Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers in Industry 55 (2004) 181–196
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

An XML implementation process model for enterprise applications

Yu-Hui Tao[a,*], Tzung-Pei Hong[b], Sheng-I Sun[c]

[a]Department of Information Management, National Pingtung University of Science and Technology, Pingtung County, 1, Hseuh-Fu Rd., Nei-Pu Hsiang, Pingtung County, 91201, Taiwan, ROC
[b]Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung City, Taiwan, ROC
[c]InfoChamp System Corporation, Kaohsiung City, Taiwan, ROC

Received 20 May 2003; accepted 20 February 2004

Available online 19 June 2004

Abstract

This study proposes an implementation process model for integrating eXtensible Markup Language (XML) into enterprise applications, which also meets the inter-organization data exchange standard of RosettaNet. This model is motivated by the Manufacturing Execution System (MES) group of Taiwan's Mechanical Industry Research Laboratories (MIRL) in meeting their needs on enterprise applications integration for producing the data exchange specifications. There are seven stages in this process model, but only the analysis-and-design stage is specific to XML characteristics for defining Data Type Definition (DTD) to be used with XML in the inter-organizational data exchange. This paper illustrates the core sub-process of defining DTD and producing XML documents. We start with a MIRL example regarding information exchange of a work order between the MIRL-MES and an ERP system connected by IBM MQSeries. Then the RosettaNet components are used to demonstrate how additional elements can be incorporated in the analysis-and-design stage by comparisons.

Keywords: Data exchange; Data Type Definition; Enterprise applications; eXtensible Markup Language; RosettaNet

1. Introduction

As the application of supply chain management (SCM) grows its popularity in business, the concept of adopting an inter-organizational data exchange standard becomes a common recognition among the top management. However, data exchange is an old issue that was often ignored or underestimated in internal enterprise applications. It may be even more difficult to design and implement an internal data exchange than an external one for SCM, since, not only the external requirements, but also the internal requirements from different business applications need to be met.

The research and development group of the Manufacturing Execution System (MES) at the Mechanical Industry Research Laboratories (MIRL) in Taiwan has already successfully implemented its MES software in 18 Taiwan's semi-conductor manufacturers

* Corresponding author. Present address: Department of Information Management, National University of Kaohsiung, Kaohsiung City, Taiwan, ROC. Tel.: +886 7 5919191; fax: +886 7 5919374.
E-mail addresses: ytao@nuk.edu.tw (Y.-H. Tao), tphong@nuk.edu.tw (T.-P. Hong), shengi.sun@msa.hinet.net (S.-I. Sun).

0166-3615/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2004.02.001

over the last few years. Recently, existing clients are demanding an eXtensible Markup Language (XML) [1] data exchange mechanism to work with partners' RosettaNet standard. Therefore, the MES group hopes to define an XML implementation process to meet the internal enterprise application integration (EAI) needs of MIRL as well as the external RosettaNet standard that has been adopted by many of its existing clients in their SCM applications [2].

MIRL-MES's scenario can be seen in many organizations that evolve into a transition point between the data processing era and the information system era, which spends considerable expenditure on integration and information sharing within the organization [3]. Moreover, modern information systems have been evolved into what Ward and Peppard [4] called strategic information systems in their three-era model, which reflects the fast-changing nature of business driven and external focus. Consequently, new information technology like XML plays an important role in current information systems integration within an organization and inter-organizations for competitive advantage.

Accordingly, MIRL-MES's need presents a global business issue to be addressed. A solution to this issue can be described as follows: MIRL is currently providing MES, enterprise resources planning (ERP), and product data management (PDM) to Taiwan's high-tech corporations. While, MES is primarily used in manufacturing and PDM in design and production, ERP generally covers all functional areas in sales, distribution, production planning, and control, procurement, material management, accounting, financial, cost control, human resource, and so on. One scenario is that the ERP requests a production update from the MES, and another scenario is that based on the build-to-order (BTO) customer order of its mass customization process, the ERP selects a product structure or configuration from the PDM, which is to replace or enhance the traditionally known bill of material (BOM). Currently, integrated middleware, which claims to support heterogeneous enterprise applications and databases, may be used to exchange data across a range of enterprise applications. However, the middleware is constrained by the applicable scope in nature and required to quickly respond to any specification changes of the supported applications. If MES, ERP, and PDM, were XML-based systems with a unified DTD complying to the RosettaNet standard, then they can be a more coherent and efficient set of MIRL enterprise applications without subject to the capability of the third-party middleware.

This research proposes an implementation process model for integrating XML into enterprise systems, which is illustrated by core MIRLs-MES. The organization of this paper consists of the literature in Section 2, the process model in Section 3, examples of the XML design in Section 4, followed by the conclusions in Section 5.

2. Literature review

To further refine the research goal, some related works about XML in manufacturing applications, MES related research, information technology deployment, RosettaNet, systems integration, and collaboration are reviewed in Sections 2.1 to 2.5, respectively.

2.1. XML in manufacturing applications

XML 1.0 was first published by World Wide Web Consortium (W3C), in 1998. It is a subset of Standard Generalized Markup Language (SGML), for describing information, so computers can easily understand its meanings [5]. Roy and Ramanujan [6] indicated the distinction among SGML, HTML, and XML, as follows: HTML is suitable for quick publishing of simple web pages; SGML is appropriate for applications of complex structure, but is not widely accepted in Internet; and XML is good for documents that are structured and for long-term usage and, thus, very applicable for internet applications.

In addition to its own standard, there are other XML related standards, such as Document Type Definition (DTD) for describing the schema of a structured document, eXtensible Style Language (XSL) for displaying and translating the document style, and eXtensible Link Language (XLL) for defining the linkages of documents. The above standards further XML development to its maturity for practical use [7]. Generally speaking, XML has the following nine characteristics: (1) it can directly be applied on internet; (2) it can support widely used software applications; (3) it is compatible with SGML; (4) it is easy to develop related software for processing XML documents; (5) the options for its functionalities should be minimized as much as possible; (6) its documents should be easy-to-read and clearly understood; (7) its design should be concise and careful; (8) its production should be quick; and (9) its syntax should not be unclear [8].

The applications of XML in manufacturing are increasingly popular. Huang et al. [9] listed the advantages of XML/electronic data interchange (EDI) for building business-to-business (B2B) information exchange, including readability, popularity, flexibility, heterogeneity, rich format, and low cost. Shiau et al. [10] used Common Object Request Broker Architecture (CORBA) communication framework and distributed multi-agents for extended distributed collaborative design in concurrent engineering tasks, and XML for setting Manufacturability Markup Language (MML) in data exchange for extended product information or production equipment. Glushko et al. [11] proposed using XML and Web agent for replacing CORBA and traditional EDI for achieving inter-business data exchange and electronic commerce. Ziao et al. [12] used XML as the data exchange standard for IC SCM and referred RosettaNet as the process standard between businesses.

