Suo Tan
e-mail: suo_tan@encs.concordia.ca

Yong Zeng[1]
e-mail: yong.zeng@concordia.ca

Concordia Institute of Information
Systems Engineering,
Faculty of Engineering & Computer Science,
Concordia University,
Montreal, QC,
H3G 1M8, Canada

Bo Chen
Minicut International Inc.,
Montreal, QC,
H1J 2YS, Canada
e-mail: chenbo@gmx.net

Hamzeh K. Bani Milhim
e-mail: h_banimi@encs.concordia.ca

Andrea Schiffauerova
e-mail: andrea@encs.concordia.ca

Concordia Institute of Information
Systems Engineering,
Faculty of Engineering & Computer Science,
Concordia University,
Montreal, QC,
H3G 1M8, Canada

Environment Based Design Approach to Integrating Enterprise Applications

Enterprises tend to depend on various legacy applications in supporting their business strategies and in achieving their goals. In order for an enterprise to be efficient and cost-effective, their legacy applications should be seamlessly integrated within and beyond the enterprise. Some research work in enterprise applications integrations (EAI) analyzed the problem, while others proposed solution models for the syntactic and semantic integration of business processes. In this paper, the EAI is considered as a design problem and is analyzed from design point of view. Environment based design (EBD) methodology is applied to handle the integration problem by analyzing and clarifying the design requirements to generate appropriate solutions. A framework is proposed for EAI problems based on the EBD approach. A case study is also provided to show how the approach can be applied within a company to generate satisfactory EAI solutions with low cost, high efficiency, and enhanced scalability. [DOI: 10.1115/1.4007171]

1 Introduction

Along with the rapid development of information technologies, various applications have been developed in almost every aspect of enterprise business processes, such as supply chain management (SCM), customer relationship management (CRM), enterprise resource planning (ERP), and product lifecycle management (PLM), in order to enable organizations to improve their use of information systems in supporting their operational and financial goals [1–3]. A SCM application focuses on co-ordination and inter-enterprise supply chain upstream and downstream relationships [2]. A CRM application supports organizations to improve their front-end operations regarding the value they offer to their customers [4]. An ERP application is a software package that attempts to integrate all the organizations' departments and functions into a single system to serve the departments' needs [5]. Most of these applications have been built with very specialized focus on specific business needs and are often developed without thorough consideration of how to share with other existing applications. With such a great quantity and variety of applications, there is a growing need for facilitating the data sharing among them. Each individual application is often upgraded independently from others that might be developed by different IT companies. As a result, different applications with the similar functions or several generations of the same application may co-exist in an enterprise. Inefficient data sharing among those applications in a business process inhibits seriously the productivity of the enterprise. This situation is mostly due to the complexity and the multidisciplinary nature of the business processes [6]. It was estimated that more than 30% of IT dollars were spent to link different systems together [6]. The IT industry has been trying to bring competitive advantages to businesses by working on various EAI solutions. It is a challenging task to integrate an individual application into other application platforms in a manner that fits into the general business logic existing in an enterprise.

EAI has emerged to address the demands of horizontal and vertical integrations, within and beyond enterprises, in more flexible and maintainable way [7,8]. Irani [9] stated that "EAI addresses the need to integrate both intra and interorganizational systems." This can be achieved by incorporating functionality from different legacy applications in use. Linthicum [6] defined EAI as "unrestricted sharing of data and business processes among any connected applications and data sources in the enterprise." Du et al. [10] considered that "the integration of applications enables information sharing and business processes, both of which result in efficient operations and flexible delivery of business service to customers." Chen et al. [11] gave a similar definition to enterprise integration as "enterprise integration is the process of ensuring the interaction between enterprise entities necessary to achieve domain objectives." Chen et al. [11] indicated that enterprise integration can be approached in various manners and at various levels, such as (1) physical integration (interconnection of devices, NC machines via computer networks), (2) application integration (integration of software applications and database systems), and (3) business integration (co-ordination of functions that manage, control and monitor business processes). The scope of implementing EAI can be classified into three major components according to Ref. [9]: (1) intra-organizational component that integrates packaged and custom systems, (2) interorganizational application integration that incorporates cross-enterprise business processes and systems throughout a supply chain, and (3) hybrid application integration component that integrates business-to-customer applications with other interorganizational solutions.

[1] Corresponding author.

Contributed by the Computers and Information Division of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received August 23, 2011; final manuscript received April 18, 2012; published online August 6, 2012. Assoc. Editor: Ashok K. Goel.

The major business objectives of EAI implementations are to improve productivity, to facilitate data sharing, to achieve greater agility and flexibility in supporting business processes integration, to avoid trusted legacy systems replacements, and to solve the issues related to organizational mergers. In this paper, integrations from all levels are considered as a whole enterprise-wide application integration problem.

The research conducted in the EAI area emphasized on many issues, including the semantic and syntactic issues, the analytical issues, and the technical issues. In Liu et al's state of the art review [12], they discussed the EAI from two generic points of view: semantic and syntactic integration. In their review, they concluded that the current EAI solutions do not satisfy the manufacturing companies' needs and that organizations should focus more on semantic integration. Sharif et al. [13] proposed an EAI impact framework to address research challenges in EAI area. However, the framework itself is complex and is hard to implement for an EAI solution. Themistocleous [7] classified the major EAI drivers to eight major factors: barriers, benefits, costs, external and internal pressures, support, IT sophistication, evaluation framework for integration technologies, and assessments of EAI packages and IT infrastructure. Compared to Ref. [7], Lam [14] investigated the critical success factors for EAI from administrative perspective and categorized them into three major groups: (1) top management support, (2) overall integration strategy, and (3) EAI project planning and execution. Vasconcelos et al. [15] verified the dimensions that should be represented as part of information systems architecture of EAI. Scheibler et al. [3] introduced a platform independent approach to modeling, describing, and enacting EAI patterns in service oriented architecture. Losavio et al. [16] modeled the EAI based on Brown's conceptual model of integration, which was extended to obtain more unified and organized views in EAI by Sandoe and Saharia [17]. Kamal et al. [18] evaluated the implementation of EAI solutions to Welsh Local Government Association by distinguishing between the factors of adoption and nonadoption of the EAI. Puschmann and Alt [1] presented an implementation of EAI in Robert Bosch Group and addressed the need for standardized integration architecture. In addition, studies on customized EAI solutions attracted a lot of researchers. For example, Nia et al. [19] proposed a collaborative engine to facilitate application integration for tooling companies. With this engine, both technical and business process integration requirements were satisfied. Also, security issue in information sharing was addressed in the proposed integration solution. While some of the studies discussed the design issues of the EAI solutions, Gleghorn [20] suggested that the design objectives of the integration solutions are configurable, extensible, and reusable. Lublinsky [21] presented a procedure to design an EAI solution: define and catalog business processes, business events, IT components, process rules and messages, and message contents. Umapathy et al. [22] proposed a study on conceptual design of enterprise integration solutions, which indicated that there are two main prerequisites for the design process: an understanding of the design requirements represented in business process models, and the expertise that professionals brought to the project as design knowledge acquired from enterprise integration patterns. Johannesson and Perjons [8] suggested a methodological support for modeling the alignment of application integration to business processes, and some design guidelines for design, validation, and presentation of applications integration were given. It fell into two groups: the first group to assist the designer in obtaining different views of the model, while the second group to check the completeness of the process diagram.

