USING ebXML FOR SUPPLY CHAIN TRACEABILITY
Pitfalls, solutions and experiences

A. Bechini, M.G.C.A. Cimino, A. Tomasi

Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni. University of Pisa, Via Diotisalvi 2, 56122 Pisa (Italy). Fax: +39 050 2217600; tel: +39 050 2217599; e-mail: {a.bechini, m.cimino, a.tomasi}@iet.unipi.it

Abstract: In this paper, we analyze the main issues of traceability along the supply chain: architectural solutions, business process interaction, lot identification, traceability information management, and communication standards. In particular, the employment in this setting of two different standards for inter-enterprise business collaboration (as ebXML and Web Services) is discussed. Moreover, different standards for lot identification and data capture, as EPC for RFID and EAN/UCC for bar code are taken into account, uncovering their potential contributions in reducing the cost of procedures for tracking goods. The Cerere project experience is finally reported: it is shown that the architecture of a Web Information System framework (developed to assess food supply requirements) witnesses the actual possibility to support process integration and semantic interoperability via XML-based technologies.

Key words: Traceability, Information Systems, Web Services, ebXML.

1. INTRODUCTION

Traceability may be defined in general as the "ability to trace and follow any product through all stages of production, processing and distribution"[1]. Traceability systems are constituted by three basic elements[2]: i) univocal identification of units/batches (denoted as *lots* in the following) of every product components, ii) information collection about time and location for every lot transfer/transformation, and iii) a method to relate this kind of data. In practice, traceability systems are record-keeping procedures that store

information about and show the path of a particular product/component from the supplier(s) into the business, throughout all the intermediate steps (which process and combine components into new products) and all through the supply chain to the end consumers.

Therefore, both products and processes form key components in a traceability system, which is in charge of storing information about them. Information representation and data flows may be dissimilar in distinct actual traceability systems, depending on the particular functional specifications coming from legal requirements, certification needs for the target system, incidents prevention policies or practices for quality improvement. The availability of existing information systems along the supply chain stages have to be taken into account, both in the definition of the data model and in the implementation of the message interactions, as well as in the choice of the communication structure and protocols.

The traceability information flow can follow the product flow along the supply chain, embedded in it, or can be passed through data repositories of any kind, external to the supply chain. The traceability information flow can be directly linked to the product flow by leveraging a proper lot identification mechanism. The more straightforward form of identification is based on the assignment of a numeric or alphanumeric string to every lot. Such a string actually gives no information about the lot, but it is intended to provide a univocal key to retrieve traceability data stored elsewhere. To guarantee the uniqueness of this string, standard systems have been introduced, as will be discussed in sect. 3.

Once the lot identification has been performed in some way, the retrieval of information crucial to traceability asks for established standards for the elicitation, representation, and storage of the required data. Considering that a large number of companies along a target supply chain might hold their own information systems, possibly containing data relevant to traceability as well, the crucial role played by communication and information exchange facilities becomes particularly evident: in this setting, the communication infrastructure is thus required to overcome the heterogeneity of the involved information systems. Technologies for XML-based business interchanges[2,4] can serve as the framework for defining new process cooperation paradigms. Among such paradigms, ebXML[5,6,7] enables enterprises to conduct business over the Internet in more straightforward and efficient ways. The ebXML modular suite of specifications provides organizations with a common, automated method of exchanging business messages, conducting trading relationships, communicating data using common terms, and defining and registering business processes — such as ordering, shipping, and invoicing[8]. The ebXML Message Service (ebMS) defines the message enveloping and header document schema used to transfer ebXML messages over a

communication protocol such as HTTP or SMTP, and the behavior of software sending and receiving ebXML messages. The use of ebXML-related solutions can be evaluated by comparison with other technological approaches, i.e. based on the Web Services stack, facing with integration and interoperability requirements. A discussion on these topics is presented in sect. 3 and sect. 4.