In XML related standard making, Lu [7] pointed out that XML's standard can be classified into three kinds: (1) basic standard, such as XSL, Xlink, Pointer and DTD; (2) vertical XML business standard in referring to XML specifications for a certain domain or business; and (3) horizontal XML business framework in referring to common data elements, DTDs, schema, and frameworks for XML specifications. The most popular progress in the vertical XML standard is RosettaNet for semiconductor industry, while the counterpart in the horizontal business framework includes BizTalk and ebXML.

To summarize the above findings, XML has become the mainstream in data exchange. Many organizations involved in setting XML related standards, and academic references also showed that XML is the de facto standard in data exchange. Currently, XML defines its traditional format based on DTD, but W3C has listed XML Schema as a better alternative. XML Schema is in general a better tool than DTD because it supports additional features that include supporting standard and user-defined data types, namespace, inheritance, and using XML syntax [13]. However, DTD may not be easily replaced by XML Schema due to the following three reasons: all SGML tools and many XML tools can process DTDs, a large number of document types are already defined using, DTDs, and there are widespread expertise and many years of practical application in DTD [14]. Because, RosettaNet has been adopting DTD as its structure definition style, this research follows the usage of DTD but suggests observing the future development in XML Schema.

2.2. MES related research

Parker [15] mentioned that MES utilizes an integrated relational database for tracking variables, such as personnel, materials and production, and outputs detailed instructions to work floors. Stephen [16] also indicated MES must take manufacturing information, including BOM transformation, production routes, and item numbers from engineering systems. Also, in the management cycle of Plan-Do-Check-Action, the information delivery must be transparent in order for ERP or SCM to work in production. This can be achieved by making production information systems and work flows (Do) under careful production control (Check, Action) for advanced planning (Plan) [17], where information gap between information systems can be bridged by MES.

Manufacturing Enterprise Solutions Association (MESA) defined MES, as an information deliver that promotes the optimized production activities from work orders to final products. When a situation occurs, MES quickly responses to it by reducing non-value-added production activities based on real-time information available in order to maintain production efficiency [18]. MIRL has devoted to research in manufacturing operational control systems over the years. MIRL-MES integrated information from six functional modules, including Basis Module, Customer Order Management, Material Management System, Production Scheduling System, Work-in-Process (WIP) Tracking, Equipment Management System, and Statistical Process Control [19]. Compared to the MESA International functionalities, the MIRL-MES system has an extra module for management, whose major role is to perform order information exchange with the ERP system, while MESA has additional modules in performance analysis, human resources management, and document control [20].

2.3. Related research in implementing information technology

Many methodologies have been proposed in the literature for implementing information systems (IS) in a business. From a strategic viewpoint of Ward and Peppard [4], information technology (IT), and IS have a supply–demand relationship. In other words, IT is an enabler to IS and, thus, an IS implementation should encompass special IT treatments, especially when the IT is new. It is, therefore, necessary to establish an XML-based implementation approach by taking into consideration critical success factors or implementation methodologies from existing ISs. We review the literature on specialized and generalized information application systems below.

2.3.1. Specialized information application systems

Three information system applications are of our particular interest—ERP, concurrent engineering, and manufacturing technology. A typical ERP implementation approach can be seen with SAPs accelerated SAP [21], which has five phases including project preparation, business blueprint realization, final preparation, and ‘‘go live & support’’.

Two major concurrent engineering information applications are, engineering data management and PDM. The implementation approach proposed by Chen and Tsao [22] consists of nine steps: business analysis, process analysis, and modeling, process reengineering, requirement analysis, and system planning, system design, system modeling, software component selection, system implementation, system integration and testing, and improvement.

Chen and Small [23] proposed an integrated planning model for advanced manufacturing technology (AMT). AMT has a wide application scope, such as Computer Aided Design/Computer Aided Manufacturing, Computer Aided Process Planning, Flexible Manufacturing System, and Final Assembly Schedule. AMT implementation activities can be organized into eight units, which are strategic planning for AMT adoption, product and process matching, AMT monitor, management and control, inter-department relationship, external environment relationship, AMT, and measuring system performance evaluation.

2.3.2. Generalized information application systems

In the process model for information technology by Copper and Zmud [24], implementation of IT is defined as an organization focusing on expanding proper IT to employee users. Zmud and Apple [25] proposed a process model, which corrects IT implementation through merging some behavior models. In other words, Zmud and Apple viewed IT implementation from the technology translation perspective, which takes every opportunity seeking IT applications in order to globally root IT in an organization. Launi [26] proposed an off-the-self software structure with eight steps that include defining and evaluating needs for software packages, preparing projects and confirming needs, installing and modifying baseline products, correcting business internal processes, translating static data, translating transaction data, providing training, and providing related documents.

2.4. RosettaNet

The following summary of RosettaNet related to this paper's context is retrieved from the RosettaNet organization's web site http://www.rosettanet.org. The RosettaNet organization was established as a not-for-profit organization in 1998. It was called by the largest SCM software and service provider—Ingram Micro, and joined by IBM, HP, Intel, and Microsoft. The goal of RosettaNet is to develop and promote global industry SCM standards. It currently includes three committees: Information Technology, Electronic Components, and Semiconductor Manufacturing. RosettaNet focuses on developing dictionaries, frameworks, partner interface processes (PIPs) and e-Business processes for system-to-system business exchange. Under the RosettaNet framework, a dictionary contains the vocabulary used in its dialog of PIPs, such as an order or price query model.

From a process-oriented view, RosettaNet PIP presents a good interface for system-to-system exchange by clustering the core business process of a typical supply chain into eight clusters, each with operational segments that contain corresponding PIP. Fig. 1 shows the eight clusters, namely, RosettaNet support, partner-product and service review, product information,