Although research has been done for specific problems in EAI, enterprises are still faced with many problems. Enterprises tend to depend on various legacy applications in supporting their business strategies and in achieving their goals by implementing software applications for special requirements of different functional sectors, which are mutually independent at most of the time [10]. Enterprises need to integrate their business functions into a single system, in order to be able to adopt the changes in their environments and in the technology development. Traditional approaches, such as electronic data interchange, database-oriented middleware, and distributed objects (e.g., CORBA, DCOM, etc.) technology, do not fully automate the desired integration [7,9]. Innovative solutions are needed to achieve this goal more easily. It leads to new integration technologies supporting effective EAI solutions, such as message brokers, process brokers, middleware systems (e.g., SOAP, J2EE, etc.), adapters, wrapper's API, webservices (e.g., W3C, WebServices.or, etc.). Most of the commercial integration products and services fall into two basic categories: hosted integration solutions and server-based middleware solutions [6,9,10,12,20]. There are many EAI solutions in the market, such as "Vitrea," "ActiveSoftware," "CrossWorld," and many others [14,23]. Meanwhile, many integration tools have been developed by different vendors, such as "BizTalk Server 2004" developed by Microsoft, "WebSphere MQ" developed by IBM and several others [24]. However, enterprises are forced to give up their legacy applications and invest a huge amount of money to buy those new integrated solutions in order to manage their business. This is not necessarily to be the only option. The reasons lie mainly in three aspects: first, from the implementation point of view, different enterprises may have different EAI requirements, which come from various applications and variety of business data and processes. It is practically impossible to achieve all those requirements in one EAI system [19,22]. Different applications need different integration methods. Second, many applications have their own built-in functions for development, e.g., VBA, ACTIVEX, AUTOCAD script, and API for AUTOCAD. However, it is often costly and time consuming to integrate enterprise applications by using those built-in functions, which requires not only the understanding of different built-in functions for different systems but also the internal data and process structures within each related applications. Third, most of existing EAI solutions are focused on answering questions, such as "what do we need to do to integrate the various applications?" instead of considering the implementation process as the sum of existing applications and the context of the business environment [6,12].

In this paper, EAI problems are discussed from the design point of view. By applying EBD methodology to EAI problems, a framework has been proposed. Through a case study, a low cost and high efficiency EAI solution were developed to validate the framework. The rest of the paper is organized as follows: Sec. 2 introduces the EBD methodology, followed by the framework and the case study to show the implementation process of the EBD to generate an integration solution. Evaluation of the generated solution is addressed thereafter. Sec. 4 concludes the paper.

2 Environment-Based Design

Apart from traditional design methodologies, the EBD methodology [25–27] was logically derived from the axiomatic theory of design modeling [28], which was founded on the recursive logic of design [29]. The basic idea is that a design problem is implied in a product system and is composed of three parts: the environment in which the designed product is expected to work, the requirements on product structure, and the requirements on performance of the designed product. The requirements on product structure and performance are related to the product environment. In addition, the product environment includes three major environments: natural, built, and human. As illustrated in Fig. 1, the EBD includes three main activities: environment analysis, conflict identification, and solution generation. These three activities work together progressively and simultaneously to generate and refine the design specifications and design solutions.

The objective of environment analysis is to find out the key environment components, in which the product works, and the relationships between the environment components. From the environment implied in the design problem described by the customer(s), a designer will introduce extra environment components that are relevant to the design problem at hands. The results from

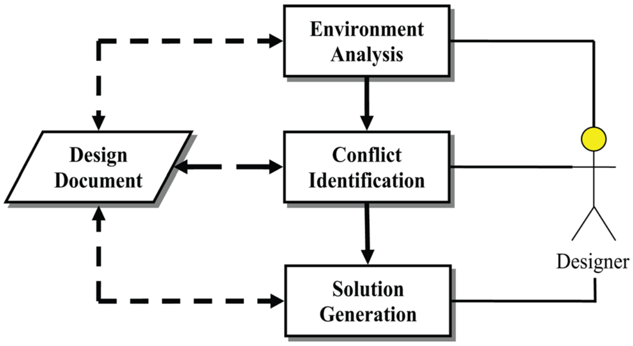


Fig. 1 Environment-based design: process flow [27]

this analysis constitute an environment system. One of the key methods for environment analysis is linguistic analysis [30]. The recursive object model (ROM) was proposed by Zeng [31] to conduct this work. The ROM includes two types of objects, which are object and compound object, and three kinds of relations between any two objects: connection, constraint, and predicate (as shown in Table 1).

While a ROM diagram is generated, some questions should be asked to clarify every object in the ROM diagram. Wang and Zeng [32] gave the rules on question asking to conduct a comprehensive environment analysis. In order to verify the completeness of the extracted environment components and their relations, a roadmap was proposed as guidance for requirements modeling [33]. In this roadmap (see Fig. 2), requirements (structural or performance) are categorized by two criteria in terms of different partitions of product environment: One criterion classifies the product requirements by partitioning product environment in terms of product lifecycle (refer to Fig. 2(*a*)) and the other one classifies them by partitioning the product environment into eight levels (refer to Fig. 2(*b*)). The eight levels can be grouped into the natural, built, and human environments for better extraction of the environment components. Following the pattern of such environment analysis, conflicts could be identified among the relations between environment components [26,27]. There are several rules to follow for conflict identifications [27,34]:

Rule 1 *If an object has multiple constraints, then potential conflict exists between any pair of constraining objects.*
Rule 2 *If an object has multiple predicate relations from/to other objects, then potential conflict exists between a pair of those predicate relations.*
Rule 3 *If an object is constrained by another object, then the relation is inherited by its subobjects, e.g., an object O_1 is constrained by another object O_2, and O_1 has two components O_{11} and O_{12}. O_{11} and O_{12} are considered being constrained by O_2.*

Figure 3 shows three forms for a possible conflict existing in a ROM diagram. `A`, `B1`, `B2`, `R1`, and `R2` are existing objects