Article 3 of the EU General Food Regulation (178/2002), which is in appliance since 1 January 2005, forces traceability concerns to the food supply[1]. A generic supply chain technical disciplinary have to accurately set the rules for each certified food supply chain, with regard to material flow, production process management and execution, document flow and business process collaboration through partners agreement, and responsibility assumptions. Traceability requirements are stated as well as quality and safety goals.

Nevertheless, a preliminary analysis of existing traceability systems reveals some open problems[9,10]. For instance, typically only a few stages of a supply chain are supported by a traceability information system, and in many cases no kind of information system is used at all. Whenever present, often the information system is a legacy one, hampering both integration and interoperability. Answering to many of the above-mentioned issues, the main goal of the project named *Cerere* at the University of Pisa is the development of a robust and efficient traceability framework, implementing a Web Information System applied to the food supply. The overall architecture of the underlying data repositories is discussed in the next section, and the Cerere Traceability WIS is briefly depicted in sect. 5.

2. TRACEABILITY INFORMATION MANAGEMENT ON THE SUPPLY CHAIN

An abstraction of traceability information systems can envision a massive, centralized database capturing in a single location all the information about each lot at each stage of the supply chain. The logical view of a lot contains attributes for each feature of every product and its components, as well as details of the processing phases. Any actual traceability system adopting a centralized solution can be regarded as an implementation of the so-called *push* model[11]. According to this paradigm, as soon as each actor in the supply chain collects traceability data, it pushes these data into the centralized traceability database; each single data recording that is relevant to traceability is completely transferred from the actor to the central database. However, the implementation of a centralized traceability database is neither realistic nor efficient in most actual settings.

In fact, we should keep logically separated the traceability information (related to product transformations and flows on the chain) and other kinds of product characteristics (either constant or variable over time): while the former can be managed through a centralized system, the latter are suitable to be stored just once at the source site. Moreover, the involved data repositories can be proficiently associated with different stages or with external data trustees, thus obtaining a physically distributed architecture encompassing different nodes, each of them possibly addressing local specific implementation problems.

Indeed, food supply chain is characterized by a number of peculiarities that can be summarized as follows[12]:

i) Heterogeneous Structure and Naming of Data. For several years, important agricultural communities have wrestled with the task of identifying the relevant information to be captured and stored in an agricultural database for a given product, and developing a standard naming convention for each data element in that database. Producers have failed in building consensus for any single standard for any single commodity, and there is no reason to believe that consensus will ever be reached.

ii) Confidentiality and Control of Data. Food chain participants, at all segments of production, are often highly protective of their own data, thus they would not agree on sharing their company's data. The industry is concerned that a centralized database would create issues of data confidentiality and trade disruption. Ownership, movement, and location data, might be used for purpose other than the goal of traceability. Further, there are potential data integrity issues.

Given the obvious benefits from value traceability for increasing corporate profitability, the implementation of a logically centralized database should consider possible alternatives. The architectural solution that is achieving widespread consensus accounts for the distribution of traceability information among different robust databases along the supply chain, and allow for a connectivity backbone between such databases. Actually, the system is not required to operate with constant connectivity. Data may be held locally either within the management system of each actor of the supply chain or associated with the lot itself. When connectivity is achieved at a key point of the chain, the cache of information on the lot is updated. Thus, different actors can use different structure and naming of data and agree on a common vocabulary only when interaction is required. Further, each actor is responsible for confidentiality of its data and will provide the other actors with only the information concerning the traceability. Typically, the distributed architecture uses intermediate data trustee. A data trustee is a private, third party intermediary between the responsibility actors each to other and with other entities: companies, government, individuals, or

associated consumers. Each actor transfers its location and ownership data to a data trustee. The data trustee acts like an escrow agent, holding the actor's data until a legitimate product health investigation need would be established.
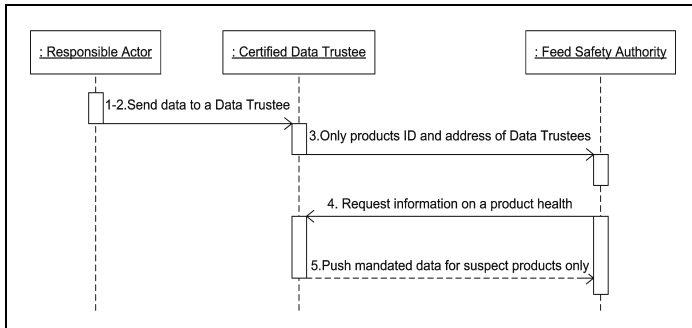
Figure 1. "Pull" model: in this paradigm, the actor is expected to send all the required data towards a certified data trustee.