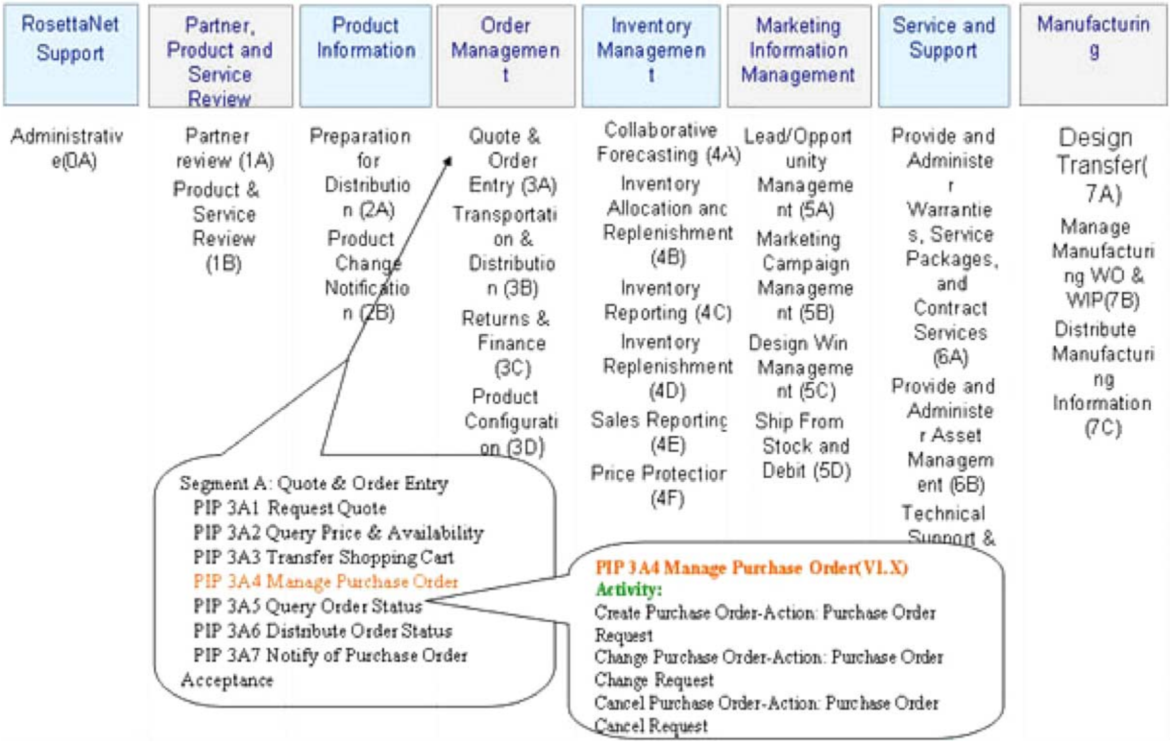

Fig. 1. RosettaNet clusters, segments, and PIPs.

order management, inventory management, marketing information management, service and support, and manufacturing. Segments and PIPs 3A4 for Quote & Order Entry are also included. Fig. 2 illustrates how a buyer makes an order to a product provider via PIPs in segment 3.

Among the coalition, architecture, execution, and solution partners of RosettaNet members, architecture partners physically involve developing the exchange standards, and are responsible for contributing the selected membership fee to the organization, providing feedback on standards development projects, such as PIPs, Dictionaries, and the RNIF, placing the organization's vote, if voting privileges are applicable. In other words, all architecture partners decide what should be included in the standards. Consequently, the RosettaNet standards work effectively in system-to-system information exchange, but not necessary in a concise and efficient way due to the various needs of many architecture partners. Therefore, the RosettaNet DTDs are usually huge and may contain high percentage of attributes not needed or suitable for certain types of information exchange processes.

2.5. Systems integration and collaboration

The technical issues related to heterogeneity are primarily caused by different network technologies, devices and operating systems; middleware solution and communication paradigms; programming languages; services and interface technologies; domains and architectures; and data and documents formats. The problem becomes even worse if the integrated solutions must also cope with non-functional requirements, such as security, availability, and transactions [27].

Stal [27] also pointed out that EAI technologies could not provide a common solution because they tried to solve the problem using an incomplete set of proprietary technologies. Therefore, standardized approach is adopted, in which the technologies remain heterogeneous, but their interface and collaboration

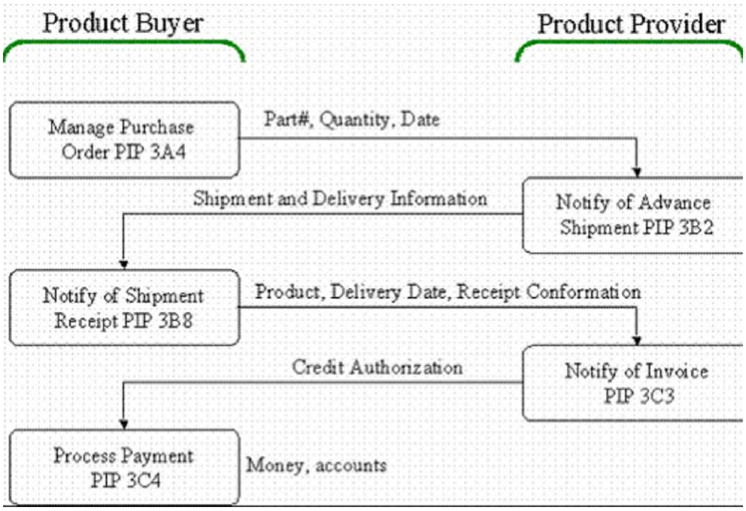

Fig. 2. The quote and order entry process by PIPs in segment 3.

patterns are standardizing using lightweight standards, such as XML and Internet protocols. For example, in their framework of web-based EAI, Pendyala, Shim and Gao [28] used XML as the complete solution for seamless integration of enterprise application using available technologies. However, among the four crucial areas, namely, communication, data, process, and monitoring, their XML Data conversion did not consider external B2B standards, such as RosettaNet.

Nurmilaakso [29] classified eight active XML-based B2B frameworks into three categories: cross-industry, industry-specific, and process-centric frameworks. Cross-industry frameworks, such as cXML, OAGIS and xCBL, provide cross-industry vocabularies but are limited to the rough process approach and do not concern messaging a lot. Industry-specific frameworks, such as papiNet and RosettaNet, provide a comprehensive description of business processes in a particular industry by applying the detailed processes approach. Process-centric frameworks, such as BPML, ebXML, and XPDL, provide no vocabularies but focus on businesses taking the generic process approach.

Transformation between frameworks and versions are not costless, and complete transformation may not be possible in some cases. ebXML, in particular, clearly deals with public business processes between trading partners and puts efforts on messaging and other issues. Because, ebXML aims to be a cross-industry standard that can be developed for use within industry-specific standards, the industry-specific RosettaNet can actually plug into ebXML [29]. Dogac et al. [30] had successfully implemented an ebXML infrastructure through UDDI Registries and RosettaNet PIPs. On the other hand, in the new RosettaNet architecture roadmap, Damodaran [31], the chief technologist at RosettaNet, plans to converge standards between RosettaNet with ebXML and Web Services.

Bussler [32] from Oracle described a B2B integrated technology architecture considering both the B2B and application-to-application (A2A) integration concurrently in one uniform integrated model, in which the designed integration models are stored in the design-time repository as metadata. In the detailed introduction of the B2B engine, Bussler [33] depicted how different standards, such as EDI, RosettaNet, and ebXML, can be accommodated at the same time. In other words, trading partners with different public processes standards, such as RosettaNet and ebXML for two supply chains, can be supported by this B2B integrated engine concurrently.

Gokhale, Schmidt, Natagajan and Wang [34] proposed that component middleware and model-integrated computing (MIC) could be combined to overcome their individual limitations. Middleware is reusable software, such as CORBA, Java 2 Enterprise Edition, and. NET that resides between the applications and the underlying operating systems, network protocol stacks, and hardware. MIC is a development paradigm that systematically applies domain-specific modeling languages to a class of applications.