Table 1 Elements defined for the ROM [31]

	Type	Graphic representation	Description
Object	Object	$\boxed{O}$	Everything in the universe is an object
	Compound object	$\boxed{\boxed{O}}$	It is an object that includes at least two objects in it
Relation	Constraint relation	●— $\boxed{\xi}$ →	It is a descriptive, limiting, or particularizing relation of one object to another
	Connection	- - $\boxed{\iota}$ - →	It is to connect two objects that do not constrain each other
	Predicate relation	— $\boxed{\rho}$ →	It describes an act of an object on another or that describes the states of an object

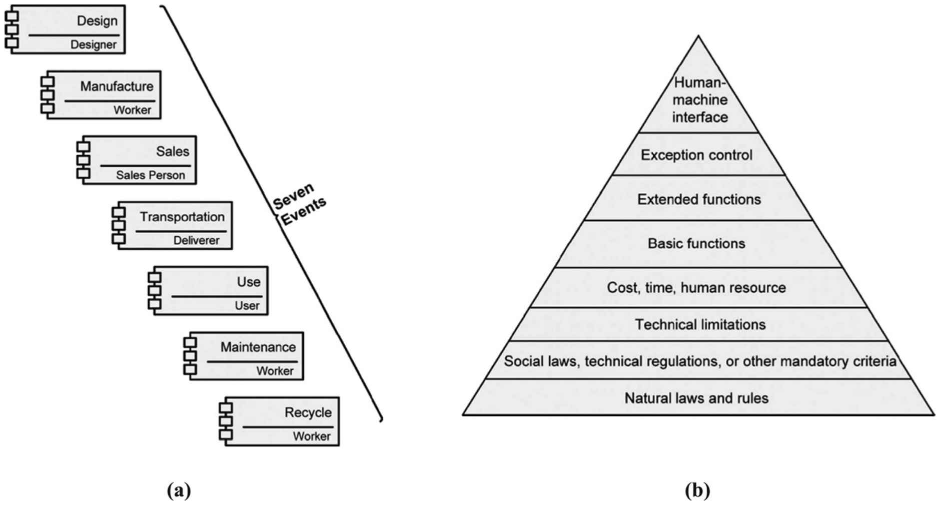


Fig. 2 A roadmap for environment analysis [33]

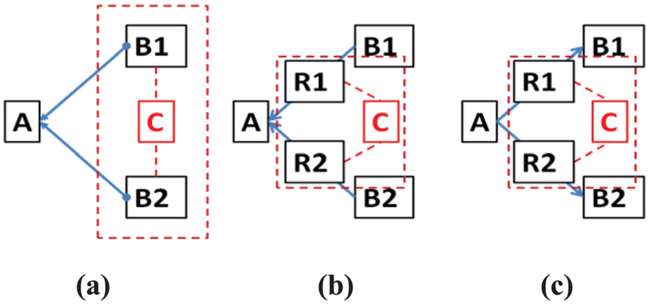


Fig. 3 Three forms of an existing conflict in a ROM diagram [27,34]

(relations are seen as objects as well [28]). C stands for a possible conflict. A is called the resource object, B1 and B2 or R1 and R2 are two competing objects [34]. Specifically, Fig. 3(*a*) shows a possible conflict between two constraints—resulted from conflicting constraint relations. Figures 3(*b*) and 3(*c*) show two other forms—resulted from predicate relations. Please notice the different directions of predicate relations in Figs. 3(*b*) and 3(*c*). How to apply them to practice is shown in the case study later. At the third stage of the EBD, a set of key environment conflicts will be chosen to be resolved by generating some design solutions. This process continues until no more unacceptable environment conflicts exist.

Customers' requirements are commonly described in natural language. Compared to other approaches to EAI (function-oriented, operation-oriented, business process-oriented, etc.), the EBD is easy to start and to follow due to its natural language processing ability and well-defined rules. Through the recursive process of the EBD along with the ROM, implicit requirements can be revealed. In addition, the lifecycle of an EAI solution is considered, semantic and syntactic integration issues and technology issues are focused at the design stage, which will definitely enhance the reliability of the final solution, at least with less design defects. All of those benefits will be illustrated in Sec. 3.

3 A Framework to EAI Using EBD

3.1 The General Framework. Since the purpose of EAI is to seamlessly integrate various kinds of applications into a business process to increase the employees' work efficiency and to promote the company's business effectiveness, we start by drawing a ROM diagram as shown in Fig. 4. The environment analysis will then be conducted based on the initial ROM diagram. Obviously, "integrate" is the most constrained object as a predicate relation is a bidirectional constraint. According to the rules defined in Ref. [32], questions will be asked first about its constraining object that has the most constraints. In this case, they are "business effectiveness," "work efficiency," "applications," and "business process." By applying the same rules, for each one of them, questions are asked as listed in Table 2. Those questions have to be answered to narrow down the scope of an EAI case. When answering a question, it is suggested to give a quantifiable or measurable answer. For example, for a question like "what do you mean by effective?," an answer with a quantifiable or measurable factor is preferred, e.g., "a representative is able to process two orders in 10 min (was one in 10 min)."

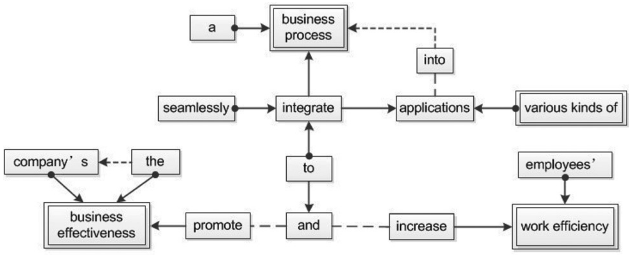


Fig. 4 The ROM diagram for an EAI objective in general

Table 2 A list of general questions asked based on Fig. 4

Main object	Questions to be answered
Business effectiveness	What is the company? What is the company's business effectiveness?/What do you mean by effective? Why/How to promote?
Work efficiency	What are the employees? What is employees' work efficiency?/What do you mean by efficient? Why/How to increase?
Application	What are various kinds of the applications?
Integrate	Why/How to integrate applications to business process? How to integrate to promote? How to integrate to increase? What do you mean by seamlessly?

Once an answer is provided, the ROM diagram is updated accordingly. If every object in the most updated ROM is clear, domain-specific questions will be asked next. Domain-specific questions are related to the lifecycle of the product to be designed. For an EAI problem, the final product is an EAI solution. Several events within the lifecycle of a commercial EAI solution can be summarized as design, manufacturing, sales, deployment, use, maintenance, and disposal. For each event, more requirements are collected according to Fig. 2(*b*). With those requirements, the ROM diagram is updated again, and questions are asked based on the new objects. In this way, environment components and their relations can be sufficiently gathered.

