel agent process interacting with a hotel would be requests to make a room reservation and subsequently to modify it

- **Cancellation and error handling** – cancellation rules indicate whether the result of a completed action can be cancelled, and if so, the constraints under which such a cancellation is permissible. Error handling rules manage error conditions (e.g. be the maximum waiting time for the response to a request). Commentary text can added to the TPA to cover other negotiated – but not processed – issues.

---

**Figure 3. TPA and BPF structure**
TPA preparation and setup with BPF

BPF includes a suite of tools to support the TPA lifecycle including preparation, setup and generation of new TPA documents and the generation of interaction processing from them.

- Authoring tools assist the creation of new TPAs. In many cases, it may be convenient to construct a new TPA by combining elements from those previously deployed, tailoring the result for the new target interaction. The authoring tools for TPA include modelling and template functions to simplify TPA assembly from pre-existing parts

- Registration tools are used to populate a registration database with interface information identifying the user application logic to be bound into the business to business interactions, and user provided helper functions such as specialized parsers, security handlers and encoders

- Code generation tools use the TPA and information from the registration database to generate program objects (executable files) which together with the runtime BPF support implement the required business to business interaction processing.

Hence starting from a TPA authored with the tools described above, and using a combination of generated and runtime logic, BPF provides a complete implementation of all message and interaction logic needed for a particular trading partner interaction. As TPAs are XML based, the strong syntactic checking available within advanced XML editors can be used in these tools to diagnose and warn of inconsistencies.

Runtime management

At runtime, the BPF Manager uses the data in the registration database and the generated program objects to manage information flow through the system and control execution of the business rules in the TPA. In addition, functions such as event handling, security services and logging are handled by BPF’s Runtime Services. The BPF generated code enables inter-enterprise integration, alleviating the need to develop custom programs to manage networking, business event handling, message processing and sequencing.

Figure 4. BPF setup and runtime operation

(a) Authoring tool
(b) Registration tools
(c) Generation tool
Generating interaction code from TPAs allows appropriate security validation and message logging steps to be added systematically. Hence the level of security is adjusted to the communication context of each business partner connection, providing appropriate protection for secure interactions across a publicly accessible network. BPF generated checks for responsiveness of the partner and valid message sequences improve reliability further.

As noted earlier, to provide an effective single point of consolidation for interactions, a business-to-business infrastructure needs to provide a broad range of connectivity options. In addition, it is important to provide services for defining, parsing and manipulating XML messages together with facilities for importing other message libraries and services. This can simplify the mapping of other existing application and message formats both into, and out of, XML messages.

**BPF runtime services**

When interacting with business partners across open networks such as the Internet, there are likely to be considerable variations in request/response times. In these cases, a single business flow requiring responsive interaction with multiple partners, such as in the travel example above, will need to employ parallel concurrent conversations. During these types of conversations, partners may detect unexpected conditions requiring specialized action by the interaction or business logic software. The BPF provides services that map interaction events from message processing into business events which can be meaningfully handled in the business applications. These services enable the coupling of application processing to appropriate interaction processing, with the ability to control how interaction events are filtered and combined.

Other BPF services provide support for:

- **Synchronous call requests** — to allow business applications to make simple synchronous calls requesting an interaction. The framework transparently maps these into asynchronous messaging interactions, preserving the synchronous appearance to the application.

- **Interaction event definitions** — a service to allow the definition of interaction events and event combining rules which result in applications being ‘called back’ only when a specified set of interaction events or time-outs has occurred.

- **‘Call back’ rules** — to indicate which application methods should be invoked when a particular business event occurs.

- **Subordinate conversations** — to provide logic that specifies that a subordinate conversation with a new partner is to be started and considered part of the one currently in progress.

- **Cancellation and Compensation** — a cancellation/compensation framework facilitating cleanup when a business interaction has to be cancelled.