To sum up, integrations across B2B and A2A are complex and have been evolving rapidly in recent years. Managing heterogeneous technologies is more of an integration issue within an organization, but a collaboration issue across organizations. Intra-organization integration has been realized via standardized middleware, while inter-organization collaboration has been addressed by collaborative B2B standard framework, such as RosettaNet. Moreover, solutions for multiple B2B frameworks integration have been proposed, such as for RosettaNet and ebXML, and Oracle B2B and A2A integration engine.

3. Process model for integrating XML into MES

With an understanding of the systems integration and collaboration issues as surveyed in Section 2.5, the scope of the intended process model for integrating XML into enterprise systems is specified as follows: the process model is addressing primarily the common DTDs for a selected set of enterprise applications following the concept of MIC by Gokhale, Tenenbaum and Meltzer [12], and for RosettaNet-based supply chains. The intra-organization messaging is left unmentioned so that either the middleware, such as IBM MQSeries, or the customized tool, such as the XML-based EAI portal proposed by Pendyala, Shim and Gao [28] can be flexibly incorporated. Because of the targeted RosettaNet standard, the process model is feasible with multiple supply chains as long as they are compatible with the same RosettaNet standard.

With a brief analysis of the implementation methods in Section 2.3, we observed three basic criteria. First, all IS implementation methods should embed the life-cycle development due to the evolving nature of IS history; Second, most implementation methods should share a high percentage of similarities that account for their characteristic of generalization; Third, all similar methods encounter some differences that greatly distinguish one from each other. Accordingly, we propose an implementation process model for XML-based IS applications, which evolves through a four phase life cycle, namely, definition and scope, analysis and design, implementation, and maintenance. It can also be deployed into seven stages, including requirements analysis, initial planning, process analysis and design, system design, system implementation, production, and system maintenance and support, as depicted in Fig. 3. The description in the following sub-sections is MES-centered.

3.1. Requirement analysis

The requirement analysis stage considers data exchange requirement analysis, system evaluation, requirement confirmation, and requirement management, as described below.

3.1.1. Data exchange requirement analysis

The data exchange for MES is classified into two kinds: data exchange between internal information systems and data exchange between two business organizations. The first kind mainly addresses data exchange between enterprise information systems. Therefore, the applicable enterprise systems need to be identified first. The requirements for data exchange between these identified systems can then be analyzed. For example, if only ERP and Advanced Planning System (APS) are interacting with MES for data exchange, the data exchange activities between ERP and MES, and between ERP and APS need to be listed, such as when a WIP lot enters a certain workstation, the MES-WIP tracking module will report the work order to ERP about the states of the workstation.

In the B2B data exchange environment, business organizations are mostly upstream or downstream partners within the same supply chain. It is thus, commonly seen that upstream manufacturers send their parts to downstream partner for manufacturing final products, such as the relationship between a semiconductor foundry and a packing/testing foundry. Thus, the requirements in this situation are mainly the exchange of production data. For example, a semiconductor foundry manufacturer may query about the location of their WIP and the states of the production progress, and MES at a packaging/testing foundry can respond with a states update report.

To meet both the internal and external data exchange requirements, the priority for selecting the data classes or element names is: the external standard

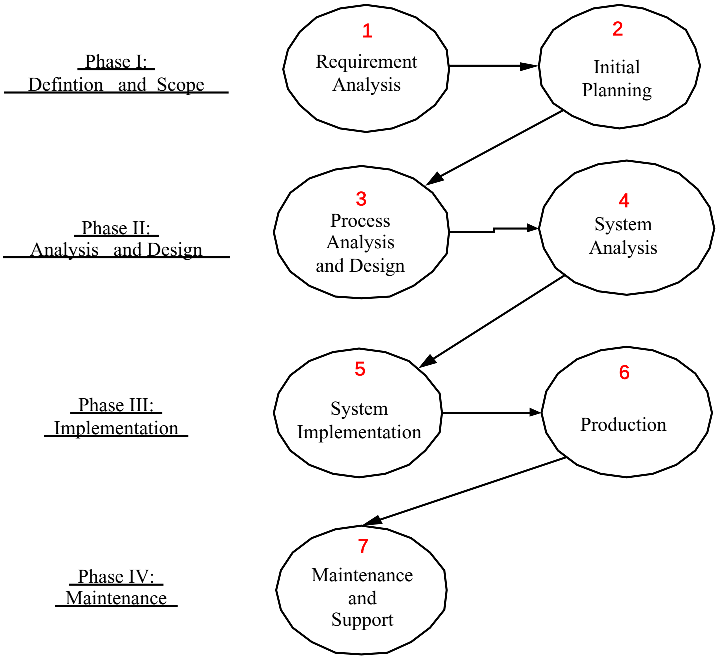

Fig. 3. The implementation process.

first, most common ones among enterprise applications second, and the remaining ones used by the enterprise applications last. The rationale is to follow the most adopted classes and names as much as possible, which will ensure the least conflicts among different inter-/intra-organizational applications.

3.1.2. Data exchange system evaluation

With the results of requirements analysis, a system evaluation will make sure the required environment for data exchange and the adequate capabilities for supporting these requirements.

3.1.3. Data exchange requirements confirmation

Confirmations of the requirements and system capabilities based on the evaluation determine which requirements are validated and verify the validity by certain validation principles.

3.1.4. Data exchange requirements management

In the system life cycle, requirements should be continuously tracked, controlled, and managed, in order to expect proper corrections regarding to any changes.

3.2. Initial planning

The purpose of initial planning is to identify potential risks in organizational software development, and find corresponding strategies for overcoming the risks. This is a way to prevent disastrous results for unexpected effects. Items to be planned include item implementation plan, software development plan, quality assurance plan, software testing plan, document assembling plan, user training plan, and total support plan.

3.3. Process analysis and design

In this stage, all processes related to MES data exchange, either internally or externally, need to be analyzed, and an optimal data exchange style will be designed. For instance, XML requires both parties agree on a set of common DTDs in order to mutually understand the XML contents. In other words, this phase conducts XML document analysis and design for both internal and external needs based on the requirement analysis. The internal requirements can be illustrated as shown in Fig. 4, where the production

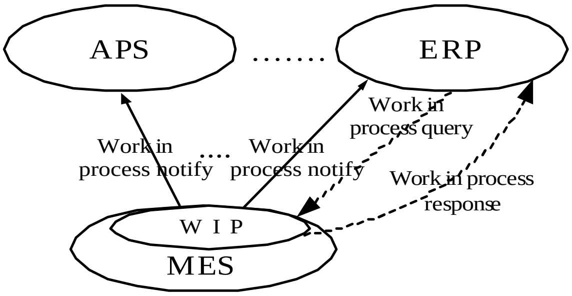

Fig. 4. The intra-organizational MES system data exchange process.

information flows between enterprise WIP, ERP, and SCM. The WIP subsystem will routinely request production states in manufacturing lines from ERP or SCM systems, and ERP can query the WIP database for production batch information in real time.