Figure 1 introduces the "pull" model by means of an UML sequence diagram, where the data flow is handled in a five-step process. First, the actor inputs data into its private software system normally used to manage the operation. Second, this software system is linked to a data trustee chosen by the actor. Obviously the two initial phases can be accomplished manually if there is not an information system. Third, mandated data (no commercial or production data) are pushed to the data trustee, who exposes only the product's identification number to external databases. No other ownership, location or movement information is sent at this point. Fourth, authorized users or government officials request information on the food, as in the case of detecting a consumer's health incident. Fifth, the data trustee publishes to the requester the mandated data for suspect food only.

We anticipate that there would be multiple data trustees, and these trustees would be certified and audited by the government or the government's appointed agency, such as a trade association of a certain class of food product. Actors are free to choose which data trustees they wish to collect and store their data. Larger actors might even choose to apply for certification as a data trustee similar to the self-insurance approach many large corporations use for risk management.

In the "pull" architecture, there can be still a single, central database. However, this database would store only the identifiers of the product and of the data trustees where the location, ownership, and movement of the product were stored. Information on a product might be stored at more than one data trustee. The identifier can allow to know where the data on each

product were stored, thus safeguarding business information. Of course, the system could be checked to determine that each data trustee is performing its obligations.

This "pull" architecture has proved to be effective in other fields such as the global credit card organization. Credit card transactions can occur within a matter of seconds even though the technology must seamlessly link a large number of separate databases. This is also the architecture used in the Brazilian national animal identification program, which covers a national herd roughly twice the size of that in the United States[13].

A "pull" database strategy can also equally meet or exceed the trace-back standard and address the privacy and data mining concerns raised above. Some producers and processors may still opt to "push" their data in a global public database, e.g. for enhancing the value of their products by information about the source origin, or about particular quality features, or about the identification with a valuable brand. However, the use of a data trustee will provide an alternative, helping actors protect the confidentiality of their data, protect the integrity of their existing trading relationships, and increase data integrity within the system.

3. BUSINESS PROCESS INTEGRATION INFRASTRUCTURE

Retailers are not going to be willing to have interconnections with a huge number of disparate traceability systems. A fast food outlet, for example, would not like to use a separate system for their meat, their baked goods, their dairy products, their lettuce, their tomatoes, their catsup, and so forth. They would prefer to access a single system able to provide all the necessary information.

The best solution is to build independent, private data sharing networks that are very loosely interconnected. A private data sharing network[12] begins with one sponsoring company at any segment of production, and proceeds linking to individual supplier and customer companies, inter-connecting with each to expand the initial network.

Typically, private data sharing networks focus on a certain class of food product. In an ideal future, there would be many such private data sharing networks for each commodity. Once built, each independent, private data sharing network could operate autonomously and also be loosely linked with other such independent networks via technology that makes the system appear a single to a downstream customer without exposing the data from one independent system to another.

To raise the optimal way for designing a given independent private data-sharing network, first of all one must avoid point-to-point connections. A strong temptation is to begin from connecting one application to another using standard application program interfaces (API's), XML data structures and SOAP (Simple Object Access Protocol). However, this road leads to significant problems. When there are only two applications to be connected, there is only one interface that needs to be built, but as number of participants increases to n, all the n would need to write software interfaces with each other so that leading to a total of n x n interfaces. The problem gets even worse if at each segment of production there is more than one pre-existing application program that will need to be interconnected (e.g., procurement system and separate manufacturing system). The bottom line is that the point-to-point method of connection just is not sustainable.