Followed by environment analysis, conflict identification will be conducted. According to the general forms for a conflict defined in Fig. 3, along with the predefined rules in Sec. 2, major conflicts could be identified. Analysis of the dependency between the major conflicts will thus lead to the root conflict(s) which will be resolved first. This will be illustrated in the case study.

Since EAI solutions are mostly case dependent, an EAI solution cannot be expected to fit into all the companies. However, the objective of any EAI is the same, which makes a general framework feasible. Back to the generic questions and the roadmap, several indispensable major environment components are the company, the business process, the employees, maintenance, functions, and human–machine interface. All of those major environment components are indeed connected by the so called workflow in the modern business. The realization of the workflow depends on information/data flow throughout the business process among involved parties/stakeholders (employees, customers, partners, etc.). Ideally, for EAI implementation, a single standard file format, such as the universal file format (UFF) [35] would be perfect, which reserves and conveys the necessary information required by all applications. However, it is impossible in real practice due to security reason or business realities. Vendors worldwide have launched software systems that follow particular standard in each domain. That is why aforementioned brokers or adapters in introduction have been developed to bypass the format limitations or support UFF as an intermediary agent to enable a smooth information/data exchange. In EAI, "*seamlessly*" can be achieved by minimizing the human operations within the workflow and by replacing them with digitalized information/data flow. In order to provide extendability (maintenance consideration) to an EAI solution, APIs are expected for future integration. It involves domain knowledge in administration, management, and information technology in this regard. Thus, a general framework is graphically represented in Fig. 5. Four layers have been adopted for the framework—a data management layer, an API layer, a workflow layer, and a GUI

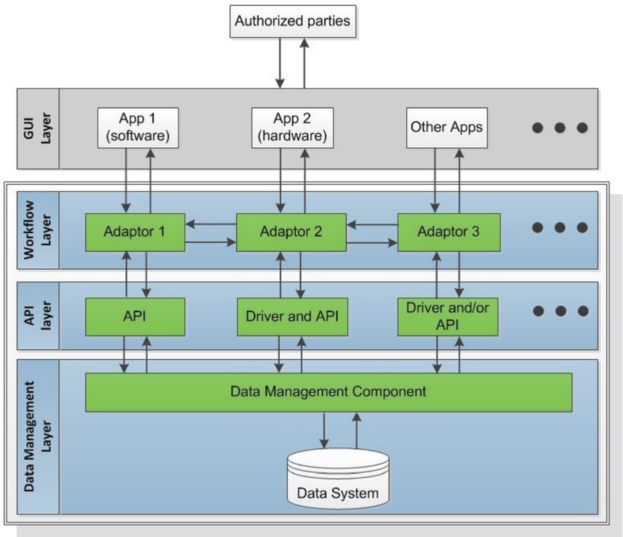


Fig. 5 A framework for the EAI problems

(graphic user interface) layer. The authorized parties can operate any application (hardware or software) to access/send information/data to a single data system regardless of the application type. In this way, an EAI solution is able to seamlessly integrate all applications into the business process.

3.2 A Case Study. This case study was taken from a metal cutting tool developing, designing, and manufacturing company. This company has approximately 100 employees and has more than 40-yr history in the metal cutting industry. Like many other enterprises with long history, this company had a very urgent demand on integrating different applications they have been using. One of the challenges faced by the company is give all their employees necessary, transparent, and real-time access to information needed for achieving their business goals. Many of the legacy applications still in use today were developed by aged and proprietary technologies.

In the past two decades, the company has been constantly investing into enterprise applications to improve its management as well as product design and realization. In the early 1990s, the company was one of the first few in North America to use an early ERP system MAPICS. The system includes inventory management, product data management, production monitoring, customer orders management, accounting management, sales analysis, and other modules [36]. Therefore, almost the entire company is managed by this system. Although it is a very old system without friendly user interface, the ERP system has been adapted and combined with the company's business processes in a very high level. It is acting as the heart of the company. Thus if the system stopped working, the company's normal business would be severely interrupted. Although the MAPICS is very powerful, the developer (IBM) has abandoned the software and developed a new ERP system to replace it. IBM can provide data transfer tools if the company decided to upgrade to the new ERP system; however, it cannot guarantee that everything can be transferred properly. Because of this concern, most of the managers of the company rejected to upgrade the ERP.

In addition, many other enterprise applications, such as AUTOCAD, WORD, and EXCEL were introduced for different purposes. Many of the legacy applications still in use today were developed by aged and proprietary technologies. There is not a data sharing framework for those applications. All data transferring/converting are implemented manually. As a result, the business processes do not run smoothly within the company. There was an increasing demand of either upgrading their enterprise applications to sophisticated ones, which would cost the company dearly, or integrating the existing applications, which is time-consuming. After some investigations, a comparison between two possible options is shown in Table 3. It clearly shows that the best solution for this problem is to develop an EAI solution which can maintain the legacy applications, especially the MAPICS system, and address issues related to these applications.

Table 3 Upgrading versus developing an EAI solution

Problems	Upgrading	EAI
Cost	High (>$380,000 [38])	Low (<$10,000 [36])
Better interface of the ERP	Yes	Yes
Affect the current production	Yes	No
Data transfer	Yes	No
Employee training	Yes	No
Integration of ERP and office	Yes	Yes
Integration of ERP and CAD	No	Yes
Integration of CAD with other physical devices	No	Yes

Obviously, for the concerned company, developing an EAI solution is a better option to make use of the legacy applications and to resolve the problems from the previous analysis. The EAI development can be viewed as a design problem. The objective can be summarized as "design a solution to integrate the current enterprise applications and to smoothen the business processes." From this one-sentence objective, we can start requirements analysis.

3.2.1 Environment Analysis. A ROM diagram can be drawn as in Fig. 6(*a*) according to the objective. It is clear that "*applications*," "*processes*," and "*solution*" are the current key environment components, since they are the most constrained objects in the diagram. Then generic questions about the objects in the ROM diagram have to be asked, according to the question asking rules defined in Ref. [32]. For example, we start with the most constrained object—"*applications*": "what are the current applications?" As mentioned above, there are three major types of applications: MAPICS, AUTOCAD, and MS OFFICE. With the answer, the ROM diagram is updated as shown in Fig. 6(*b*). Then, we can go on to ask a question about the most constrained object in Fig. 6(*b*)—"*processes*": "what are the business processes?" For the company, the business processes can be modeled as shown in Fig. 7. Generally, there are 12 steps for the company to finish an order, which are listed in Table 4.