On the other hand, B2B data exchange addresses the processes between production partners. Fig. 5 illustrates that the semiconductor foundry, such as Taiwan Semiconductor Manufacturing Company, is the upstream partner of the downstream packaging/testing foundry, such as Advanced Semiconductor Engineering Inc. When a semiconductor foundry outsources a work order to its packaging/testing foundry, it continuously monitors production states as the testing/packaging foundry routinely reports updated information.

3.4. System analysis

This stage addresses the system environment where MES operates its XML data exchange. For example, WIP exchanges product data through IBM MQSeries, which requires an adds-on XML translation module on

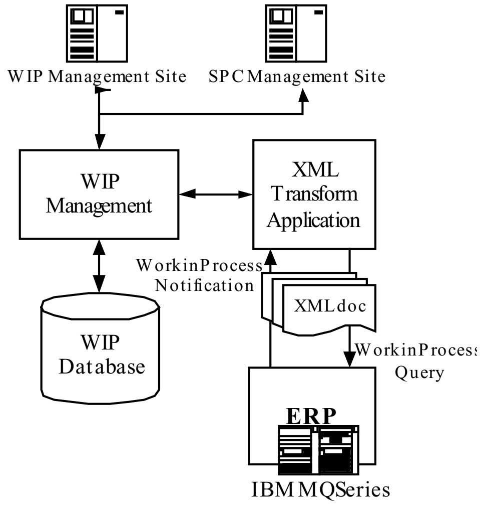

Fig. 6. The framework of enterprise internal system.

the WIP end, in order to translate raw data to an XML format for ERP to receive. Similarly, the ERP system can do the reverse process to translate the XML format into the WIP system format. This translation process is shown in Fig. 6.

Business organizations communicate WIP information through intranet as shown in Fig. 7. The infrastructure on both ends is the same, and follows the same XML translation module as indicated in Fig. 6. But, the information delivery is performed on the Web server, which exchanges data via HTTP post to the Web server of the counterpart.

3.5. System implementation

The system implementation in the XML integration domain refers to the design of DTD specifications and

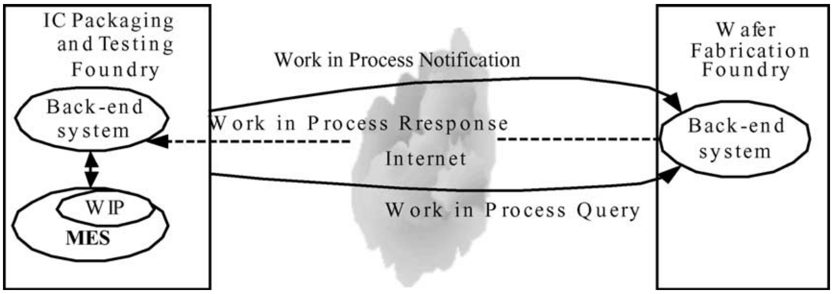

Fig. 5. The B2B MES system data exchange process.

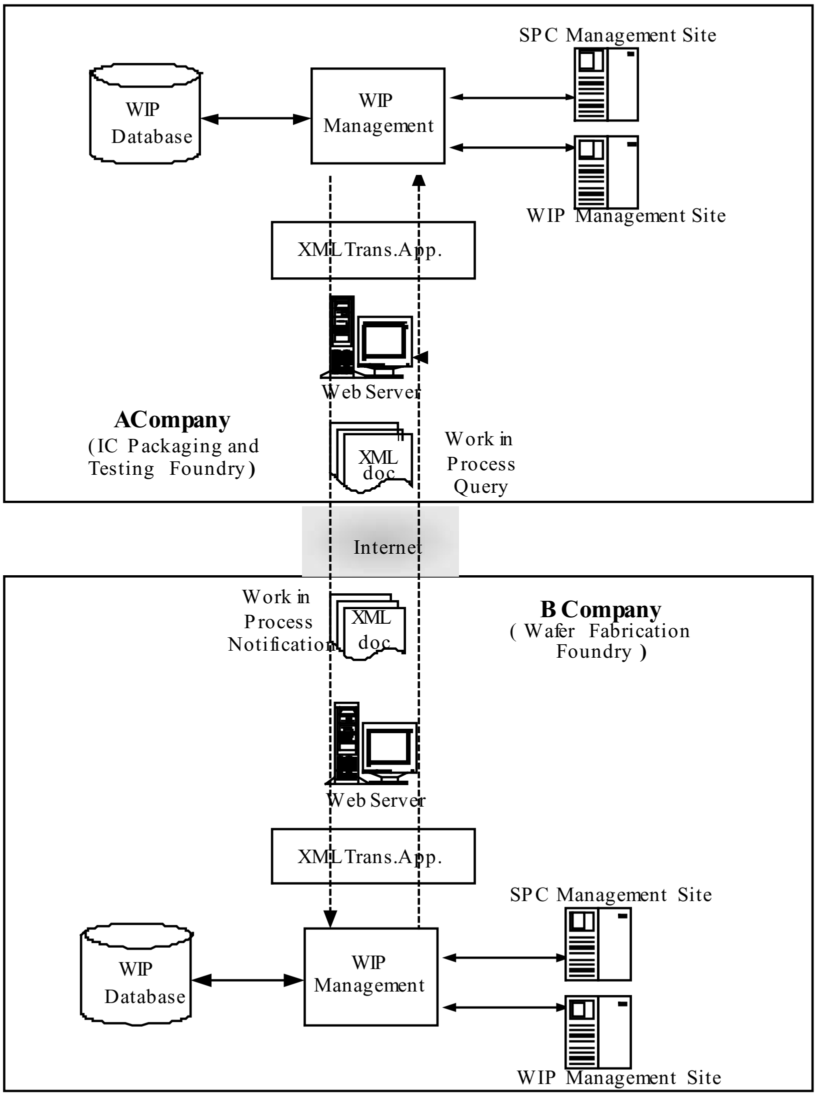

Fig. 7. The framework of B2B system.

the corresponding XML documents. With proper preparation from the above stages, system implementation can have a safe start. The steps involved in system implementation include confirmation of the delivered information, translation of the hierarchical tree, design of the DTD, and production of the XML document.

XML system implementation is the core stage of this process model, and an example will be used to illustrate it in Section 4. There is an important assumption in MIRL-MES decisions, i.e., MIRL-MES needs to meet the international data exchange standard for higher compatibilities among its existing clients and, thus, the RosettaNet standard is followed as well as the internal requirements not discussed in RosettaNet. In order to make sure that DTD meets both requirements internally and externally, the process first fills the requirements of data types that has already been available in RosettaNet, and then adds the remaining data types only required for internal information exchange.

3.6. Production

The goal of this stage is to prepare related operations before the system goes into production so that the system will be stable enough for organizational members to fully operate.