- **Each of these services contributes toward providing a simplified and generic view of external events to internal business applications. Information about the conversation state is available to internal applications to assist in event management. The combination of user provided interface and rules code with the BPF services mediates between the entirely generated interaction handling support derived from a TPA, and business logic written as a standard application or business workflow.**
Using the BPF

To build successful business-to-business IT solutions implies a need to understand in detail how to interact and exchange messages with applications and business processes owned by external trading partners. However, the variety of protocols needed for this messaging is innately complex and related to technology rather than the structure of a particular business process. Defining interaction styles using TPAs, together with BPF runtime services and support from appropriate middleware, provides a rapid, easy to use and flexible approach to the development and deployment of business-to-business solutions.

Utilizing the BPF can improve and accelerate development of new inter-enterprise e-business solutions in a number of ways, including:

- Interaction definition simplification – structuring and simplifying the definition of interactions with trading partners, specifying what each party will expect of the other.
- Choice of connections – handling all popular modes of communication to enable interaction and data interchange with any potential trading partner.
- Automated code generation – automating the generation of connection and interaction logic enabling rapid commencement of trading interactions with new partners.
- Code reuse – allowing generic logic within existing business processes and applications to be reused for various partners using different types of interconnection.
- Evolutionary approach – enabling smooth transitioning to broader e-business capabilities, building on any existing EDI or message based integration capabilities.

The value of TPAs

XML based TPA documents provide value in that they capture the essential information which trading partners must agree on in order for their applications and business processes to communicate. In integrating the business process of multiple partners, there is value in distinguishing ‘interaction’ information – on which there must be agreement – from more private design and implementation decisions – which can be made autonomously and independently by each participating partner.

TPAs and their associated authoring and registration tools can help speed the process of defining how new trading partner interactions should take place. Since the TPA is a formal XML document, XML tools can provide a valuable level of consistency checking to detect obvious mismatches immediately. In many cases, the definition of the interaction style can be further simplified by starting from a sample TPA template as a model, or basing the TPA on one of the emerging standards for e-business such as Open Buying on the Internet (OBI) or RosettaNet.

Support for all widely used communication modes makes it possible to reach agreement on interaction rules – and therefore a TPA – with almost any trading partner. This ability to interact with any potential partner provides the opportunity to attract and select e-business partners from the broadest possible marketplace.
The value of the BPF

Using the BPF tools, a TPA document can be rapidly created to define the interaction between an enterprise’s business systems and those of a trading partner. Subsequent automatic code generation from this TPA – combined with runtime services – provides a complete implementation of the required interaction and communication logic. Therefore, within a relatively short space of time, an enterprise can be ready to connect with a particular partner.

When partners have been identified and agreement reached to conduct e-business, (assuming an agreed interaction mode is also ready) conversations can be quickly and easily initiated and business trading begun. This ability to start conducting e-business with new trading partners immediately, without waiting for programming help, provides the ability to attract and select e-business partners in real time. To be able to move quickly and leverage full advantage from these new trading interactions, the programs implementing internal business processes need to drive generic ‘reusable’ business-to-business interactions.

Choosing new trading partners and starting interactions dynamically also depends on being able to support a range of interaction choices. Since all the necessary protocol checking and messaging logic needed for any one TPA is automatically generated, internal applications issuing higher level interaction requests will be able to reflect just the business process steps. This separation of partner interaction from business logic flow makes it easier to dynamically add or change partners and helps simplify the logic within business applications. This makes it easier to evolve and extend the function of automated business processes.

BPF helps separate network interaction processing cleanly from business processes. This is more important in business-to-business integration than within an enterprise because of the greater variety of interaction technologies involved.

In summary, BPF provides

• tools for authoring or assembling electronic TPA documents and generating interaction code from them.

• the runtime system to support the automatically generated interaction code.

• operational tools for managing and monitoring e-business operations with trading partners.

• tools, APIs and services for interfacing existing applications and business processes to the generated logic for interacting with trading partners.