Inf Technol Manag (2012) 13:233–249
DOI 10.1007/s10799-012-0119-8

e-Business systems integration: a systems perspective

Song Wang · Ling Li · Kanliang Wang · James D. Jones

Published online: 17 April 2012

Abstract Systems science has emerged as a meta-discipline and a meta-language, correspondingly, which can be applied to discuss issues in e-business systems and relevant enterprise architecture and enterprise integration. A lot of researches on enterprise architecture and enterprise integration in e-business systems have their theoretical findings and effective practices naturally influenced by systems theory and relative methodologies. This paper strives to review the contribution of systems theory to enterprise architecture and integration. It also tries to summarize methods or tools applied on enterprise systems level, and to investigate many crucial scopes, concepts and their inter-relationship in e-business systems integration activities. Finally, this paper presents new prospects in enterprise architecture and integration for e-business systems. All of these may be useful to deal with the increase complex informatics issues of modern enterprises.

S. Wang (✉)
School of Economics and Management, Harbin Institute of Technology (Weihai Campus), Weihai 264209, Shandong, China
e-mail: benqwang@126.com

L. Li
Department of Information Technology and Decision Science, Old Dominion University, Norfolk, VA 23529, USA

K. Wang
School of Business, Renmin University of China, Beijing 100872, China

J. D. Jones
Computer Science Department, School of Engineering and Computational Sciences, Liberty University, Lynchburg, VA 24502, USA

Keywords Systems science · Systems engineering · Enterprise systems · e-Business systems · Enterprise architecture · Enterprise integration · Industrial Information Integration Engineering (IIIE) · Warfield version of systems science (WSS)

1 Introduction

Systems theory and system thinking have become foundational in natural sciences, social sciences, and engineering disciplines as the second half of the twentieth century witnessed the ushering in of the age of systems science [114, 146, 154]. Meanwhile, during the past 60 years, computers and networks have been increasingly applied to many business applications, and across different industrial sectors. This trend has been accelerated by continuous innovations in information technology [90]. In this development process, as one of the bases of the information systems discipline, systems theory not only has been applied to information systems research, but also contributes greatly to information and communication technology (ICT) as well e-business systems (EBS).

EBS have confronted some crucial challenges from both theoretical and practical viewpoints, such as semantic level e-business messaging, heterogeneous services integration, etc. [52, 84, 126, 160]. A more generic and systematic way thinking about EBS design, planning and deployment should be considered to meet these challenges. Systems theory and its applications in enterprises electronic business systems integration are in great demand [153], thus theorists and practitioners in enterprise architecture (EA), enterprise integration (EI) and e-business engineering should continually change the manners in which they think of EBS. Although many researches in these fields have provided instructive theoretical results and empirical findings, the impacts of systems theory or systemic thinking on EBS has not been well investigated. Thus, this paper will scan the contribution of systems theory to EA and EI, and present some holistic perspectives on EBS.

2 Evolution of EBS

When Börje Langefors first introduced the concept of "Information Systems" (IS) in 1965 [18], IS as a discipline was quickly established. Since then, IS has become the enabling technology that allows companies to capture greater profits in an ever-increasing competitive circumstances. Especially from the 70s to 80s decade in last century, e-business systems based on IS and ICT have emerged and evolved from business-to-business to the integrated and collaborative business services among various IS [84, 162]. Thus, EBS can be defined as the use of IS to work and empower enterprise commerce, enterprise collaboration and to realize web-enabled business processes both within a networked enterprise and with its customers and business partners. So, EBS are distinct from the E-commerce systems that they heavily refer to the cross-function systems integrated to perform the front-end products and services provision [62, 144].

From a historical point of view [17, 142, 152], EBS have spanned almost half a century and crossed the boundaries of traditional business functions [30, 36, 100]. A brief history of EBS is displayed in Table 1 and it can be divided into the following phases: Enterprise Resources Planning (ERP) (and its precursors MRP, MRPII), Customer Relationship Management (CRM) systems, Supply Chain Management (SCM) systems, Enterprise Application Integration (EAI) and Enterprise Collaboration Systems (ECS) [83, 85]. The purpose of these related phases is to control production-planning activities in modern enterprises and connect the marketing, finance, customer management, even human resource management [145] and enterprise strategy seamlessly. Each is a successor of the previous phase [14]. For instance, the same logic used for production-planning in MRP is also used in ERP [44, 81] and the integration logic and patterns penetrate through SCM to EAI [155]. These systems have become the key infrastructures in modern enterprises. They monitor demand, supply, product, inventory, accounting, marketing in an integrated manner. Development and management issues ranging from database integration to crucial architectural issues of other systems are creating new challenges.

The pressures of intense global competition compel enterprises to streamline their operations. They must minimize delays, and reduce costs. Some examples of these costs include inventory costs, production costs and transportation or delivery costs. New forms of organizations have emerged, such as extended enterprises or virtual enterprises [56, 57, 99, 105, 149, 158, 161]. Partners in these kinds of organizations must acquire strong coordination and commitment capabilities to achieve business goals. The information resources of today's business enterprises can be viewed as a network of multiple heterogeneous information sources over which various complex business procedures are executed. These independent information systems are traditionally built to automate existing data-intensive business functions (for example, billing) that are otherwise performed manually in separate organizational entities. By automating these functions separately, an enterprise typically ends up with many stand-alone systems between which related information is distributed and more importantly, not shared. In the competitive global circumstances, it is crucial to eliminate these so-called "Islands" (of information processing and information resources.)

Current research and practice in EBS do not provide business modelers (or system architects) with an adequate formal perspective from which business processes can be analyzed in an open, continuous, overall manner. This gap is serious. Systematic analysis is essential for determining how changes in EA, enterprise process logic and various business parameters affect business performance. Obviously, such efforts can help make better managerial decisions. Moreover, an underlying formalism would help the enterprise system architect generate multiple, all encompassing views of the enterprise at various levels of abstraction. Such a capability is essential for managing the enterprise e-business systems.

Conventional knowledge points to the fact that business results are tied to physical processes, whereby resources are converted to products satisfying market demand. However, contemporary enterprise modelling approaches do not adequately reflect the interrelationship between business and engineering. Engineering approaches generally focus on the physical conversion at one end of the process spectrum, while business approaches focus on the market, finance and enterprise strategies at the other end. To be more effective, EA based on e-business engineering should draw these two aspects closely and emphasize processes plan, design, control, and demand management with managerial view [138].