3.7. System maintenance and support

System maintenance includes corrective, adaptive, and comprehensive work. Corrective maintenance corrects any problems due to unexpected input data and interfacing with other software or hardware. Adaptive maintenance makes software adjustments in order to be adapted to the changes caused by external environments. Comprehensive maintenance increases the software usability by expanding its functionalities. To sum up, not only the mistakes or inappropriateness in the original software design need to be modified, but also the functionalities and usability need to be enhanced in this stage repeatedly and continuously.

4. Examples of system implementation

Among the seven steps of the proposed process model, system implementation is the core and the most unique step in the XML integrating process model. In this section a simple example, the work order sent from MIRL-WIP to IBM MQquery, is first presented to demonstrate the complete process in Section 4.1. Then a RosettaNet WIP component is added in Section 4.2, for an easy comparison to the first simple example, and shows how the process may change from the beginning due to the large DTD size of the RosettaNet WIP.

4.1. Simple system implementation—internal components

The simple example described here not only demonstrates how the DTD/XML is implemented, but also serves as a foundation for comparing the integration of RosettaNet in Section 4.2.

4.1.1. Confirmation of delivered information

The information of the work order entering the production line is displayed in Table 1. MIRL-WIP informs ERP through the MQSeries system.

4.1.2. Translation of a hierarchical tree

The work order information is transformed into a hierarchical tree as shown in Fig. 8. With this hierarchical tree, the DTD specification can be written as described next in Section 4.1.3. In Fig. 8, the root element is the work order itself, which branches down to three descendant nodes as the sub-elements, including *work-order*, *material-preparation* and *returned-material*. These sub-elements consist of nine, four and three sub-sub-elements, respectively.

4.1.3. Design of DTD

The hierarchical tree can be easily transformed into the DTD specification as shown in Table 2. Line 1 in Table 2 indicates the version and the number of this DTD document. Line 2 is the root element consisting of three sub-elements, which can be seen from Line 3 to Line 5, respectively. The elements after Line 6 are called the basic elements because they do not have any descendant elements. The notation (#PCDATA) next to the element name indicates that this value can be analyzed as a text instead of a Markup or an element.

4.1.4. Production of an XML document

The XML document can be constructed according to the DTD specification from Section 4.1.3. Table 3 is

Table 1
The work order data

Type	Date field
WorkOrderData	[A] WorkOrderNum, FactoryNum, ORG, ProductNum, WorkOrderQuantity, ProcessNum, StandardTotalCycleTime, [InsideOutsideOrder], WorkOrderType
MaterialPreparationData	[B] WorkOrderNum, MaterialNum, Quantity, OperationNum
ReturnedMaterialData	[C] WorkOrderNum, MaterialNum, Quantity

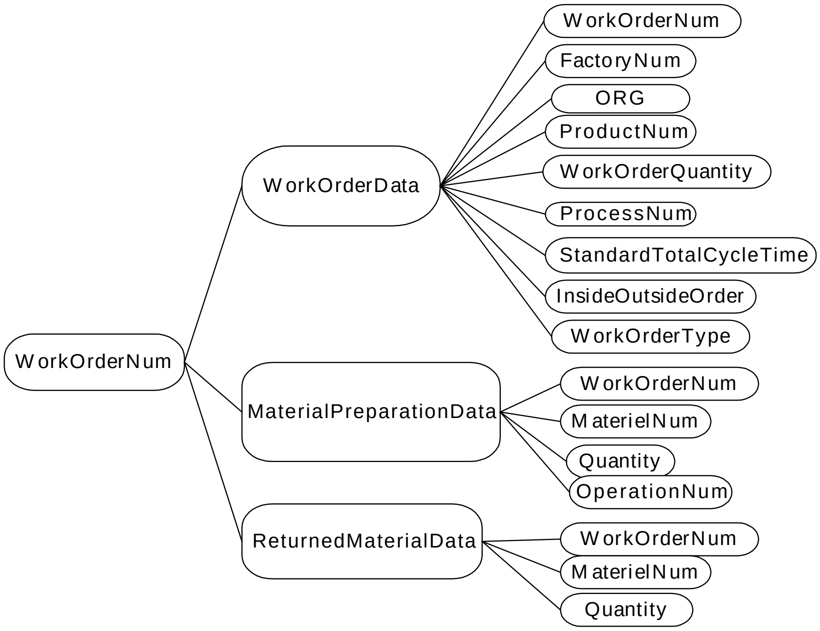

Fig. 8. The hierarchical tree.

a sample XML document based on the DTD specification in Table 2. Line 1 in Table 3 indicates that the norm of XML version 1.0 is used for describing this XML document and the Big5 code is chosen. 'Standalone = no' represents this document will call up other documents. In this case, the file workorder.dtd as listed in Line 2 is called. After Line 3 is the data area, which is the main content of this data exchange activity.

The above example is a most straightforward case in designing the DTD for enterprise data exchange. It can be easily expanded to include RosettaNet-defined DTD in the first two steps assuming RosettaNet DTD is considered in the first place. The last two

Table 2
The DTD specification

1	<?xml version="1.0" encoding="big5"?>
2	<!ELEMENT OrderNum (WorkOrderData, MaterialPreparationData, ReturnedMaterialData)>
3	<!ELEMENT **WorkOrderData** (WorkOrderNum, FactoryNum, ORG, ProductNum, WorkOrderQuantity, ProcessNum, StandardTotalCycleTime, InsideOutsideOrder, WorkOrderType)>
4	<!ELEMENT **MaterialPreparationData** (WorkOrderNum, MaterialNum, Quantity, OperationNum)>
5	<!ELEMENT **ReturnedMaterialData** (WorkOrderNum, MaterialNum, Quantity)>
6	<!ELEMENT WorkOrderNum (#PCDATA)>
7	<!ELEMENT FactoryNum (#PCDATA)>
8	<!ELEMENT ORG (#PCDATA)>
9	<!ELEMENT ProductNum (#PCDATA)>
10	<!ELEMENT OrderQuantity (#PCDATA)>
11	<!ELEMENT ProcessNum (#PCDATA)>
12	<!ELEMENT StandardTotalCycleTime (#PCDATA)>
13	<!ELEMENT InsideOutsideOrder (#PCDATA)>
14	<!ELEMENT WorkOrderType (#PCDATA)>
15	<!ELEMENT MaterialNum (#PCDATA)>
16	<!ELEMENT Quantity (#PCDATA)>
17	<!ELEMENT OperationNum (#PCDATA)>

Table 3
The XML document

```
<?xml version="1.0" encoding="big5" standalone="no"?>
<!DOCTYPE WorkOrderNum SYSTEM "E:\XML_project\final_report\workorder.dtd"> <WorkOrderNum>
<WorkOrderData>
  <WorkOrderNum>W042001-67-001</WorkOrderNum>
  <FactoryNum>One factory</FactoryNum>
  <ORG>TEST</ORG>
  <ProductNum>PCB001-23</ProductNum>
  <WorkOrderQuantity>500</WorkOrderQuantity>
  <ProcessNum>3</ProcessNum>
  <StandardTotalCycleTime>1000</StandardTotalCycleTime>
 <InsideOutsideOrder>0</InsideOutsideOrder>
  <WorkOrderType>1</WorkOrderType>
</WorkOrderData>
<MaterialPreparationData>
  <WorkOrderNum>W042001-67-002</WorkOrderNum>
  <MaterielNum>A-0002</MaterielNum>
  <quantity>200</quantity>
  <OperationNum>FT2C</OperationNum>
</MaterialPreparationData >
<ReturnedMaterialData >
  <WorkOrderNum>W042001-67-003</WorkOrderNum>
  <MaterielNum>A-0003</MaterielNum>
  <Quantity>100</Quantity>
</ReturnedMaterialData >
</OrderNum>
```

steps remain unchanged in this external data exchange scenario.