This instability becomes even more apparent when one of the participating applications changes. A single change in the application requires that all point-to-point interfaces from that application be changed. The cost of maintaining such a system is staggering.

A much better approach is to connect each connection of third-party applications in a private data-sharing network to an Enterprise Service Bus (ESB), an integration middleware, standards-based, service-oriented backbone capable of connecting hundreds of application endpoints. ESBs combine messaging, Web services, XML, data transformation and management to reliably connect and coordinate application interaction. In our scenario, ESB would translate data from one third-party system (say a producer system) to an internal "data bus" format, and then retranslate this information to another third-party system, such as the first stage processor's system. The "data bus" approach requires developing only an interface between each third-party application and the "data bus", thus reducing the complexity of the integration. Further, the infrastructure is more resilient to application and business changes.

Theoretically, each company publishing its data in a common language (e.g., XML) using the same data tag names would accomplish the task of building a private data sharing network. However, such an approach would still result in some serious problems. The first problem is that each participant in the chain has to agree on the naming convention for each data element in the XML structure. And when the supply chain crosses national boundaries, the introduction of a different language adds even more complexity. These observations suggest that the ESB has to take naming translation into charge, mapping one data element from one application system to another. Thus, each application program can use its own terminology for each data element. Figure 2 shows the generic functional components of an Enterprise Service Bus using a UML notation. Here, a

Responsible Actor, a Feed Safety Authority, or a Data Trustee interacts each other with own private interface (the lollypop symbols in the Figure) through the ESB. The Messaging Oriented Middleware (MOM) facilities provide a software infrastructure to support an asynchronous interchange of information. The ESB also offers integration with the broad spectrum of components likely to be encountered through the use of various 'standard' binding approaches such as Web Services, J2EE Connector Architecture, JMS, COM, and other common mechanisms. This integration is dealt with by the ESB in a standard, service-oriented way, independent of the particular binding technologies. The ESB also offers a level of transformation capabilities and XML services to address the problem of differing data format requirements in the heterogeneous components, and intelligent routing facilities to govern the flow between components.
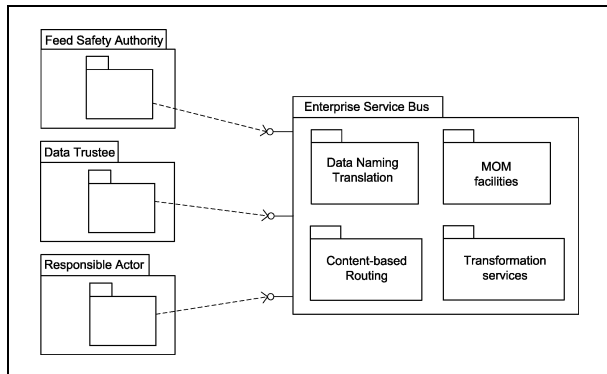

Figure 2. Generic functional components of an Enterprise Service Bus for traceability purposes.

To make traceability effective often means managing a lot of information on each product reference, each production batch, each stock movement, each shipment, etc. We underline that traceability is a tool intended for follow the path of a specified physical unit of goods. Thus labeling and automatic identification are often a bottleneck within a company, as well as between partners in supply chains. The strong separation between product information (e.g. related to quality features) and traceability information (identifiers and links between them) brings along many advantages. Actually it enables a better management of a traceability system as a more separated subsystem with respect to the individual Information Systems of each responsible actor.