3 Systems theory and systems thinking in EBS

3.1 Systems theory as a background to support EBS

EBS play the key role in enterprise business data and information processing, thus EBS are not created for their

Table 1 A brief evolution history of EBS integration

Decade	Main content/integration focus	Integration approach
1960s–1980s	Mainframe computers were used, computers and data were centralized systems, then PCs and LANs were developed and installed, and some fundamental business functions, such as inventory, billing, and payroll etc. can be informatized. (For an instance in manufacturing, MRP, MRPII and database administration methods were also developed)	Firstly programming in COBOL (Information Island) and then Supported by PC (Basic Networking)
	The focus was to automate existing processes, also to bring electronic data to the desktop to assist office workers, and to share data	
1990s	Wide Area Networks (WANs) became corporate standards. Designers moved towards system integration and data integration. No more isolated information systems. (ERP,CRM and SCM are eventually developed to meet challenges)	Network supported (Systems Integration)
	The focus was upon integrated control via decentralization and corporate-level management of inventory, manufacturing, finance and marketing. Some intelligent technologies (such as OLAP, Data Mining) were also developed to help enterprise wide information processing and decision	
2000s–2010s	WANs expand via the Internet to include global enterprises and business partners. A major emphasis was upon data sharing across systems. (ERPII, EAI, Enterprise Collaboration Systems such as Enterprise Mashup, Cloud Computing)	Network supported, IS architecture to support operation and decision (Systems Integration)
	The focus was efficiency and speed in inventory, manufacturing and customer response. There was also an obvious shift from qualified production to satisfying services and enterprise strategic decision making	

own sake. When one system is thought of as serving another, it is a principle of systems thinking that this relationship should be thought about very carefully. It is important to carefully define the nature of the system served, and how that service is being performed [23, 24]. How we envision the system being served will determine what is necessary to support it. It is obvious that one information system to facilitate accounting processes will be quite different from the information system to support manufacturing operations. As for the EBS, the dynamic processes modelling and coordination and their underlying logic of enterprise architecture should be reconsidered carefully, especially in the ever changing business environment, how to respond customers and partners requirement is the big challenge. All of this ask for the systematic thinking on the EBS development and EBS integration.

Beer's viable system design (VSD), or equivalently, the viable system model (VSM), is based on the belief that a system is viable if it is capable of responding to environmental changes by achieving the necessary variety to survive [11–13]. VSD applies cybernetics thinking to organization management with regarding an enterprise as a recursive system. VSM consists of five subsystems (Fig. 1), they are:

- System 1 Implementation—to execute primary activities by organization units that actually provide products or services;
- System 2 Coordination—to perform regulation and tactical planning that ensure System 3 to monitor and audit the activities within System 1;

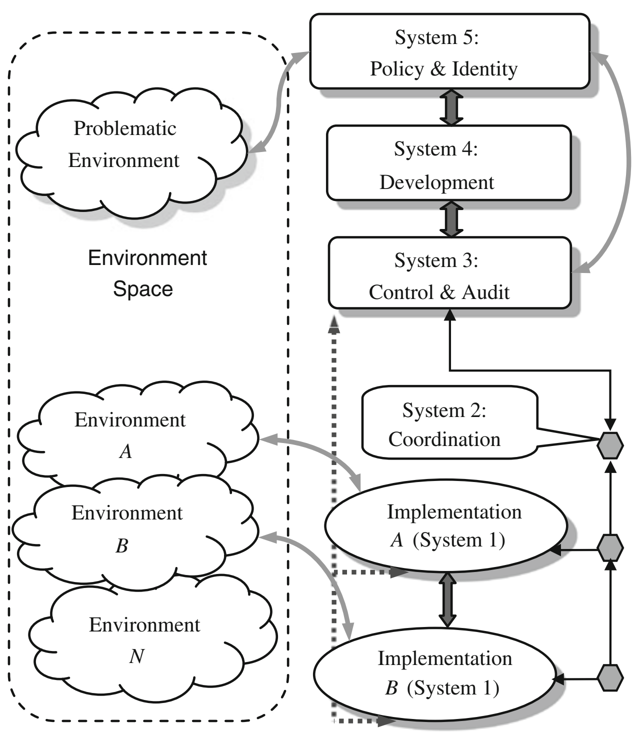

Fig. 1 The schema of variable system model

- System 3 Control—to provide audit and operational control of System 1 and interface with Systems 4 or 5;
- System 4 Development—to provide foresight and monitor external environment to make how the organization adapts to remain viable;

- System 5 Policy—to be responsible for policy decisions and to maintain identity.

When operational and strategic activities are mapped into information processes, information flows and organizational structure play a very important role in VSD. System 5 releases strategy decision information, then enterprise architect function is mainly accomplished by System 4 with focus on strategic planning and roadmaps of strategic initiatives. Thus, Enterprise Architects (System 4) with their concerns of external environment must cooperate with the Solution Architects (System 3) to perform internal and just-in-time auditing and control in Operational Architects (System 1 and System 2). Therefore, Beer's analysis on organization information structure and channels offers insight of allocating resource to managers and practitioners. VSM has become a hierarchical and comprehensible framework to design, construct, and measure EBS especially when we confront unstructured problems of enterprise systems.

Systems within which people involved are ever changing because of the uncertainty arose from human factors, thus soft systems methodology (SSM) has been created [1, 23–26]. This methodology emphasizes that any system is constantly changing, being re-created by its human actors, so systems analysts need to apply their skills to problematic situation (complexity of real world) which are not well defined, and try to understand the complicated organizational settings. This can be achieved via the learning paradigm of seven stages (Fig. 2):

(1) Enter into a situation considered problematical (i.e., unstructured);
(2) Express the problem situation;
(3) Formulate root definitions of the relevant systems;
(4) Build conceptual models of these systems;
(5) Compare these models with real-world actions;
(6) Define possible changes that are both desirable and feasible; and
(7) Take action to improve the problem situation.

Therefore in EBS development, SSM advocates that effective learning (stage (5), (6) even (7)) should take place as an interaction between reality and system thinking about the real world. The problematic situation of IS development can be perceived and depicted in stage (1) and (2), thus various systems or solutions are defined and modelled via conceptualization (stage (3), (4)), and then these models are in turn confronted with the real situation (stage (5)). So the desirable and viable changes can be found to take actions to improve design and development. Based on this iterative circle, the conceptual world (plan and design activities) and real world (situation analysis and enforcement of EA) are organically bridged, that the objective

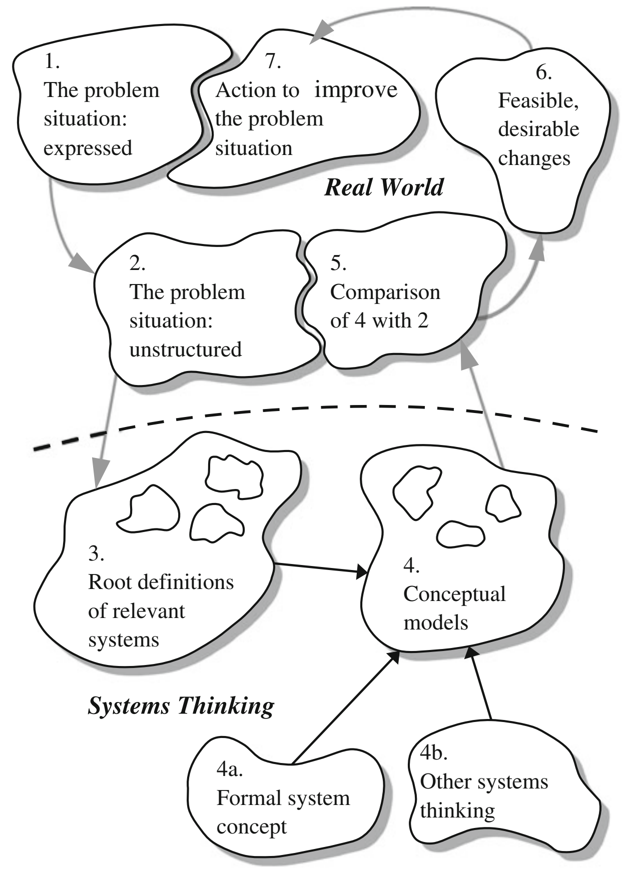

Fig. 2 The Checkland's methodology [23]

systems are eventually shaped and applied to structure and orchestrate debate among actors. For an instance, based on this thinking, Engelsman et al. [43] proposed some kind of enterprise architecture language (ArchiMate language) to shape EBS in terms of stakeholder concerns and the high-level goals from requirements.

