BUSINESS-TO-BUSINESS (B2B) INTEGRATION TECHNOLOGY

Businesses worldwide started exchanging electronic business messages with each other around 1970. This coincides with the establishment of wide-area computer networks. Businesses realized the potential immediately to send electronic messages instead of paper letters in order to conduct business electronically. Computer networks provided a significant increase of transmission speed, less failures due to message losses, and direct processing of messages upon receipt without manual transcript from paper to computer terminals or vice versa. Overall, business interactions became a lot more reliable and efficient.

The direct processing capability of electronic messages enabled the seamless integration of the messaging environment and the back-end application system processing. The electronic integration of businesses was achieved this way and the technological basis was called business-to-business (B2B) integration technology. However, B2B integration technology was not affordable to every business independent of its size due to its high deployment and maintenance cost. Therefore, only big businesses could afford electronic B2B integration. Smaller businesses continued to rely on paper letters or fax transmissions.

Today it is commonly accepted to exchange electronic messages between businesses. Current trends are to make B2B technology more accessible to every business independent of its size through ubiquitous Web technology. This new technology is called Web services and all software vendors readily provide solutions (Alonso, Casati, Kuno, & Machiraju, 2004; Bussler, 2003).

BACKGROUND

Dedicated networks called value added networks (VANs) started providing the transport of electronic messages between businesses around 1970. Since their service is very reliable they still are extremely popular today and widely used. VANs do not support direct communication between businesses. Instead, businesses have to upload and download messages asynchronously from VANs. VANs provide a persistent mailbox system where every partner has a dedicated mailbox storing the electronic messages addressed to it. Based on their behavior VANs are an asynchronous network for exchange of messages. In addition (“value added”), they provide services like storage, backup, and billing.

Through the emergence of VANs, business-to-business (B2B) integration was born (Bussler, 2003). VANs made a big difference in competitiveness since businesses could rely on the extreme high speed of electronic data transmission compared to paper-based communication through postal services. Different industries embarked on B2B integration through, at that time, a new form of communication.

It became immediately apparent that it is not advantageous at all for businesses to define their own message formats and send them over VANs. This would require a business to deal with all the different message formats defined by all its business partners if each one would provide its own message format. A significantly better approach was to standardize the message formats across businesses so that a business could use the same message formats across all its business partners from the same industry. In specific industries, message definition standards have been developed for over 30 years now. An example for the supply-chain industry is EDI (ASC, 2004), for the banking industry is SWIFT (2004), and for the insurance industry is ACORD (2004). These standards are called B2B standards and enable the fast integration of a business with others in the same industry. A business can comply with the standard and by that is ensured that the message exchange with the other partners is interoperable. The standardized message formats ensure interoperability.

While the message formats have been standardized, the technical software implementations for B2B communication have not. Every VAN is providing its own communication software for uploading and downloading the messages from the mailboxes. In order to allow messages to be automatically processed by back-end application systems, businesses had to integrate their VANs’ communication software into their information system environment. This back-end application system connectivity is necessary for retrieving and storing the data from the back-end application systems that are communicated with through the communication software. That meant that each business had to do custom implementation in order to make the B2B communication with its partners work.
INTEGRATION TECHNOLOGY

In the late 1980s and early 1990s, the first B2B integration products appeared on the software market. These products were off-the-shelf software that did not require any custom coding by the businesses anymore in order to either connect to the VAN’s software or the back-end application system software. Instead, it could automatically link the VANs’ communication software and the back-end application systems (like, for example, enterprise resource planning, ERP, systems) within the businesses. Establishing B2B integration became more and more like a turn-key software solution instead of a custom coding task.

This automatic connectivity to back-end application systems is achieved through specialized software adapters that adapt the B2B integration product’s interface to the particular interface of the back-end application system (Apte, 2002). An adapter enables the B2B integration technology to insert as well as to extract data from the back-end application system. The benefit for the businesses was that they did not have to perform any custom coding anymore but could use prepackaged B2B integration software instead. That allowed buying the integration functionality without going through internal software development processes. A standard called J2EE Connector Architecture (JCA, 2004) allows software vendors to build standardized adapters. This contributes to the turn-key nature of integration products.