Linking together the two information flows implies the use of a unique identifier for the lot. The simplest form of identification consists in a numeric or alphanumeric string. The string gives no information about the

lot, but provides a univocal key to retrieve traceability data stored elsewhere. To guarantee the uniqueness of this string, standard systems have been introduced. The most promising is certainly the EAN/UCC system[14,15]. By administering the assignment of company prefixes and coordinating the accompanying standards, EAN/UCC maintains the most robust lot identification system in the world. As regards traceability, the numbering structures of immediate interest are the Global Trade Item Number, which identifies uniquely each commercial unit, the Serial Shipping Container Code which identifies uniquely a logistic unit (dispatch unit), the Global Location Number which identifies any legal, functional or physical location within a business or organizational entity[16]. More complex form of identification can however be realized, by introducing descriptions of the key features of the item[17]. Another emerging numbering standard is the Electronic Product Code (EPC), under development by the UCC too[18]. It is a scheme for universally identifying physical objects via RFID tags and other means. The EPC Identifier is a meta-coding scheme designed to support the needs of various industries by accommodating both existing coding schemes where possible and defining new schemes where necessary.

The most used technologies for automatic identification are certainly the one-dimensional and two-dimensional (or matrix) barcodes, and the radio frequency identification (RFID). Unlike the one-dimensional barcodes, which have a very limited storing capacity, the matrix barcode can encode more than a kilobyte of data. The achievement of this performance, however, requires a very high print quality and consequently expensive printers. RFID identifiers (tags) consist of a chip, which can be attached onto or implanted into any surface of an item[19,20,21]. Apposite devices can read data from and possibly write data into the tags. RFID tags can be passive, i.e., are powered by a remote energy source, or active, i.e., contain their own energy source. Active tags have greater storage capacity and greater read range than passive tags[22]. On the other hand, they are physically larger and more expensive than passive tags. Further, their life depends on the duration of the energy source (approximately 6-10 years). As regards food traceability, RFID technology appears to be very promising, because it allows to store incremental information during the processing steps, but it is currently unsatisfactory, especially for cost and read range[23,24,25]. Unlike barcodes technology, for instance, RFID allows acquiring information from several (up to 1000) tags concurrently.

4. BUSINESS PROCESS INTEROPERABILILITY

An important standard for business-to-business communication has been used for almost a quarter of a century, Electronic Data Interchange (EDI). EDI has been applied as a fast and reliable means of achieving electronic, computer-to-computer exchange of information between trading partners[26]. Though EDI techniques work well, they are difficult and expensive to implement. Therefore, use of these techniques has been normally limited to large enterprises possessing mature information technology capabilities. The proliferation of XML-based business interchanges[2,4] has served as the catalyst for defining a new global paradigm that ensured all business activities, regardless of size, could engage in electronic business activities. Such paradigm, denoted electronic business eXtensible Markup Language (ebXML) is an international initiative established by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT)[8] and the Organization for the Advancement of Structured Information Standards (OASIS)[27].

Technically speaking, two enterprises willing to do business with each other need to agree on ways how to invoke business services or business processes on a business partner's system and how to exchange data. Obviously, both business partners need to have a common understanding of protocols and formats, as well as message content.

Of course, inter-enterprise business collaboration must take potential failure into account. For example, company A sends a message to company B, which is received. Then, however, due to a communication problem, company B cannot send a response back to company A, which leaves company A in the dark about whether the message got through or not. If company A sends the message again, it may receive two shipments. Potential failure conditions are numerous. Now, the question is if and to what extent Web services and ebXML meet the requirements that result from business collaboration characteristics and business service types. Roughly speaking, Web services are request/response with no business process semantics. This is in keeping with the three specifications that enjoy broad acceptance in the vendor community: SOAP, WSDL, and UDDI. Figure 3 shows the stack of technologies. In addition, there is BPEL, a language specification introduced in May 2003. It aims to describe how to coordinate Web Services within a process flow.

In inter-enterprise scenarios, a requester would invoke a remote service. Technically speaking, this is no issue and has already been proven to work in many cases. However, the question is how to cope with potential failure conditions: there is no way to discover a Web service implementation unavailability or modification before service invocation. Also, WSDL and

SOAP provide for technical interoperability but not for semantic interoperability[28]. Web services technology is about loose coupling in the technical sense. However, it is tight coupling in a conceptual sense. The use of a Web service does not require an agreement between requester and provider.