4.2. A comparative example with RosettaNet WIP

As mentioned in Section 3.1, the priority for selecting the data classes or element names is: the external standard first, most common ones among enterprise applications second, and the remaining ones used by the enterprise applications last. The work order data listed in Table 1 contains only three simple classes and their corresponding elements recognized by the MIRL-WIP and the ERP system. To make Table 1 more compatible with the external RosettaNet, we used the RosettaNet business dictionary version of October 2002 and the PIP version of August 2002 to discuss the design changes through some scenarios as shown below. Since, the step-by-step process is the same as that in Section 4.1 and only the delivery information changes, the following discussion only shows the changed information specification.

4.2.1. Element names

Assume a scenario of the example in Section 4.1 has no counterpart in the RosettaNet dictionary in the class level. Consequently, the design changes can only be on the elements as seen in Table 1. Table 4 shows some

Table 4
The element changes in Table 1

Entity	Data element
WorkOrder	[A] WorkOrderNum**ber**, ***ManufacturingSite***Num, ORG, **Golbal**Product**Identifier**, WorkOrderQuantity, Process**TypeCode**, StandardTotalCycleTime, [InsideOutsideOrder], WorkOrderTypeCode
MaterialPreparation	[B] WorkOrderNum, Material**TypeCode**, ***MaterialPrepataion***Quantity, OperationNum
ReturnMaterial	[C] WorkOrderNum, Material**TypeCode**, ***ReturnMaterial***Quantity

Table 5
The class changes in Table 4

Entity	Data element
WorkOrder	[A] **WorkOrder of Rosetta's 280-item data elements**, StandardTotalCycleTime
MaterialPreparation	[B] WorkOrderNum, MaterialTypeCode, *MaterialPrepataion*Quantity, OperationNum
ReturnMaterial	[C] WorkOrderNum, MaterialTypeCode, *ReturnMaterial*Quantity
WorkInProcess	[D] **WorkInProcess of Rosetta's 160-item data elements**

element changes from Table 1 to illustrate a few change scenarios:

1. *Exactly matched element names*: WorkOrderNumber is defined in RosettaNet and, thus, replaces the original "WorkOrderNum". Similarly, ProductNum is changed to "GlobalProductIdentifier".
2. *Partially matched element names*: Class name ReturnedMaterialData is changed to ReturnMaterial without the "ed" at the end of "Return" since ReturnMaterial is the official usage in the RosettaNet dictionary. FactoryNum in area A is changed to ManufacturingSiteNum since "ManufacturingSite" in the RosettaNet dictionary is compatible to "Factory" in this case. Similarly, ProcessNum is changed to ProcessTypeCode
3. *Self-explicit style*: Some element or class names maybe appropriate but inadequate to precisely represent their contexture meanings. For examples, all the three class names in Table 1 are changed to exclude "Data" at the end since it is redundant. On the other hand, "Quantity" in the B and C areas is changed to "MaterialPrepration-Quantity" and "ReturnMaterialQuantity" to reflect its precise purpose and avoid ambiguity.

4.2.2. Class names

Changing a class name may actually change the internal elements of that class. It is thus more complicated in terms of change design at the thought level, which also involves RosettaNet PIPs. In the following case, PIPs in Rosetta 7B Segment of Manufacturing Managing WO & WIP are referenced. Table 5 shows some revised results from Table 4 to illustrate a few change scenarios:

1. *Replacement only*: The WorkOrder DTD is well defined in RosettaNet as a 280-item hierarchical data type. If every element of the WorkOrderData class in Table 1 is included in RosettaNet's WorkOrder data type or those not included can be ignored, then WorkOrder can be completely replaced by RosettaNet's version, except with more elements than needed.
2. *Replacement with modification*: WorkOrderData in Table 1 contains some elements not in the RosettanNet's WorkOrder DTD. If, for instance, the element StandardTotalCycleTime is too important to be ignored in this case, then RosettaNet's WorkOrder DTD will replace the WorkOrderData class while keeping StandardTotalCycleTime. ReturnedMaterialData is equivalent to ReturnMaterialInformation at the first glance. However, RosettaNet's BusinessDataEntity Dictionary defines ReturnMaterialInformation as the "information about the authorization, billing and shipping of material to be returned" due to the natural usage between B2B exchange, which is different from the ReturnMaterialData used at a production line.
3. *New addition*: New classes can be added to a DTD. For example, if the WIP DTD is needed in this case, Table 1 can add the WIP DTD defined in Rosetta 3D8 PIP, which is a 136-item hierarchical data type.

5. Conclusions

A process model for integrating XML into an enterprise application has been proposed in this paper. Although, the MES system at MIRL is used to illustrate the proposed process model, the process model is claimed to be a generalized methodology. With the RosettaNet standard in mind, MES can be easily replaced by other enterprises applications in maintaining its generalization characteristic. However, the external standard RosettaNet can be replaced, but with more modification in the analysis and design stages as illustrated in Section 4.2.

Acknowledgements

This research is sponsored by the Mechanical Industry Research Laboratories of Taiwan, R.O.C. under Grant No 903K31BH1. The authors would also like to thank the anonymous referees for their valuable suggestions in improving the readability of this paper.

References

[1] M. Sundaram, S.S.Y. Shim, Infrastructure for B2B exchanges with RosettaNet, in: Proceedings of the Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, San Juan, California, June 21–22, 2001, pp. 110–119.

[2] Intel e-Business Group, Accelerating RosettaNet Implementing-Key Learning from the Intel Interoperability Fest, White Paper, http://www.intel.com/eBusiness/pdf/ar020305.pdf, November 1, 2001.

[3] R.L. Nolan, Managing the Crises in Data Processing, Harvard Business Review, March–April, 1979, pp. 115–126.

[4] J. Ward, J. Peppard, Strategic Planning for Information Systems, third ed., Wiley, 2002.

[5] T. Berners-Lee, W3C Data Formats [W3C NOTE], http://www.w3.org/TR/NOTE-rdfarch, October 1997.

[6] J. Roy, A. Ramanujan, XML: Data's Universal Language, IT Pro, May 2000, pp. 32–36.

[7] C.-H. Lu, National standard development for XML, http://www.tips.org.tw/nii/storehouse/data-exchange/89-1.htm#2, December 29, 2000.