Besides, from knowledge management viewpoint, any "enterprise information system" will be some kind of "knowledge-attributed system". In such a system, people select and process certain data, make meaning of them within the enterprise context to create valuable business knowledge. One crucial factor in development and implementation of a successful EBS is to govern the actions of people via certain system architecture. Kamogawa and Okada [71] pointed out that enterprise architecture and its effectiveness should be defined by the following perspectives, especially in an e-business context:

(1) Creating a Governance model is a central success factor. This is because both intra-enterprises and inter-enterprises should maintain the integrity of their business processes, and the interactions of these processes.

(2) It is necessary to develop flexible business information architecture. This is because e-business requires high performance, and processes high volumes of data.
(3) Senior management must understand the architecture. This is important, because the activities for developing an EA are conducted between enterprises, as well as in-house. Communication is often difficult. Knowledgeable and cooperative management are crucial to overcoming these difficulties.

Therefore, SSM exactly provides a referential norm when people are participating in EBS integration. It can equip IS developers (even users) with the structured and socialized analysis means to design and modify EA or software when dealing with unstructured situations.

3.2 Approach shift in EBS development

The traditional systems development approach (a bottom-up, structural design and analysis) historically has had a great impact on the construction methodologies of Information Systems Development (ISD). Such a view may seem old-fashioned, for it holds that ISD should be accomplished in a rigid manner trying to finish various tasks in a natural and logical order. ISD work is therefore partitioned into a set of manageable phases and steps to build and shape new IT systems. Each phase requires different types of specialists or experts knowledge. The conventional systems approach has been regarded as a core theory in the ISD field.

With never-ending update in software and hardware, perhaps fuelled by fairly complex business circumstances (such as the formation of global supply chains, trans-organization cooperation, and the emergence of virtual enterprises), the distributed and swift approach has became the paradigm of enterprise ISD. This is testified to and verified by industrial utilization in business or public administration [89, 159]. Great efforts have been expended to extend the conventional ISD work to more open and agile methodologies. Although with some varieties evolved, the Waterfall Model as a methodology basically depends on the successful delivery from earlier stage to next in a scheduled sequential manner, while modern methodology has increased emphasis on the dynamic and iterative manner. That is, at initial stage an original version (prototype) should be coined rapidly with essential architecture shaped to cope with possible requirements and risk in the future. Then system development is preformed in an iterative way to approach the prefect architecture and expected functions. This dynamic methodology focus on continuous improvement to integrate crucial requirements into system architecture, and finally to make information systems steady enough to evade failure. However, this new methodology also brings about more managerial challenges thus developers and planners should rethink development as a system evolution process.

Additionally, ISD approaches which had merely considered issues of information system itself are now evolving to focus on issues arose from a larger social system [97]. Regarding information systems and the relevant business processes as a complex, multidimensional "super-system" may be helpful. For example, human-centric processes (such as Customer Engagement and Customer Relationship) or interactions outside of the boundary of the enterprise (such as Supply Chain) will heavily influence enterprise ISD. Taking such a broad view from multiple perspectives is a generic approach within which systemic thinking should be done. Such a "multi-perspectives" approach can be seen as a new extension to conventional systems development.

4 Systems perspectives on EA

EA planning as an enterprise-level system design methodology, is consciously or unconsciously applied in and between many fields, thus some practical experience and lessons have been learned [116, 155, 156]. Yet, EA is not a popularly understood term [121], without mentioning its substance of a system planning. The term "enterprise" refers to the scope and territory of the organization, dealing with all organization resources as a whole. It can also refer to multiple agencies, rather than being limited to just real organizational part or component and/or project [140], which means any agency should be programmed and deployed according to a unite framework. The term "architecture" is based on descriptions of how an organization can exploit IT to optimize service processes, enhance control and decision capability, save operation costs, and finally to ensure effective IT investment and realize strategic vision. Architecture aims at creating some kind of structure in a chaotic environment using systematic approaches [4]. It can also promote responses, as to what and how information has to be made available. Thus, the strategic aspects of IT systems provide the contexts for the architectural design choices. As for the operational level of enterprise, concentrating on the real business logic, EA is the organizing logic for business processes and for the IT infrastructure, thus it reflects the integration and the standardization requirements of the firm's operating model [148].

4.1 Synthesis

System theories focus problems on the relationship between the parts and the whole. For an organization, business units or service providers are the parts and the organization itself is the whole [75, 80, 119, 157]. For a supply chain or even virtual enterprise, all members or coalition units embedded within are the parts, while the supply chain or entire virtual enterprise is a whole [133]. The enterprise as a system can accomplish certain services and bring final vaules to customers [68], while the parts of it cannot. This property is called "whole affectivity" (or "synthesis") in enterprise systems. From systems theory standpoint, synthesis in EBS technically has two fundamental facets:

- Synthesis inside is the enterprise scope itself. The scale of the synthesis covers relationships between each department, the restrictions between each resource, and so on. All of these need tools to cooperate and unite, while EA can provide unifying data and handling principles to bridge isolated "silos" of each IT subsystems.
- Overall enterprise synthesis (internal and external integration) in a supply chain or in a global marketplace causes synthesis among stakeholders with different values, thus EA should take into account the accessible interface mechanism, interconnection conflict management to balance requirements and supplies, coordinate centralization and decentralization.

For example, encircled within dynamic multidimensional data, manufacturers use Computer-Integrated Manufacturing Systems (CIMs) [20, 63, 102, 134, 142, 170] to facilitate their operations and decisions. Thus, CIMs typically entail multiple information systems that, individually need to meet their own functional requirements while collectively, must work together and form an integrated environment for the enterprise as a whole.

Additionally, EA identifies the scope of individual systems, and the boundaries between them. Therefore, EA is essentially a planning activity rather than development activity. The distinctions between these two activities are unwittingly ignored in practice. Without having such an understanding, organizations usually focus on the improper sets of issues when developing EA. As a result, they get little value from the architecture. In practice two basic problems in synthesis often occur:

- Having too large of a scope for the EA. That is, the architecture is too grandiose. This results in an open-ended effort that is too ambitious to be successfully implemented.
- Having the architecture burdened down with too low of a level of detail.

In such situations, the problem comes that focusing on the functionality of individual systems is rather than on the interconnection and interaction among them, namely maximizing performance of the parts, at the expense of performance of the whole. The outcome is that the performance of the system as a whole will suffer inevitably.

Although the IT architecture means the formulation of the entire information system structure (including communications between components, function assignment and physical distribution even the system flexibility achieved by all components), the Enterprise Architecture is not only the description of these software/hardware configuration but of the governing, processing even the quality architecture [101]. Finally EA should provide an overall managerial solution to meet sustainable IT development challenge. Further, the planning of an EA must delicately consider the balance between the costs of EA deployment and the efficiency that could be provided. Stated differently, there is the huge risk that cost control could have the unexpected effect of reducing the efficiency of resulting system. Therefore, such synthesis is not merely the technological problem but more complicated strategy problem.

4.2 Abstraction and mapping

Traceability is a challenge throughout the life cycle of EA development. The challenge of mapping the objectives (which we define as business needs, risks, system issues, opportunities for change, and other nonfunctional requirements) to specific architectural elements is unmanageable. We need a simpler way to ensure that the architecture meets its objectives [8, 39, 65, 109, 137]. Here, we utilize SSM and define the operation of enterprise as the "Management Domain", then define the operation of IT/IS as the "IS Domain". "Domain—Mapping" can be applied to guide the EA activities, including analysis, design, and evaluation.