Behavior	BPEL	
Service	WSDL	**ebXML**
Message	SOAP	
Type	XML Schema	
Data	XML	

Figure 3. Web Services and ebXML standards.

ebXML represents a set of modular business collaboration-oriented specifications. Business collaboration requires a solid and consistent conceptual foundation, encompassing the concept of inter-enterprise business collaboration based on mutually accepted trading partner agreements, as well as the concept of a technical infrastructure which enables businesses to find each other and provides for the reliable and secure exchange of business messages between collaborating business partners. Of course, business partners can also be represented by functional units within the enterprise. Like Web services technology, ebXML provides technical interoperability through a vendor-neutral protocol. ebXML uses Collaboration Protocol Agreements (CPAs) to declare bindings to business collaboration specifications. ebXML requires collaborating partners to mutually agree upon the formats and semantics of business documents, which are XML-encoded. However, it is not an actual constraint to only allow XML-encoded messages within ebXML, which would even leave room for transmitting EDI messages. In an inter-enterprise business collaboration scenario, both business partners would use the ebXML Message Service (ebMS) to securely and reliably transport business documents. The ebMS is defined as a set of layered extensions to the base Simple Object Access Protocol (SOAP) and SOAP Messages with Attachments (SOAPAttach) specifications[29], defined by the W3C organization[30].

However, the ebMS just represents the message envelope. It requires an additional content standard to define the semantics of a business document (which represents the content of the envelope, i.e. the message payload). As there are several horizontal and vertical content standards in existence, a

novel initiative, called Universal Business Language, it is in achieving a universal XML business language over ebXML. ebXML-based business collaboration assures a reliable and recoverable message exchange as well. Business level failures are completely taken into account with the Business Process Specification Schema (BPSS). For example, if a party fails to respond within a pre-defined time period, then the BPSS reverts to the previously known secure state. The message-exchange agreement between two business partners is described by means of a Collaboration-Protocol Agreement (CPA). However, if one business partner afterwards changes the interface of a business service identified in the CPA, it renders the CPA invalid and requires a new CPA to be built. However, it doesn't affect the technical message exchange. Hence, the sender can be sure that the message gets delivered and the recipient has to deal with a potential problem. ebXML has its major strengths when it comes to inter-enterprise business process integration. However, ebXML is also suitable for intra-enterprise business process integration in that functional units (e.g. divisions) are treated as separate mini-enterprises. In B2B scenarios, the specific strengths of ebXML and Web services can be combined in that ebXML is used for managing enterprise-spanning business transaction services in the context of collaborative business, while Web services find their place in intra-enterprise integration of back-end systems.

In this perspective, the Cerere project has designed an interoperability architecture for traceability systems based on the ebXML technology as the reference specification in defining and exchanging business documents. Figure 4 shows the Cerere architecture, which is general enough to be applied to systems in different contexts; in particular, it has been used in a case study involving the food chain. It is worth pinpointing that the main component of the message switching system is the Message Service Handler (MSH): it takes care to validate Cerere documents and to send and receive them over the Internet using HTTP as its transport protocol. Furthermore, it provides error-handling facilities for a number of situations that may arise in real life.

Finally, a final consumer is able to access traceability information through a Web Interface connected to the database by means of a Web Services based infrastructure. Whenever Process Collaboration is the main goal, MSH System provides the proper interaction protocol among peers, as a document-centric approach is followed[28]. On the other hand, the access to Business Information Services can be proficiently achieved via Web Services, able to provide an efficient and lightweight RPC-based interaction. Registration and authentication services are provided for different classes of users, and are implemented within a Web Application module. The support