The main problem of B2B integration, besides integrating back-end application systems and the communication software to connect to VANs or the Internet, is data definition mismatches. An example of a data definition mismatch is that an address can be defined as a single string or as a structured record, where each address element like city or zip code is stored in a separate field. Mediation (or, synonymously termed, transformation) overcomes the data definition mismatch problem that exists between B2B standards and the data models of back-end application systems. The same piece of data can be represented very differently in a B2B standard and in a back-end application system. When data is extracted from the back-end application system in its format, it has to be reformatted to the message format of the B2B standard. This reformattting is called mediation (Bussler, 2003; Omelayenko & Fensel, 2001; Rahm & Bernstein, 2001). Not only the structure of the data might have to be changed (e.g., from a one-string representation into a structured record), but also the content representation (e.g., “Ireland” in one representation has to be replaced by “IRL” in another one). The change has to happen in such a way that the meaning of the data (data semantics) is not changed at all, i.e., the data semantics have to be absolutely preserved.

Not only data structures and vocabularies can mismatch, but also the message exchange sequences. For example, while a back-end application system might send one message and expects a return message back, a B2B interaction might have more messages, including acknowledgement messages, in order to establish an exactly once transmission. These differences in message exchanges are overcome by process mediation (Fensel & Bussler, 2002). Process mediation ensures that all communicating parties receive the messages in the sequence they require.

When businesses send and receive messages they have to make sure that they know their communication partners. Otherwise they might accept messages that are not sent by a contractually bound partner. If this is the case then messages might be processed that should not. On the other hand, messages must only be sent to partners, not to any other organization. In order to define partners as well as their specific properties, standards like collaboration protocol profile/ agreement (CPP/A) are established (ebXML, 2004).

Adaptation, data and process mediation, as well as partner management are the most essential concepts that integration technology has to implement.

CURRENT DEVELOPMENTS

The availability of software products for B2B integration lowered their price significantly, mainly because custom coding was not necessary any more. This enabled smaller business to participate in B2B integration and soon more businesses made use of the B2B integration technology then ever before.

In the mid-1990s XML (XML, 2004) emerged as viable technology and was picked up by software developers around the world. At the same time, the first B2B integration products based on XML as the message syntax were formed and appeared on the market. The promise was that the new technology was going to make the B2B integration task a lot easier and a lot cheaper. However, this was not the case as the same integration problems like adaptation, mediation, or partner management had to be solved, just on a different technological basis. Mediation, adapters, connectivity to network software, etc. all require a solution in the context of XML-based integration technology, too.

The Internet was available as a communication platform and many B2B integration products started using the Internet instead of VANs. That proved difficult since the open and unreliable Internet required the development of secure and reliable communication protocols. This is in contrast to VANs that are proprietary networks and their access is restricted to the VANs’ customers. In addition,
Related Content

Transformation-Based Database Engineering
www.igi-global.com/chapter/transformation-based-database-engineering/11228?camid=4v1a

Remarks on a Fuzzy Approach to Flexible Database Querying, Its Extension and Relation to Data Mining and Summarization
www.igi-global.com/chapter/remarks-fuzzy-approach-flexible-database/52299?camid=4v1a

A Framework to Build Fuzzy Object-Oriented Capabilities Over an Existing Database System
Fernando Berzal, Nicolás Marin and Olga Pons (2005). Advances in Fuzzy Object-Oriented Databases: Modeling and Applications (pp. 177-205).
www.igi-global.com/chapter/framework-build-fuzzy-object-oriented/4811?camid=4v1a

An Attempt to Establish a Correspondence between Development Methods and Problem Domains
Oscar Dieste, Marcela Genero, Natalia Juristo and Ana M. Moreno (2004). Advanced Topics in Database Research, Volume 3 (pp. 166-187).
www.igi-global.com/chapter/attempt-establish-correspondence-between-development/4359?camid=4v1a