Although some steps (for example, steps 4 and 5) may not be necessary for an order, most of the steps are conducted by employees. This situation will definitely result in many potential conflicts, which will be identified later. With this answer, the ROM diagram is updated again. We can move to another object—"*solution*," just like what we did for the other two objects. The answer to the "*solution*" is summarized as "cost-effective (budget for $10,000), easy to use (user-friendly interface), reliable (not subject to human error, guarantee data integrity, emergency handling)".

After we finish question asking for the environment components, we should start with the relation objects. If we take "*integrate*," for instance, a question like "how to integrate?" can be asked. To answer such a question, we need specific knowledge, which means investigations have to be made in order to give a good answer. For this case, fully automated application integration type [37] with data structure model are employed to minimize human errors in the business processes. Using this structure, it is not necessary to manage the point-to-point information transferring/converting between two applications. Each application can retrieve data from a sharing database. For the question "how to smoothen?," the company indeed wanted to speed up the whole business cycle and to release the employees from tedious repetitive work, by eliminating unnecessary manual operations. Such a release can eliminate some potential human errors to improve the reliability of the data within the company. As a matter of fact, the answer to "how to integrate?"

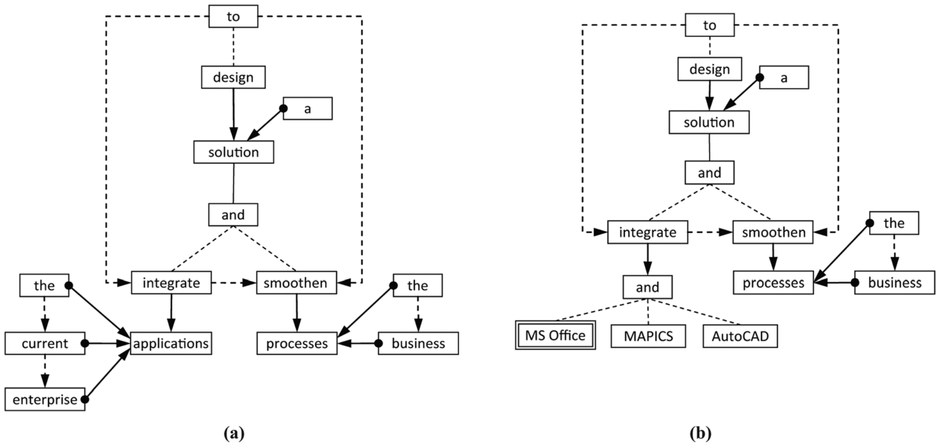


Fig. 6 ROM diagrams. (*a*) The ROM diagram for the objective. (*b*) The updated ROM diagram after a generic question.

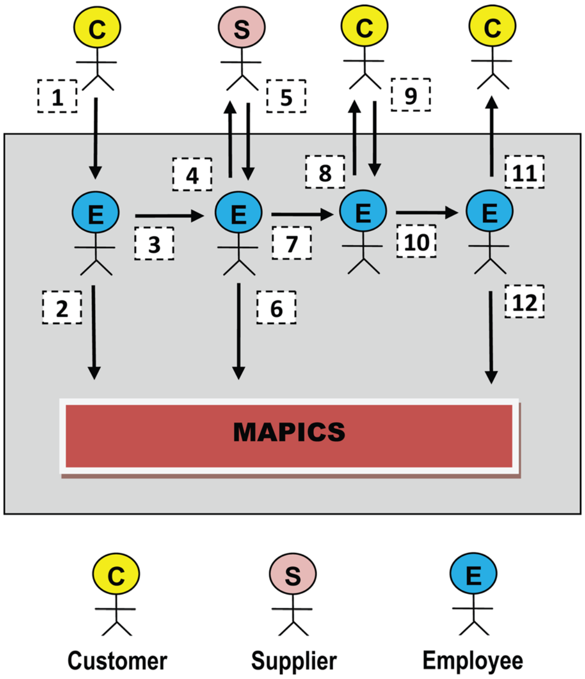


Fig. 7 A model of business processes for the company

Table 4 The general business processes for the company

Step	Description
1	Make an order
2	Create the order to MAPICS
3	Inform the manufacturing department
4	Inform resource suppliers for materials if insufficient in inventory
5	Resource received, manufacture the ordered product
6	Check or update the MAPICS records
7	The ordered product is ready
8	Send bill to the customer
9	Payment received
10	The order is ready to ship
11	Ship the product to the customer
12	Update the MAPICS records, complete the order

Table 5 Features of the current enterprise applications

Applications		Features
MAPICS	Pro	Reliable, powerful, modularized, electronic reports, supports a script language
	Con	Command based, no GUI, manual operation only, too expensive to upgrade and no guarantee of correct data transferring, difficult to cooperate with other applications, documents cannot be read by other applications directly
AUTOCAD	Pro	Preview of drawings
	Con	No preview for odd version drawings, naming chaos of drawings, duplicated drawings, no batch printing function
MS OFFICE	Pro	Widely used for business documents
	Con	N/A

has partially addressed this question. If we put these two answers back to the ROM, we should further proceed our environment analysis by asking questions like “what are the kinds of data?” In order to answer this question, we have to go back to the three enterprise applications to investigate the types of data sharing required. Table 5 shows the features for each application.

The process of asking the right questions, getting the proper answers, and updating the environment components continues, until the designer defines all the relevant components to the design problem.

According to the roadmap mentioned in Sec. 2, since the EAI solution is specialized for the company, its lifecycle is divided into four kinds of events: design, manufacture, use, and maintenance. For each event, the requirements are further classified into natural, human, and built environments. The details are shown in Table 6. Based on the analysis carried out so far, the ROM diagram can be updated and simplified in Fig. 8.