The Management Domain consists of activities or particular processes that encompass the enterprise operation. These activities include various functionalities of the organization, management processes, message communication mechanisms, decision making procedures, monitoring-feedback mechanisms, etc. The Management Domain is the platform on which logistics, capital, and information circulate.

The IS Domain consists of the hardware infrastructure, the software architecture, and functional models. These models are the conceptual views and the logical processes of the enterprise, and are based on information technology. The mapping between these two domains (Management and IS) can be realized by abstraction of the Management Domain directing the IS domain. This systematic approach ensures that the EA is analyzed and executed in a comprehensive manner, with clear boundaries.

There is little research to explicitly present the "Domain-Mapping" methodology. Modeling in EA under this approach has been discussed in some of the following work [37, 75, 80, 96, 109, 143]. This domain-mapping can be seen as the quasi functional activities in EA development, with different levels of abstraction determined by high/low degree of mapping form Management Domain to IS Domain, and it can further specify objectives of each level and finally provide clear definition and descriptive logic of the whole enterprise systems [39, 75, 110, 137]. For instance, at the real operation level of an enterprise, the operation processes logic is abstracted into business domain, but to verify and control these processes we should define the management domain to create the attributes of entity (the objects processed), the relationship among these entities, and the processing principles. But finally how to define and realize these businesses into reality, we should find the mapping approach from business domain to IS domain via management domain. Thus, the mapping methodology becomes primary in EA planning and should be emphasized especially when dealing with individualized EBS development.

4.3 Hierarchy and granulation

The structural hierarchy (of the systems, not of the organization) is one of most import issues in EA. Designers and information systems developers should not only be aware of the existence of subsystems and their interactions, but also be able to identify their types and hierarchical functionalities. This is to confirm the level of abstraction at which analyses can be carried out. It is difficult for developers and users to have a mastery of the knowledge of systems [27, 78, 125, 131, 151]. For example, when value chains extend beyond an enterprise, supplier and customer systems become their own information architectures. Data are distributed over a multitude of heterogeneous systems, and the communication among them can not be easily tackled. A traditional approach to settle this problem is to divide the enterprise and its stakeholders into a layered framework. This framework is called the Layer Model (LM), and includes the following layers: conceptual layer, business layer, applications layer, technology layer, etc. The rapid transformation of organization structures also requires an integration of the layers to support the business processes effectively [58].

EA provides ways to deal with this complexity. These include work (who, where), function (how), information (what) and infrastructure (how to) [67]. Under this new hierarchical perspective, EA can accommodate the inter-organizational processes. A specific example of this is the integration of independent ERP systems via some message passing methods. This achieves higher performance without having to discard legacy systems. This Layer Model provides a basic but distinct modeling mindset, which is helpful for developing EBS and for evaluating their performance [7, 24, 33, 69, 110, 147, 165].

Based on systems viewpoint, granulation can be seemed as the extraction of notable attributions of original system thus to form new functions level and endow it with new constraints. This is accomplished via screening features of lower levels and formulating corresponding relations between original system and newly derived system. Granulation may result in new hierarchy when considering different levels have their own constraints and behaviors. Surely, amongst possible granulized levels only those relevant to the whole enterprise can be adopted into the final architecture. The granulation approach has already become an implicit designing method or guideline for analysis [164, 165]. Granulation plays a fundamental role in the creation of architectural principles. It also establishes constraints imposed upon the organization, and/or the decisions made to support business strategies. With respect to EA, it is used to guide design decisions, and to limit the solution space by setting constraints. In practice, when presentation level (user interface, UI or client input/output applications) invokes business level (technically speaking, such as data processing, communication protocol) to responde and accomplish some requests, if various UI technologies (asp.net, mobile terminals) are applied that will make presentation level become messy, the granulation can be aptly carried out to reorganize a new service level to decouple presentation and business level and facilitate the communication between these two levels.

4.4 Framework and modelling

Being treated as an enterprise informatization axiomatics (sets of definitions, directions and principles), EA needs a framework to specify how IT are applied to overall business processes. The framework describes relationships among technical, organizational, and institutional components of the enterprise [47, 120, 163]. There are many EA frameworks proposed, but most influential ones are as follows:

- The Zachman IS Architecture [164] initiated framework construction in EA; it then evolved into Zachman EA [165] and finally became Zachman Framework [166]. Perhaps it is the best-known EA framework. The presentation of the widely used eXpresApp Framework (XAF) as a matrix reflects the influence of this framework;
- The NIST framework [48] is more comprehensive EA methodology that places great emphasis on interoperability between systems. The subsequent Federal Enterprise Architecture Framework (FEAF) has been developed mainly based on NIST;

- The Technical Architecture Framework for Information Management (TAFIM) and its successor —The Open Group Architecture Framework [40, 135], and TOGAF has became the popularly applied framework in commerce and industries.

Zachman Framework views EA holistically from 5W dimensions (what, how, when, where, why) on which architecture can be planned to process data, function, people, and time. It is more a planning device than a developing framework with less description on detailed architecture design. TOGAF is comparably more practical and agile that it can provide an architecture plan in detail and manage the architecture processes dynamically. Compared to these two frameworks, FEAF is both the methodology and the outcome of architecture design; in addition, FEAF has also provided the Maturity Model and Govern Model of EA which help to make architecture development systematic. In brief, a framework should provide methodological support to modelling in EA. Based on these mainstream frameworks, some systemic view-modelling are proposed to frame EA conceptualization and implementation [103]. From view-modelling perspective, any EBS metamodel have three complementary view layers:

- The Functional View is applied to describe functional settings and restriction (e.g. requirements from various users should be satisfied by applications delivered timely). This view incorporates the architectural model that has been adopted, referring to it as "system architecture". Such as client-server models are created and sketched based on functional view and in E-commerce the cloud computing models are treated in the same way.
- The Topology View refers to the definition of sites composition and processing logic among different sites. The term "site" here means an entity (such as resource allocation, key processing logic nodes etc.) and any site has its sub-site should be properly organized in conceptualization.
- The Physical View provides an infrastructure blueprint to the EBS, such as servers networking with each node composed of hardware and software interfaces.

Any resources (mainly means heterogeneous data must be processed in EBS) can be mapped from functional view into sites that should be allocated. Then Topology and Physical views can be decomposed to the same hierarchical levels of details to respond services and applications and some relevant presentation methods can accordingly be developed (like UML).

4.5 Adaptivity of complex systems

Complex Adaptive Systems (CAS) is usually defined as being composed of a slew of interwoven agents; the interaction of these agents will result in some emergent system-level phenomena. CAS was coined about 20 years ago by researchers at the Santa Fe Institute, for example Murray Gell-Mann, John Holland and others [59, 112]. The core idea of CAS rests on the view of adaptive agents; these agents communicate with their environment dynamically and attempt to understand how their individual behavior affects the system-level responses [6, 118]. Therefore, CAS can also refer to systems that can be simulated and described by Multi-Agent Systems (MAS). Most of the work in CAS has been conducted in highly abstract and artificial systems [1]. Furthermore, Desai [41] proposed the "adaptive complex enterprise (ACE)" derived from Holland's CAS [60]. ACE depicts that many human-designed systems and processes are not complex themselves and focus should be placed on the complexity of non-designed processes. This emphasis on nontraditional processes fairly reflects the real world, analysts should consider interactions between all systems and processes, whether designed or not. CAS assumes the following prime features: co-evolution, requisite variety, connectivity, adaptability and flexibility [67].