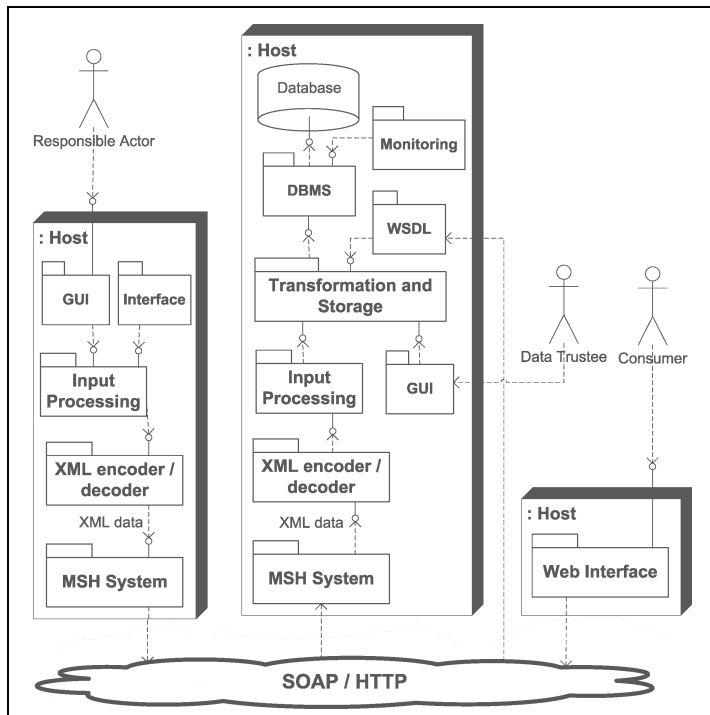for publication is implemented by a Content Management System (namely OpenCms v. 5.0.1).

Figure 4. In the Cerere architecture shown above, the communication infrastructure leverages a Message Service Handler for exchanging and validating documents among the involved nodes.

The overall architecture has been designed to provide a clear, well-structured organization for the modules involved in a traceability system, addressing the typical issues for this broad category of information systems. Anyway, we must also underline that each specific domain covered by actual traceability systems presents particular challenges in obtaining the required performance level. Although performance is not the main focus in the presented architectural design, it must be always taken into account: from this standpoint, we can easily note that the messaging modules represent the main bottleneck. Thus, any attempt to improve performance (and subsequently scalability) must solely concentrate on the design, implementation, and configuration of the messaging subsystem. Although at first sight the treatment of XML documents might seem a computing-intensive activity, today the employment of highly optimized libraries for

this kind of tasks can easily overcome any performance problem from the XML-processing modules.

5. CONCLUSION

The food production environment is largely composed of a wide set of small and medium enterprises that play different roles in the supply chain, with different levels in technological competence, economic resources, and human skills. In order to achieve data interoperability along the supply chain for traceability purposes, a common, widely accepted set of specifications for collaboration is required. In this context, XML and SOAP can be surely regarded as emerging enabling technologies. Anyway, the plain help from XML and SOAP is not sufficient to address all the semantic aspects of each document exchange (which process generates the data exchange, what is the meaning of each data item, etc): document communication and sharing among business partners should be unambiguously modeled.

A recently proposed standard to provide semantics, elements, and properties necessary to define business collaborations is ebXML: thus, its employment in the context of traceability should be taken into account. The goal of the ebXML Specification Schema is to provide the bridge between e-business process modeling and specification of e-business software components. Business process models describe interoperable business processes that allow business partners to collaborate.

The Cerere project has developed a Traceability Web Information System for the food chain, and the ebXML technology has been successfully adopted to support collaboration among all the involved actors at different stages of the supply chain. The dictionary of XML business components and the set of document types built upon them are the first step of the Cerere data model definition[2]. The supply chain processes are defined by means of CPA documents and messages are delivered to a data trustee repository using ebXML mechanisms. The front-end interface of the Traceability Information System is implemented by a Web Application, and it has been tested on a specific food supply of typical Tuscany products.