3.2.2 Conflict Identifications and Solution Generation. Based on the updated ROM diagram, we are able to identify many conflicts by following the aforementioned rules given in Sec. 2. For example, in Fig. 8, for the object “*solution*,” it is required to be “*not subject to human errors*”. And it is obvious that the “*solution*” includes two branches (indicated by the object “includes”). According to the defined Rule 3 in Sec. 2, the constraint “*not subject to human errors*” also constrains the object “*smoothen*”. Thus, for “*smoothen*”, we found two constraint relations as shown in Fig. 9. This is a conflict. In addition, we could figure out that “*manual operations*” exist in “*12 steps*”, “*MAPICS (ERP)*” and “*manually converting electronic reports into MS OFFICE (.doc and.xls formats)*”, by following the object “*including*”. Thus, based on the ROM

Table 6 Lifecycle analysis of the design problem

Event	Natural	Built	Human
Design (designer)	N/A	MAPICS (ERP system), difficult to cooperate with other applications, expensive to upgrade, and no guarantee of data transferring correctly, MS OFFICE, AUTOCAD, no batch print function for AUTOCAD	Fully automated, applications integration, smooth business processes
Manufacture (programmer)	N/A	Electronic reports cannot be read by other applications, only AUTOCAD drawing after version 2000 can be previewed, budget for $10,000, MAPICS supports a scrip language, its own database, modularized, no interface, AUTOCAD drawings with naming chaos and duplications	A sharing database, development tools
Use (employee)	N/A	MAPICS, MS OFFICE, AUTOCAD, the company	12-step business process, user-friendly interfaces, not subject to human errors, emergency handling, data integrity guarantee, release employee from tedious and unnecessary manual operations
Maintenance (technician)	N/A	The company	Installation, deployment, add new modules/applications, emergency handling

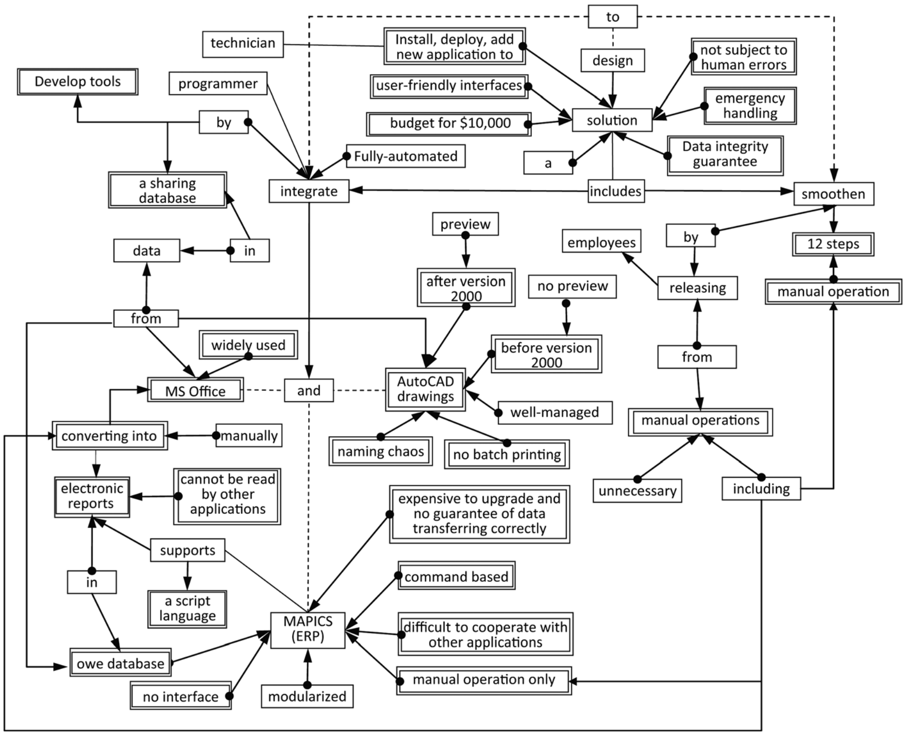


Fig. 8 The most updated ROM diagram after question asking

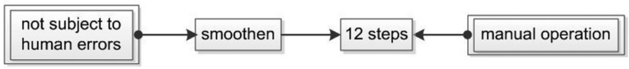


Fig. 9 An example conflict

diagram, we are not only able to identify conflicts but also able to figure out where the conflicts are originated from. In such a manner, proper solutions can be proposed to resolve the conflicts. By following the same pattern and the rules, several significant conflicts have been identified[2] as illustrated in Table 7 from Fig. 8.

According to the EBD, before starting to resolve any conflict, we should analyze them first. The principle is to find out the dependencies among them as one conflict may be resulted from others. Two rules should be followed in resolving conflicts: (1) resolve the conflict from the natural environment first, followed by those from the built environment, with the conflicts from the human environment always being resolved last and (2) resolve the root conflict first. As such, handling a root conflict may eliminate other dependent conflicts. In this case study, any conflict resulted from human operations (conflicts 1, 2, and 3 for instance) are considered as the least significant ones, since a proper solution to conflict 6 may eliminate them. Figure 10 provides the proper answer to handle conflict 6.

By providing those application operators, employees can use their favorite application to get the information they are permitted to access. By refining these automated operations, conflicts 5 and

[2] The conflicts are manually identified in this case study. However, ROM diagrams are able to be generated semiautomatically by a software called ROMA in the Design Lab at Concordia University. Conflict identification is to be implemented in the ROMA.

Table 7 Conflicts identified from the most updated ROM diagram

Number	Conflicts	
1	Eliminate unnecessary human operations	Many human operations exist
2	Not subject to human error	Many human operations exist
3	Fully automated solution	Many human operations exist
4	Preview for drawings after version 2000	Drawings before version 2000 exist
5	MS OFFICE is widely used	Electronic reports cannot be read by other applications
6	A sharing database	Data from three applications
7	User-friendly interfaces	No interface
8	Too many drawings are naming chaos and duplicated	Get right drawings data from the sharing database
9	No batch print function	A desired function
10	Emergency handling and data integrity guarantee	A sharing database