As for the adaptability of EA, it means that EA itself has the strategic ability to cope with unexpected changes in enterprise system. In increasing order of complexity, these changes include variations in the physical level (hardware, networks), platform (operating system or DBMS), processes, and applications [106, 123]. Adaptability can be measured by determining the effort that is required to modify the EBS design and structure to cope with these changes. The less the effort that is required, the more adaptable the EA is [5, 9, 15, 31, 53, 73].

Another request on EA is flexibility, which means EBS derived from EA should satisfy the requirements from every level of enterprise with customized solutions, and on the other hand, responding to every customer service in a level, EA can organize and provide some equally effective solutions via EBS. Technically this can be supported by flexible infrastructures based on autonomous or smart technologies [46, 49, 64, 136], but systematic planning in EA conceptualization is necessary. Although flexibility is usually thought of as beneficial, its costs are not well identified [42]. If organizations want to take advantage of CAS, the EA has to be structured to provide proper level of control (not too little, nor too much). The satisfying situation is that an EA can allow flexibility such that systems are not frozen because they are too tightly constrained, nor can they disintegrate in an uncertain circumstance. With respect to the realization of EA, EI plays primary role in bridging heterogeneous entities via asynchronous messaging and distributed computing. All of those systematic requirements on EA should be considered similarly in EI.

5 EI and the application of systems theory

Thacker [134] defines integration as "the information required by each activity available on a timely basis, accurately, in the format required, and without asking". Some experts assume fairly broader concept of EI as the task of improving the performance of large complex processes [109, 111], and the aim of EI is to provide timely and accurate exchange of consistent information between business functions seamlessly to support tactical or strategic goals [130]. Thus, EI should ensure information processing consistency and processing following the guideline of EA.

5.1 Integration modes of EI

Based on systems thinking, EBS can be viewed as the aggregate of multiple components (such as applications, services, platforms, etc.), the mode via which these components communicate is a proactive activity, and should be determined during system analysis and design. Several researchers have identified three basic integration modes, which are concluded and presented in Fig. 3 [3, 10, 19, 34, 39, 54, 77, 119, 132, 141, 150, 155, 156, 171]. These three modes are: (a) peer-to-peer integration, in which communication and interfaces are directly between the individual IT applications, (b) broker-based integration, in which the broker acts as an integration hub enabling real-time processing with middleware between the IT applications, and (c) business-process integration, which extends broker-based integration with knowledge of the business processes, thus the business process model captures the workflow between IT applications and humans.

Obviously, each kind of integration mode has its own merits and limitations that enterprises should choose from among them. The peer-to-peer mode facilitates the direct communication between heterogeneous entities and enhances the timely response with the cost of inflexibility. Although this mode demands frequent interface updating, it also keeps the individual IS entity (as indicated in Fig. 3, the SCM and ERP) autonomous. As the opposite, the process broker mode can provide complex communication and agile coordination among entities while it also increases the system dysfunction risk caused by some key embedded processes failures.

5.2 Integration fashions and platforms

Subjected to various business settings, EI can be approached in different manners [40, 64, 65, 102, 122, 139]. CEN TC310 WG1 has recognized three primary levels of integration:

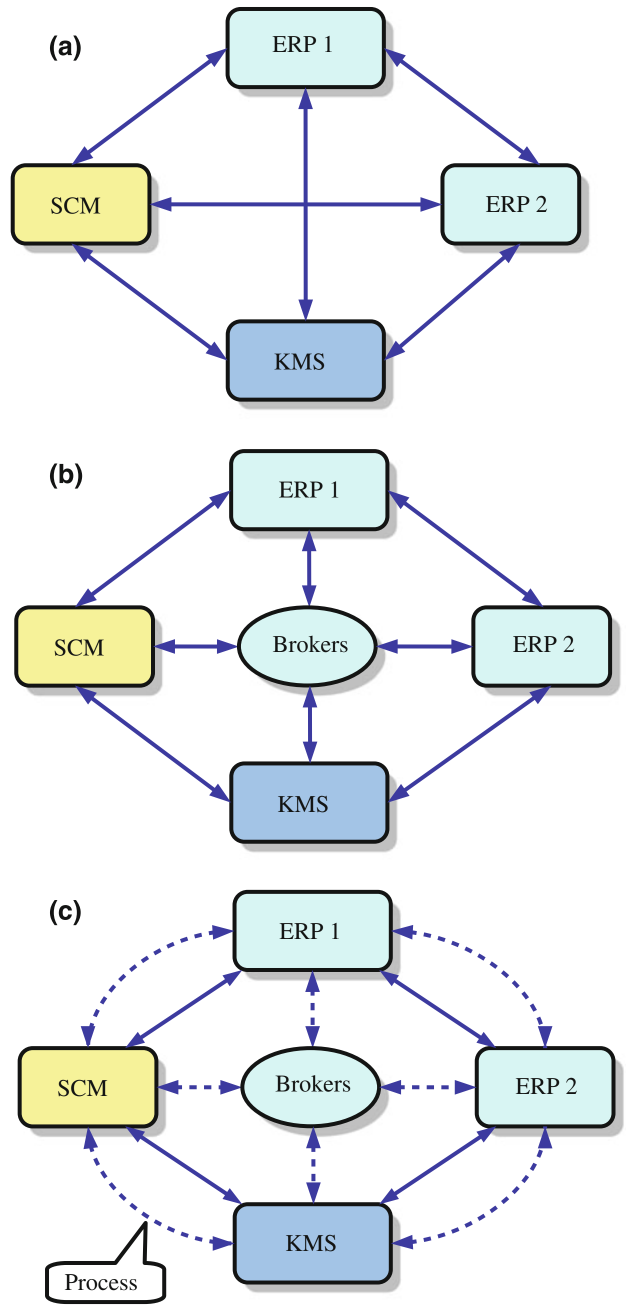

Fig. 3 Three basic integration modes. **a** Peer-to-peer integration. **b** Broker-based Integration. **c** Business-process Integration. *KMS* knowledge management system

- Physical level integration (interconnection of devices, NC machines, etc., via computer networks);
- Application level integration (dealing with interoperability of software applications and database systems in heterogeneous computing environments); and
- Business level integration (coordination of functions that manage, control and monitor business processes).

With an emphasis on the objects to be integrated, Chen and Vernadat [28] present integration modelling in terms of

(1) data (data modelling), (2) organization (modelling of systems and processes), and (3) communication (modelling of computer networks, for example using the 7-layer OSI model). Integration can be achieved by unification (of methods, architectures, constructs, or reusable partial models) or by federation (of interfaces, reference models, or ontology). In addition, more flexible networked cooperation among enterprises and their business units has already promoted trans-organization integration [58], thus traditional vertical integration is giving way to horizontal integration. Lee et al. [79] proposed that the biggest challenge in EI may be the behavioural integration. Processes reorganization and business transformation may be very difficult and sensitive issues in an enterprise as human impacts are critical to the success of EI. As such, both technical and behavioural integration should be taken into account equally.

Unlike traditional integration fashion, enterprise applications integration (EAI) has been introduced in the mid-1990s and it bridges between different applications with specific middleware to realize horizontal and flexible integration. EAI can effectively moderate the conflict between legacy systems and business expansion, and it can provide expedient solutions to coordinate the technical and behavioural integration in enterprise. EAI may be the preferred integration fashion when various difficulties in integration should be dealt with. In perspective of systems development, a layered category of integration platforms is briefly summarized in Table 2.