[8] C.-N. Cheng, I.-C. Chen, Application of Web XML, Unalis Corporation, October 1999.

[9] S.M. Huang, I. Kwan, D.C. Yen, S.Y. Hsueh, Developing an XML gateway for business-to-business commerce, in: Proceedings of the First International Conference on Web Information Systems Engineering, Hong Kong, China, June 19–20, 2000, pp. 67–74.

[10] J.Y. Shiau, S.M. Ratchev, G. Valtchanov, Distributed collaborative design and manufacturability assessment for extended enterprise in XML-based agent system, in: Proceedings of the IEEE Ninth International Workshop on Enabling Technology: Infrastructure for Collaborative Enterprises, Gaithersburg, Maryland, March 14–16, 2000, pp. 260–265.

[11] R.J. Glushko, J.M. Tenenbaum, B. Meltzer, An XML framework for agent-based e-commerce, Communication of the ACM 42 (3) (1999) 106-ff.

[12] C. Ziao, G.-L. Cheng, H.-W. Zen, G.-H. Chen, Exploring the interactive models of IC supply chain manufactures referencing RosettaNet, Mechanical Industry Magazine, December 2000. pp. 138–154.

[13] J. Roy, A. Ramanujan, XML Schema Language: Taking XML to the Next Level, IT Pro, March/April 2001, 37–40.

[14] J. Evdemon, XML Schemas: going beyond validation, presentation at XML Asia Pacific, available at http://www.kku.ac.th/chugra/xml/XMLSchema.pdf, 2000.

[15] K. Parker, Marking the business case for manufacturing execution system, Manufacturing Systems, July 1992, pp. 18–23.

[16] B.J. Stephen, Closing the execution gap, Industrial Computing, May 1995.

[17] L.-S. Yeh, Translated, Integrated manufacturing information system for manufacturing plants, Technology of Chemical Engineering 7 (9) (1999) 214–220.

[18] MESA International, Execution-driven manufacturing management for competitive advantage, White Paper, No 5, 1997.

[19] MIRL Web Page, MES System Introduction, http://www.mirl.itri.org.tw/rd/automation/eb-cim/MES/2.asp, 2003.

[20] MIRL Web Page, Automation technology, http://www.mirl.itri.org.tw/rd/automation/index.asp, 2003.

[21] SAP, Accelerated SAP roadmap rel. 3.1 SAP Reference Manual, 1997.

[22] Y.M. Chen, T.H. Tsao, A structure methodology for implementing engineering data management, Robotic and Computer-Integrated Manufacturing 14 (1998) 275–296.

[23] I.J. Chen, M.H. Small, Implementing advanced manufacturing technology: an integrated planning model, Omega International Journal Management Science 22 (1) (1994) 91–103.

[24] R.B. Cooper, R.W. Zmud, Information technology implementation approach, Management Science 36 (2) (1990) 123–139.

[25] R.W. Zmud, M.E. Apple, Measuring information technology infusion, 1989 (unpublished manuscript).

[26] J.D. Launi, A structured methodology for off-the-shelf software implementation, Journal of Systems Management 42 (10) (1991) 6–9.

[27] M. Stal, Web services: beyond component-based computing, Communications of the ACM 45 (10) (2002) 71–76.

[28] V.S. Pendyala, S.S.Y. Shim, J.Z. Gao, An XML-based framework for enterprise application integration, in: Proceeding of the IEEE International Conference on E-Commerce, Newport Beach, California, June 24–27, 2003.

[29] J. M. Nurmilaakso, XML-based supply chain integration: a review and a case study, Thesis of Department of Computer Science and Engineering, Helsinki University of Technology, 2003 (unpublished).

[30] A. Dogac, Y. Tambag, P. Pembecioglu, S. Pektas, G. Laleci, G. Kurt, S. Toprak, Y. Kabak, An ebXML infrastructure implementation through UDDI registries and RosettaNet PIPs, in: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, June 4–6, 2002. pp. 512–523.

[31] S. Damodaran, RosettaNet architecture roadmap, available at http://www.eidx.org/conference/presentations/EIDX-RosettaNet-Arch-Roadmap-Aug-2003.pdf, August 2003.

[32] C. Bussler, B2B integration technology architecture, in: Proceedings of the Fourth International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, Newport Beach, CA, U.S.A., June 2002.
[33] C. Bussler, The role of B2B engines in B2B integration architectures, SIGMOD Record 31 (1) (2002) 67–70.
[34] A. Gokhale, D.C. Schmidt, B. Natagajan, N. Wang, Applying model-integrated computing to component middleware and enterprise applications, Communications of the ACM 45 (10) (2002) 65–70.

Yu-Hui Tao is an associate professor of Information Management in National University of Kaohsiung. He received his Ph.D. degree in Industrial and Systems Engineering from The Ohio State University in 1995. From 1994 to 1997, he was with Bank One Credit Card Company as a senior software consultant. Then, he became an assistant professor and an associate professor at the Department of Information Management in I-Shou University from 1997 to 2003. His current research interests include e-business, management information systems, and Internet applications. Dr. Tao is a member of the Chinese Association of Information Management, The Association for Computational Linguistics and Chinese Language Processing, and Chinese Maritime Research Institute.

Tzung-Pei Hong is a professor at the Department of Electrical Engineering and the Dean of Academic Affairs in National University of Kaohsiung in Taiwan, R.O.C. He received his B.S. degree in chemical engineering from National Taiwan University in 1985, and his Ph.D. degree in computer science and information engineering from National Chiao-Tung University in 1992. From 1987 to 1994, he was with the Laboratory of Knowledge Engineering, National Chiao-Tung University. Then he was an associate professor at the Department of Computer Science in Chung-Hua Polytechnic Institute from 1992 to 1994, and at the Department of Information Management in I-Shou University from 1994 to 1999. He became a professor in I-Shou University from 1999 to 2001. He was in charge of the whole computerization and library planning for National University of Kaohsiung in Preparation from 1997 to 2000 and served as the first director of the library and computer center in National University of Kaohsiung from 2000 to 2001. His current research interests include parallel processing, machine learning, data mining, soft computing, management information systems, and www applications. Dr. Hong is a member of the Association for Computing Machinery, the IEEE, the Chinese Fuzzy Systems Association, the Taiwanese Association for Artificial Intelligence, and the Institute of Information and Computing Machinery.

Sheng-I Sun is a system analyst in charge of the whole sales order system in InfoChamp System Corporation in Taiwan, R.O.C. He received his B.S. in Information Management and Master degree in Information Engineering from the I-Shou University in 2000 and 2002, respectively. From 2000 to 2003, he was a part-time instructor in Extention Education Center at I-Shou University. Mr. Sun has involved several software development projects from private sectors and government agencies over the past few years. His current research interests include e-business, data mining, sales forecast, ERP system, SCM systems, and Web programming. Mr. Sun is a member of the Phi Tau Phi scholastic honor society of R.O.C.