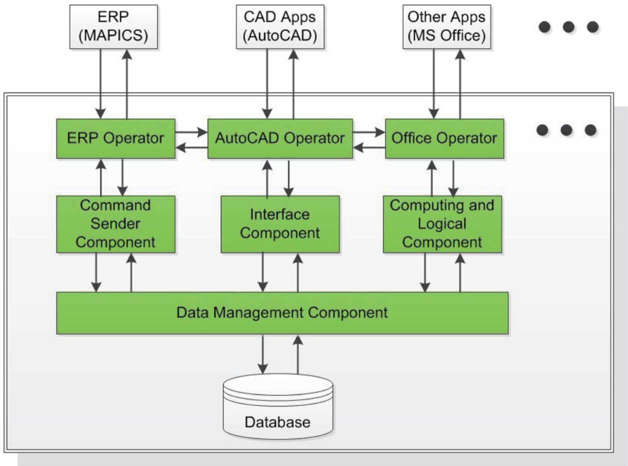
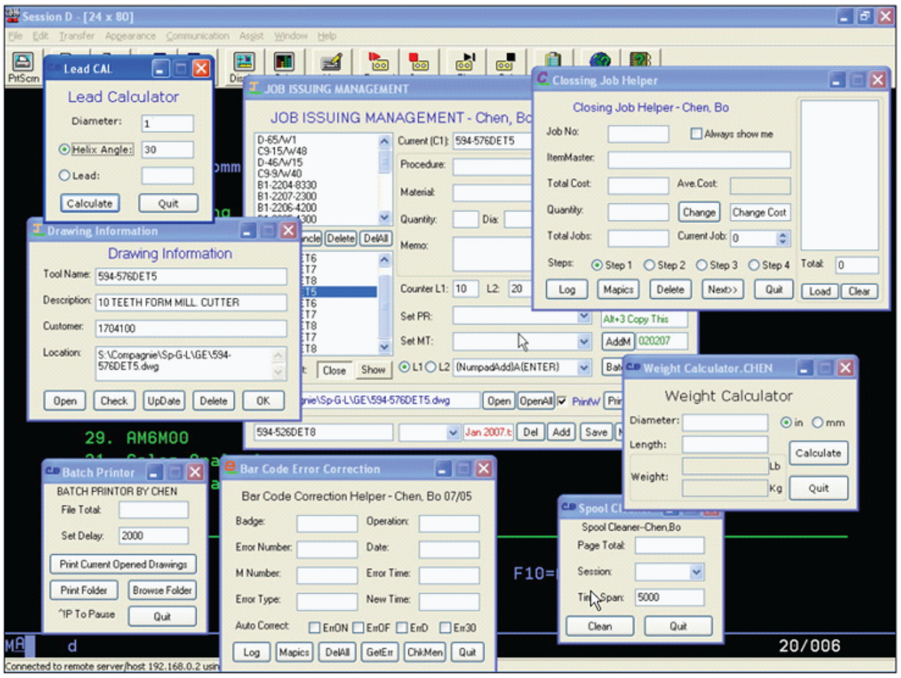


Fig. 10 A database centered structure for applications integration [36]

9 can also be handled. For conflict 7, it is actually a part of the solution for conflict 6. Since the MAPICS supports a scrip language, with some small scale programming to refine the MAPICS' functionalities, the data in the ERP system's database could be retrieved to the shared database. Furthermore, through the ERP operator, the MAPICS can be manipulated, which is not possible before this solution. By using the two databases, one (the MAPICS' database, not directly accessible for other applications) can serve as a backup database, which enhances the robustness of the EAI solution in case of emergency. In order to resolve conflict 4, a segment of code was written using VBA, ACTIVEX, AUTOCAD script, and AUTOHOTKEY [36]. The code can automatically open the earlier version drawings that are without preview function by AUTOCAD 2008 and save them as new versions that can be previewed. For each generated solution, it was put back to the ROM diagram again, and the three activities are recursively applied until no more unacceptable conflict exits. These recursive processes reflect the essence of the EBD—recursive logic [28,29]. After several iterations, the final solution was generated, which was implemented in the company. Figure 11 shows the functional graphic user interface for the final EAI solution.

3.3 Evaluation. In the real world, many enterprises are forced to upgrade their legacy applications/solutions for better business flow. However, it does not necessarily mean that every company has to give away its legacy applications for a major or total upgrade. For this case study, the final solution costed the company less than $10,000 [36]. With the final EAI solution, the life of the legacy applications was extended. The business



Diameter: 1
● Helix Angle: 30
○ Lead:

D-65/W1
C9-15/W48
D-46/W15
C9-9/W40
B1-2204-8330
B1-2207-2300
B1-2206-4200

Current (C1): 594-576DET5
Procedure:
Material:
Quantity: Dia:
Memo:
Counter L1: 10 L2: 20
Set PR:
Set MT:
● L1 ○ L2 (NumpadAdd)A(ENTER)
Alt+3 Copy This
020207
☑ PrintW
594-526DET8
Jan 2007.t

Tool Name: 594-576DET5
Description: 10 TEETH FORM MILL CUTTER
Customer: 1704100
Location: S:\Compagnie\Sp-G-L\GE\594-576DET5.dwg

Job No: ☐ Always show me
ItemMaster:
Total Cost: Ave.Cost:
Quantity:
Total Jobs: Current Job: 0
Steps: ● Step 1 ○ Step 2 ○ Step 3 ○ Step 4 Total: 0

Diameter: ● in ○ mm
Length:
Weight: Lb Kg

File Total:
Set Delay: 2000

Badge: Operation:
Error Number: Date:
M Number: Error Time:
Error Type: New Time:
Auto Correct ☐ ErrON ☐ ErrOF ☐ ErrD ☐ Err30

Page Total:
Session:
TimeSpan: 5000

Connected to remote server/host 192.168.0.2 usin

Fig. 11 The user-friendly interfaces of the final EAI product [36]

processes were improved significantly, and employees were much more content to work with it. Compared to upgrading (see Fig. 3) which would have cost the company at least $380,000 [38] and would not be compatible with some of the legacy applications, the customized EAI solution not only satisfied the company in functionality without any side affects but also provided a surprisingly huge saving—97% of the cost for upgrading. Furthermore, since the design process and source code are available, the EAI solution could be well maintained and extended. Its configurability, extensibility, and reusability indicate more potential savings and benefits for the company. In addition, the case study validated the effectiveness of the EBD approach and the proposed framework.

4 Conclusion

During the last few decades, EAI applications have become a very important issue for many enterprises. Many methods and technologies have been developed for EAI problems. In this paper, the EBD methodology is employed to resolve EAI problems from design point of view. With the support of ROM, it becomes logical and systematic to identify and clarify the requirements for the EAI problems. Based on that, a general framework for EAI problems is proposed. A case study from a small and medium enterprise is also presented. The EAI solution is used to demonstrate how a functional, cost-effective, and efficient solution can be generated to smoothen the entire business processes within the company. Different from other methods, the EBD enables an EAI solution designer to identify the customer requirements and to clearly understand how each given solution affects the whole product in its lifecycle at design stage. Therefore, the imperfect and defective solutions can be identified and eliminated at a very early stage.

Acknowledgment

The research presented in this paper was partially funded by an NSERC Discovery Grant (Grant No. RGPIN 298255). The authors are also grateful to anonymous reviewers for their constructive comments on this paper.

References

[1] Puschmann, T., and Alt, R., 2001, "Enterprise Application Integration-the Case of the Robert Bosch Group," Proceedings of the 34th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, p. 10.

[2] Ma, X., Duval, J., and Rui, C., 2009, "New Developments of CRM in Theory: The Integration Between CRM and Other Business and Technology Orientations," International Technology and Innovation Conference 2009 (ITIC 2009), Xian, China, pp. 1–5.

[3] Scheibler, T., Mietzner, F., and Leymann, F., 2009, "EMod: Platform Independent Modelling, Description and Enactment of Parameterisable EAI Patterns," Enterprise Inf. Syst., **3**(3), pp. 299–317.