5.3 Diagram and mechanism of EI

5.3.1 Synergy and interoperability

The EBS integration is widely applied to resolve the problem of "Information Islands". Based on the idea of synthesis, the aggregation of EBS provides functionality beyond what the subsystems can individually provide. Furthermore, dynamic requirements on an organization evolve constantly in rapid changing environments, which means EBS integration should meet not only current requirements, but also future challenges in advance. A rational systematic approach for analysts and developers is to build an enterprise integration platform (EIP). The synergy needed to build an EIP is to aggregate information from different resources, keeping a seamless linkage between varied processes and applications in lower levels, while facilitating complex services at higher levels. The final appearance of EIP is in the modules. For example, the aggregation which results in synergistic e-commerce, builds a platform for B2B, B2C, Logistics, CRM, and other related systems [32, 76, 82, 86–88, 95, 124, 128, 168]. This enhances the transactions between the corporations along the supply chain by releasing and storing all the internal and external information, so that the corporation's value can be added during this process. All of these call for an EBS to ensure interoperability penetrating almost all the computing actions for all business partners.

Interoperability can be defined more broadly as the ability of information systems, and the business processes they support, to exchange data and enable sharing of information [37, 53, 93, 108, 150]. These systems have programs (software or applications) at different levels (vertical integration) and different functions (horizontal integration) [97]. IDABC (Interchange of Data between Administrations, Businesses and Citizens, a community programme managed by the European Commission's Directorate General for Informatics) advances the European Interoperability Framework [113], which suggests that the solution to interoperability problems should follow the same standardized framework for organizational, semantic, and technical interoperability. The IEC standard [74] also approves that the interoperability as a concept in software engineering is a level of compatibility. According to the standard, interoperability happens when software interaction can exist in at least one of three levels: data, functionality, and process [2, 66].

From stratified system viewpoint, technical interoperability plays an elementary role in integration. It requires technical compatibility among protocols or interfaces of the primary specifications for data and applications. Semantic interoperability is built on technical interoperability and it can

Table 2 The layered integration fashions and platforms

Integration objectives	Integration mechanisms/platform architecture	Strategy level	Scope
Cross organization	Collaborative software, strategy information system	Strategic layer	Inter-Enterprises
Inter-organizational decision	Collaborative software, DSS KM system, SOA		
Processes (internal & external to enterprise)	OLAP, workflow, enterprise reference architectures, CORBA, COM, SOA, SCM, CRM, web service	Tactical layer	Intra-Enterprises
Applications (Services)	Inter-processes communication, remote procedure calls, data warehouse, OLAP, CORBA, SOA, ERP, web services	Operational layer	
Data	Data dictionaries, database, XML		

only be achieved when elementary functions and their meanings are shared without ambiguity among stakeholders. Semantic interoperability deals with the "what" of integration, and it is different from technical interoperability, which focuses on the syntax of "how" to do integration. Technical and semantic interoperability can easily be realized if partners operate similar functions. Organizational interoperability is fairly different from both of them with its focus on processes integration. It concerns the information collaboration and governing across organizations despite their internal IS architectures. Organizational interoperability aims at effective processes modelling and qualified service delivery to end users across organizations. Organizational interoperability makes higher demands on business processes reengineering and human tasks cooperation, thus it can not be ignored and should be taken into full consideration when an enterprise integration plan is programmed.

5.3.2 Heterogeneity and connectivity

In its simplest form, integrating information systems means bridging communications among these systems. It is better to view EIP as an infrastructure to support inter-applications communication and generic shared services. EIP provides a unified and consistent view of these data entities in their operations, and it provides information connectivity across multiple platforms [94, 129]. The analysis on how the applications interact may be conducive to selecting suitable platforms; there are three types of applications that can be integrated:

(1) Homogeneous with one instance: One process is supported by one application and one data base. This model avoids the problems that emerge from redundant data storage and from asynchronous data exchange between different applications.
(2) Homogeneous with several instances: Several identical processes, each located in a different business unit, are supported by several identical applications. These applications run on different computers and rely on logically separate data bases.
(3) Heterogeneous: Several different processes in different business units are supported by several different applications. This presents an additional problem compared to the integration in one of the homogeneous environments. The concerned applications are built upon divergent data models, which mean that they provide different semantics of the data to be exchanged.

Integration of EBS constitutes a greater concern in many industrial enterprises encountering complex services requirement. Such enterprises must deal with heterogeneity, which means that there are multiple, dissimilar software applications. A feasible way based on systematic thinking is back to ontology, that means decompose these applications to their explicit conceptualization description, and research their interactive relationships. Izza [66] focused on some semantics-based approaches that promote the use of ontologies. In particular, these approaches use the OWL-S service ontology. The results show that the service-oriented approach can provide a very flexible way to facilitate integration with respect to dynamism. Ontologies may evolve into an interesting technology which can deal with meaning variance and semantic heterogeneities.

As for connectivity, from the perspective of reusability in software applications, middleware as a "glue" programming approach was developed to serve to connect or mediate between two separate and already existing programs [16, 22, 50]. Organizations are now developing enterprise-wide IS with intent to keep previously used applications available. A legacy application can be accessed via its specific interface. Furthermore, the cost of rewriting a legacy application is often not worth the effort. On the other hand, there are an increasing number of systems composed of heterogeneous devices interconnected in a network. Any device embedded in network performs its function via the local and remote message, so message passing as a single function can be separated into middleware programs that different applications can communicate. That means middleware can stay out of operating systems, and communication protocols can provide higher-level but transparent interfaces (making applications to be reused easily and allowing programs written accessing heterogeneous database freely).

5.3.3 Reusability and scalability

When it comes to reusability in EI, there are two categories of software systems should be mentioned:

- A software application is a mechanism for packaging and physically deploying a collection of software functions that are designed to support one or more business processes.
- A software component is a piece of software that is its own encapsulation, and as such, it can be easily replaced by another piece of software that can potentially be reused in a number of different applications.

Reuse can occur in different levels ranging from source code to framework (even architecture) with their difficulties accelerate. A well-done architecture provides configures of resilient basis for a typical enterprise to allocate relative hardware, software and their connections. Especially at the representation level, the architecture can make for a concise but fundamental description and analysis on the whole enterprise system. As a result, the architecture to facilitate some low-level reuse can be reused in itself. Within enterprise integration activities, more emphasis should be put on architecture reuse which is more productive but most hard to achieve. As only limited experience and skills have been accumulated, it is a challenge to integrators.

Sometimes reusability and scalability are mixed up for reuse that can indeed carry out some functions of scalability, and both of them are key concerns in the design of resilience systems and integration [92, 169]. But scalability primarily refers to the mechanism to ensure resources ready for the time of organization changes in number or size. As an example of integration pattern, modularized ERP modelling has been developed to meet some scalability requirements. But the hierarchy systems and CAS thinking can also be exploited to realize EA scalability. For instance, enterprise can partition heterogeneous functions into separate applications sets, as the ordering function can be served by one set of applications, selling function served by another applications set. This will allow integrators to scale each set independently in the need of resources consumption for the function. Decomposing one process into different phases and connecting them up asynchronously may be another helpful method as it can avoid synchronous coupling and higher cost, as such, the ontology methods should be applied to tailor the integration scalability [104].