6. ACKNOWLEDGMENTS

7. REFERENCES

1. Regulation (EC) n. 178/2002, European Parliament and Council of 28 Jan 2002, ch. I,V.
2. A. Bechini, M.G.C.A. Cimino, B. Lazzerini, F. Marcelloni, A. Tomasi, "A general framework for food traceability", IEEE SAINT 2005 Workshops, Proceedings pagg. 366-369, January-February 2005, Trento, Italy.
3. C. Binstock, D. Peterson, M. Smith, M. Wooding, C. Dix, C. Galtenberg, "The XML Schema Complete Reference", Addison Wesley Professional, 2003.
4. eXtensible Markup Language (XML), XMLhttp://www.w3.org/XML/.
5. ebXML official website – http://www.ebxml.org.
6. B. Gibbs, S. Damodaran, "ebXML Concepts and Application", John Wiley & Sons, 2002.
7. A. Kotok, D.R.R. Webber, "ebXML: The New Global Standard for Doing Business on the Internet", New Riders Publishing, 2001.
8. UN/CEFACT – United Nations Centre for Trade Facilitation and Electronic Business, http://www.unece.org/ cefact/.
9. M. de Castro Neto, M.B. Lima Rodrigues, P. Aguiar Pinto, I. Berger, "Traceability on the web – a prototype for the Portuguese beef sector", EFITA 2003 Conference, 5-9 July 2003, Debrecen, Hungary, pp. 607-611.
10. C.A. van Dorp, "Tracking and Tracing Business Cases: Incidents, Accidents and Opportunities", EFITA 03 Conference, 5-9 July 2003, Debrecen, Hungary, pp. 601-606.
11. W.R. Pape, B. Jorgenson, R.D. Boyle, J. Pauwels, "Let's get animal traceback right the first time", Food Traceability Report, Feb 2004, pagg.14-15.
12. W.R. Pape, B. Jorgenson, R.D. Boyle, J. Pauwels, "Selecting the most appropriate database architecture", Food Traceability Report, Feb 2003, pagg.21-23.
13. M. de Mello, P. de Azevedo, "Effects of Traceability on the Brazilian Beef Agribusiness System", Virtual Library Osvaldo "Bebe" Silva, Quilmes, Argentina 2000.
14. EAN/UCC official website – http://www.ean-ucc.org/.
15. EAN International, "EAN•UCC Traceability Implementation", february 2003.
16. EAN Belgium-Luxembourg, "EAN/UCC standards for shared supply chain traceability", http://www.eanbelgilux.be/ PAGE/brochure_tracability.pdf.
17. The Ministry of Agriculture, Forestry and Fisheries of Japan, "Guidelines for Introduction of Food Traceability Systems", March 2003, http://www.maff.go.jp/trace/ guide_en.pdf.
18. EPCglobal official website – http://www.epcglobalinc. org/.
19. K. Finkenzeller, "RFID Handbook", Carl Hanser Verlag Munich, 1998.
20. R. M. Hornby, "RFID solutions for the express parcel and airline baggage industry", Proc. IEE Colloq. RFID Technology, London, U.K. Oct 1999
21. U. Karthaus and M. Fischer, "Fully Integrated Passive UHF RFID Transponder IC With 16.7-µW Minimum RF Input Power", IEEE Journal of solid state Circuits, Vol. 38, No 10, 2003, pp. 1602-1608.
22. A.L. Annala, I. Oy, U. Friedrich, "PALOMAR Passive, Long range Multiple Access high frequency RFID system", IST-1999-10339.
23. Matrics Inc., "Matrics UHF RFID Tag Family", 2004, pp. 5.
24. Philips Semiconductors, "I CODE1 label ICs Protocol Air Interface", May 2002, pp. 42.
25. Philips Semiconductors, "UCODE HSL", Oct. 2003, pp. 9.
26. UN/EDIFACT - United Nations Directories for Electronic Data Interchange for Administration, Commerce, Transport, http://www.unece.org/trade/untdid/welcome. htm.
27. OASIS - Organization for the Advancement of Structured Information Standards, http://www.oasis-open.org.
28. Dieter E. Jenz, "ebXML and Web Services - Friends or Foes?", 27.06.2002, http://www.mywebservices.org/index. php/article/articleview/451/1/1/.
29. SOAP, http://www.w3.org/TR/soap/.
30. W3C – World Wide Web Consortium, http://www.w3. org/.