[4] Kevork, E. K., and Vrechopoulos, A. P., 2009, "CRM Literature: Conceptual and Functional Insights by Keyword Analysis," J. Mark. Pract.: Appl. Mark. Sci., **27**(1), pp. 48–85.

[5] Botta-Genoulaza, V., and Mille, P.-A., 2005, "A Classification for Better Use of ERP Systems," Comput. Ind., **56**(6), pp. 573–587.

[6] Linthicum, D. S., 2000, *Enterprise Application Integration* (Addison-Wesley information technology series), Addison-Wesley, USA.

[7] Themistocleous, M., 2004, "Justifying the Decisions for EAI Implementations: A Validated Proposition of Influential Factors," J. Enterprise Inf. Manage., **17**(2), pp. 85–104.

[8] Johannesson, P., and Perjons, E., 2001, "Design Principles for Process Modelling in Enterprise Application Integration," Inf. Syst., **26**(3), pp. 165–184.

[9] Irani, Z., Themistocleous, M., and Love, P. E., 2003, "The Impact of Enterprise Application Integration on Information System Lifecycles," Inf. Manage., **41**(2), pp. 177–187.

[10] Du, Y., Peng, W., and Zhou, L., 2008, "Enterprise Application Integration: An Overview," International Symposium on Intelligent Information Technology Application Workshops (IITAW'08), pp. 953–957.

[11] Chen, D., Doumeingts, G., and Vernadat, F., 2008, "Architectures for Enterprise Integration and Interoperability: Past, Present and Future," Comput. Ind., **59**(7), pp. 647–659.

[12] Liu, X., Zang, W. J., Radhakrishnan, R., and Tu, Y. L., 2008, "Manufacturing Perspective of Enterprise Application Integration: The State of the Art," Int. J. Prod. Res., **46**(16), pp. 4567–4596.

[13] Sharif, A. M., Elliman, T., Love, P. E., and Badii, A., 2004, "Integrating the IS With the Enterprise: Key EAI Research Challenges," J. Enterprise Inf. Manage., **17**(2), pp. 164–170.

[14] Lam, W., 2005, "Investigating Success Factors in Enterprise Application Integration: A Case-Driven Analysis," J. Inf. Syst., **14**(2), pp. 175–187.

[15] Vasconcelos, A., da Silva, M. M., Fernandes, A., and Tribolet, J., 2004, "An Information System Architectural Framework for Enterprise Application Integration," Proceedings of the 37th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Washington, DC, USA, pp. 225–233.

[16] Losavio, F., Ortega, D., and Pérez, M., 2002, "Modeling EAI," Proceedings of the XII International Conference of the Chilean Computer Science Society, SCCC'02, IEEE Computer Society, pp. 195–203.

[17] Sandoe, K., and Saharia, A., 1999, *Enterprise Integration*, John Wiley Sons, Inc., New York.

[18] Kamal, M. M., Weerakkodyb, V., and Jones, S., 2009, "Case Study: The Case of EAI in Facilitating e-government Services in a Welsh Authority," Int. J. Inf. Manage., **29**(2), pp. 161–165.

[19] Nia, Q., Lub, W. F., Yarlagaddaa, P. K., and Ming, X., 2006, "A Collaborative Engine for Enterprise Application Integration," Comput. Ind., **57**(7), pp. 640–652.

[20] Gleghorn, R., 2005, "Enterprise Application Integration: A Manager's Perspective," IT Prof., **7**(6), pp. 17–23.

[21] Lublinsky, B., 2001, "Achieving the Ultimate EAI Implementation," eAI J., **3**(2), pp. 3–26.

[22] Umapathy, K., Purao, S., and Barton, R. R., 2008, "Designing Enterprise Integration Solutions: Effectively," Eur. J. Inf. Syst., **17**, pp. 518–527.

[23] Lee, J., Siau, K., and Hong, S., 2003, "Enterprise Integration With ERP and EAI," Commun. ACM, **46**(2), pp. 54–60.

[24] Silveira, R., and Pastor, J. A., 2006, "A Model for Enterprise Application Integration Tools Evaluation," European and Mediterranean Conference on Information Systems (EMCIS), Costa Blanca, Alicante, Spain.

[25] Zeng, Y., 2004, "Environment-Based Formulation of Design Problem," Trans. SDPS: J. Integr. Des. Process Sci., **8**(4), pp. 45–63.

[26] Zeng, Y., and Gu, P., 2001, "An Environment Decomposition Based Approach to Design Concept Generation," Proceedings of 13th International Conference on Engineering Design (ICED), pp. 525–532.

[27] Zeng, Y., 2011, "Environment-Based Design (EBD)," The 2011 ASME International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), Washington, DC, USA, Aug. 28–31, DETC2011-48263.

[28] Zeng, Y., 2002, "Axiomatic Theory of Design Modeling," Trans. SDPS: J. Integr. Des. Process Sci., **6**(3), pp. 1–28.

[29] Zeng, Y., and Cheng, G., 1991, "On the Logic of Design," Des. Stud., **12**(3), pp. 137–141.

[30] Chen, Z., Yao, S., Lin, J., Zeng, Y., and Eberlein, A., 2007, "Formalization of Product Requirements: From Natural Language Descriptions to Formal Specifications," Int. J. Manuf. Res., **2**(3), pp. 362–387.

[31] Zeng, Y., 2008, "Recursive Object Model (ROM)-Modelling of Linguistic Information in Engineering Design," Comput. Ind., **59**(6), pp. 612–625.

[32] Wang, M., and Zeng, Y., 2009, "Asking the Right Questions to Elicit Product Requirements," Int. J. Comput. Integr. Manuf., **22**(4), pp. 283–298.

[33] Chen, Z. Y., and Zeng, Y., 2006, "Classification of Product Requirements Based on Product Environment," Concurr. Eng., **14**(3), pp. 219–230.

[34] Yan, B., and Zeng, Y., 2011, "Design Conflict: Conceptual Structure and Mathematical Representation," J. Integr. Des. Process Sci., **15**(1), pp. 75–89.

[35] Phillips, A. W., and Allemang, R. J., 2011, *Development of a Long Term Viable Dynamic Data Archive Format Modal Analysis Topics*, Vol. 3, T. Proulx, ed., Springer, New York, pp. 541–555.

[36] Chen, B., 2008, "Environment-Based Design (EBD) Approach to Enterprise Application Integration (EAI)," Master's thesis, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, QC, Canada.

[37] Microsoft, 2003, *Guides for Application Integration (Patterns & Practices)*, 1st ed., Microsoft Press, USA.

[38] Aberden Group, 2007, "The Total Cost of ERP Ownership in Mid-Size Companies," http://www.erpsystemguide.com/erp-costs.html