However, scalability has its two facets: the up scalability and down scalability, that means an EBS must be able to facilitate the business expansion and shrink, it must be flexible enough to handle economic booms and down turns [15, 29, 72, 77, 98, 107, 117].

5.3.4 Agility

Agility of EBS means information systems can provides tools to accommodate new ways of running businesses and, when necessary, to discard old ways of doing things [55, 127]. Agility is a never-ending quest to do things better than before, and better than the competition [73, 75]. Agility of EBS should provide resilient basis to meet with different demands or emerging events. It can be concluded into the following facets:

(1) Proactivity, which means that an entity has the ability to take steps in advance of changes as well as in responding to them. The ontological-orient SOA modelling has been applied [37] and may have potentials to enhance the proactivity of EBS in integration.

(2) Reactivity, which is the most common interpretation of agility, is the ability to properly responde to unexpected changes or events mostly reflected in software adaptation (the popularized ubiquitous computing) [70]. Reactivity relies on the smart response while proactivity places more emphasis on the solutions taken ready. Although the adaptivity can be partly achieved via reactivity, the former is fairly different with the latter. Reactivity focuses on sensitive awareness of circumstance and can be achieved via integration tools, while adaptivity is a strategy-related requirement on architecture to cope with complexity via holistic solutions.

(3) Learning. This is the most distinct attribute of agility, and is different from what is so often thought of as flexibility [45, 49, 51]. Although some tactical problems in integration can be soundly settled by autonomous technologies, to make the experience become the knowledge and memorized in the systems is the huge challenge.

A systemic approach to enhance the agility of EBS is to ascribe proactivity, reactivity to the basic and elementary components of system. As an example, with the SOA architecture and radio frequency identification (RFID) technology, events processing can be embedded in EBS to facilitate events aggregation into high level actionable information, improving total response capability of EBS and sometime will [115]. In addition, some useful cases or patterns can be coined by data mining and stored in knowledge bases, which will improve the learning capability of EBS [91]. The architecture for events processing in EBS has been proposed already and the workflow model is used to extract complex events patterns [167]. Combining RFID technology and some intelligent analysis technologies within EA may be a promising direction of enterprise integration.

5.4 Layers of specifications and relative issues

Layers of the integration standards and specifications are the crucial problem in EI. Basically speaking, EI activities can be identified by three orthogonal layers:

(1) The business logic layer, which implements the business rules that regulate the processes of an application system.

(2) The applications logic layer, which offers the ability to manage the interactions between an application system and its various presentation interfaces. These interfaces include web browsers, mobile computing devices, and other newly developed terminal devices.

(3) The data logic layer, which provides the capability to access and map data into a form that can be managed by business logic.

These layers of EI closely correspond to the EA hierarchical design [21]. The enforcement of EI planning requires

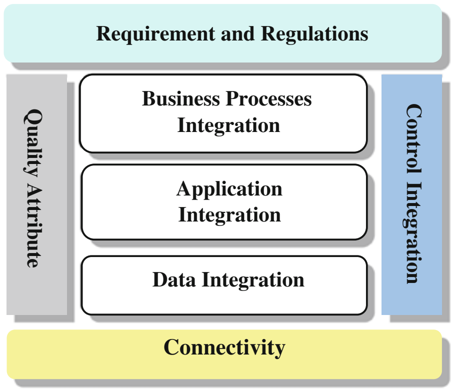

Fig. 4 Overall integration model of EIP

systematic approach for representing business, applications (services), and technology viewpoints. Thus, the planning methods should provide specific tactical solutions, including budgets and project plans. It also can forge overall understanding of infrastructure profiles, such as processors and storage, networking, data management etc. Many researchers [37, 80, 150] refer to applying EIP (enterprise integration platform) implementation methods to discriminate interrelationships among those technical issues. Here we make the conclusion and present an integration model of EIP in Fig. 4.

"Requirements and Regulations" deals with identifying business drivers and actions that dominate the whole development of the EI. Intense competitions are now driving organizations more aware of the need to capture and manage requirements. Each requirement (connectivity, reusability, scalability, and other QoS attributes) should be traced to one or more business drivers in a systematic fashion [61].

"Control integration" deals with different messaging rules between applications, and how these messages are manipulated based on different communication modes and protocols. "Connectivity" refers to data, workflow and service linkages, and how these linkages are handled by the application bridges and gateways, message-handling services, and other communication protocols. "Quality attributes" involve architectural decisions that influence quality features such as operation performance, connectivity, reusability, scalability even security. Quality attributes span several aspects of the integration model (as seen in the following figure). Quite often, quality attributes do not get enough attention and systemic consideration in EI activities, thus they are usually defined too late in the development process to hinder the whole EI development.

6 Conclusion

The research challenge in EA and EI is to conquer information systems complexity [35], and it has become the huge demand in modern EBS development. To meet this challenge, practitioners should carry out reformation throughout the enterprise e-business systems, ranging from the design mindset to the implementation activities. Systems theory and systems thinking as the theoretical foundation can provide multidimensional, hierarchical vision and methodology for EA and EI in EBS. This is the major effort of this paper. We attempted to demonstrate that systems theory can provide concrete support to EA and EI activities in EBS design and development. Building better enterprise e-business systems need to put the enterprise back into enterprise systems [38]. This means that not only analysts and developers, but also researchers should adopt systematic viewpoint when deploying their actions. As presented in this paper, systems theory has evoked an awareness of this importance in EA and EI for EBS.

As we have moved into the second decade of the 21st Century, the increasing popularity of E-Commerce, Cloud Computing and Internet of Things (IOT) demands more collaborative and complex enterprise applications and enterprise systems integration [90, 155, 156]. Hence, an EA in the framework of Industrial Information Integration Engineering (IIIE) which can unify the entire value chain to provide value-added services is in great need. Additionally, a robust, adaptive EBS with moderate agility will be the necessity of enterprise to compete in the future. As for the traditional systems architecture, innovation and relative technologies to exploit legacy systems and reduce integration cost are also in great demand. All of these require systematic exploration in EA and EI in EBS development, and call for further applications of systems theory to EBS. Therefore, systems theory and systems thinking are expected to and can make more contributions in this new era [146].

Acknowledgments This project was partially supported by the NSFC (National Natural Science Foundation of China) Grant 71132008, 70890080, 70890081 and 70971107, Changjiang Scholar Program of the Ministry of Education of China, and the US National Science Foundation Grant 1044845.

References

1. Ackoff RL (2006) Why few organizations adopt systems thinking. Syst Res Behav Sci 23(5):705–708
2. Aldina L, Cesarea D (2011) A literature review on business process modelling: new frontiers of reusability. Enterp Inf Syst 5(3):359–383
3. An L, Yan J, Tong L (2008) Methodology for web services adoption based on technology adoption theory and business process analyses. Tsinghua Sci Technol 13(3):383–389
4. Armour F, Kaisler S (2001) Enterprise architectures: agile transition and implementation. IEEE IT Prof 2(6):30–37
5. Arroyo S, Sicilia M, Lopez-Cobo J (2008) Patterns of message interchange in decoupled hypermedia systems. J Netw Comput Appl 31(2):75–92

6. Auyang SY (1998) Foundations of complex-system theories: in economics, evolutionary biology, and statistical physics. Cambridge University Press, New York
7. Aversano L, Canfora G, Cimitile A, De Lucia A (2001) Migrating legacy systems to the web: an experience report. In: 5th European conference on Software maintenance and reengineering. IEEE Computer Society, Washington, DC, pp 148–157
8. Bala I, Shankaranarayanan G, Lenard M (2005) Model management decision environment: a web service prototype for spreadsheet models. Decis Support Syst 40(2):283–304
9. Baldwin CY, Clark KB (2000) Design rules: the power of modularity. MIT Press, Cambridge
10. Bauer B, Muller J, Roser S (2008) Decentralized business process modeling and enactment: ICT architecture topologies and decision methods. In: Programming multi-agent systems. Lecture Notes in Computer Science, vol 4908. Springer, Berlin, pp 1–26
11. Beer S (1979) The heart of enterprise. Wiley, London
12. Beer S (1985) Diagnosing the system for organizations. Wiley, Chichester
13. Beer S (2000) The heart of enterprise. Wiley, Chichester
14. Beheshti HM, Beheshti CM (2010) Improving productivity and firm performance with enterprise resource planning. Enterp Inf Syst 4(4):445–472
15. Bernus P (2003) Enterprise models for enterprise architecture and ISO9000: 2000. Annu Rev Control 27:211–220
16. Bougettaya A, Malik Z, Rezgui A, Korff L (2006) A scalable middleware for web databases. J Database Manag 17(4):20–46
17. Brancheau JC, Wetherbe J (1987) Key issues in information systems management. MIS Q 11(1):23–45
18. Bubenko J, Dahlstrand I (2005) An interview with Börje Langefors from SARA to TAIS. In: Bubenko J, Impagliazzo J, Sølvberg A (eds) History of nordic computing. IFIP international federation for information processing, vol 174. Springer, Boston, pp 7–22
19. Camarinha-Matos L, Afsarmanesh H, Lima C (1999) Hierarchical coordination in virtual enterprise infrastructures. J Intell Robotic Syst 26(3–4):267–287
20. Chalmers RE (1999) Small manufacturers seek best ERP fit. Manuf Eng 10:42–46
21. Chari K, Seshadri S (2004) Demystifying Integration. Commun ACM 47(7):58–63
22. Charles J (1999) Middleware moves to the forefront. IEEE Comput 32(5):17–19
23. Platt A, Warwick S (1995) Review of soft systems methodology. Ind Manag Data Syst 95(4):19–21
24. Checkland PB (1999) Systems thinking. In: Currie WL, Galliers RD (eds) Rethinking management information systems: an interdisciplinary perspective. Oxford University Press, Oxford, pp 45–56
25. Checkland PB (2000) Soft systems methodology: a thirty year retrospective. Syst Res Behav Sci 17(1):11–58
26. Checkland PB (2001) Soft systems methodology. In: Rosenhead J, Mingers J (eds) Rational analysis in a problematic world revisited. Wiley, Chichester, pp 61–89
27. Chen D, Doumeingts G, Vernadat F (2008) Architectures for enterprise integration and interoperability: past, present and future. Comput Ind 59(7):647–659
28. Chen D, Vernadat F (2004) Standards on enterprise integration and engineering-state of the art. Int J Comput Integr Manuf 17(3):235–253
29. Chen G, Ren W, Zhang JB, Yang Z, et al. (2008) Developing rule-enhanced dynamic virtual enterprise integration frameworks. In: IEEE 8th international conference on computer and information technology workshops. IEEE Computer Society, Washington, DC, pp 694–699
30. Chen J (1999) Building CORBA applications with an object database system. Theory Pract Object Syst 5(4):209–215
31. Cheung W, Babin G (2006) A metadatabase-enabled executive information system (Part A): a flexible and adaptable architecture. Decis Support Syst 42(3):1589–1598
32. Chiang D, Lin C, Chen M (2011) The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. Enterp Inf Syst 5(2):219–234
33. Chu XN, Tso SK, Zhang WJ, Li Q (2002) Partnership synthesis for virtual enterprises. Int J Adv Manuf Technol 19(5):384–391
34. Contreras M, Sheremetov L (2008) Industrial application integration using the unification approach to agent-enabled semantic SOA. Robotics Comput Integr Manuf 24(5):680–695
35. CRA (2002) Grand research challenges in information systems. Available at http://archive.cra.org/reports/gc.systems.pdf. Accessed 20 Mar 2012
36. Cragg P, Zinatelli N (1995) The evolution of information systems in small firms. Inf Manag 29(1):1–8
37. Daniel S, Haluk D, Michael G (2007) An interorganizational knowledge-sharing security model with breach propagation detection. Inf Syst Front 9(5):469–479
38. Davenport T (1998) Putting the enterprise into the enterprise system. Harv Bus Rev 76(4):121–131
39. David M, Ali A (2007) Architectural valuation using the design structure matrix and real options theory. Concurr Eng 15(2): 157–173
40. Day B, Lutteroth C (2011) Climbing the ladder: capability maturity model integration level 3. Enterp Inf Syst 5(1):125–144
41. Desai A (2005) Adaptive complex enterprise. Commun ACM 48(4):32–35
42. Eardley A, Avison D, Powell P (1997) Strategic information systems: an analysis of development techniques which seek to incorporate strategic flexibility. J Organ Comput 7(1):57–77
43. Engelsman W, Quartel D, Jonkers H, van Sinderen M (2011) Extending enterprise architecture modelling with business goals and requirements. Enterp Inf Syst 5(1):9–36
44. Enns S (2001) MRP performance effects due to lot size and planned lead time settings. Int J Prod Res 39(3):461–480
45. Erol O, Sauser B, Mansouri M (2010) A framework for investigation into extended enterprise resilience. Enterp Inf Syst 4(2):111–136
46. Evans J (1991) Strategic flexibility for high technology manoeuvres: a conceptual framework. J Manag Stud 28(1):69–89
47. Fayad M, Hamu D, Brugali D (2000) Enterprise frameworks characteristics, criteria, and challenges. Commun ACM 43(10):39–46
48. Fong E, Goldfine A (1989) Information management directions: the integration challenge. ACM SIGMOD Rec 18(4):40–43
49. Fox MS, Barbuceanu M, Teigen R (2000) Agent-oriented supply-chain management. Int J Flex Manuf Syst 12(2–3):165–188
50. Geihs K (2001) Middleware challenges ahead. IEEE Comput 34(6):24–31
51. Goldman S, Nagel R, Preiss K (1995) Agile competitors and virtual organizations: strategies for enriching the customer. Von Nostrand Reinhold, New York
52. Gong Z, Muyeba M, Guo J (2010) Business information query expansion through semantic network. Enterp Inf Syst 4(1):1–22
53. Graham J, Decker K, Mersic M (2003) DECAF—a flexible multi-agent system architecture. Auton Agent Multi Agent Syst 7(1):7–27
54. Gudivada VN, Nandigam J (2005) Enterprise application integration using extensible web services. In: Proceedings of the IEEE international conference on web services. IEEE Computer Society, Washington, DC, pp 41–48
55. Gunasekaran A, Yusuf Y (2002) Agile manufacturing: a taxonomy of strategic and technological imperatives. Int J Prod Res 40(6):1357–1385

56. Guo J, Gong Z (2011) Measuring virtual wealth in virtual worlds. Inf Technol Manag 12(2):121–135
57. Guo J, Xu L, Gong Z, Che C, Chaudhry S (2012) Semantic inference on heterogeneous e-marketplace activities. IEEE Trans Syst Man Cybern Part A Syst Hum 42(2):316–330
58. Hasselbring W (2000) Information system integration. Commun ACM 43(6):32–38
59. Holland JH (1992) Complex adaptive systems. Daedalus 121(1):17–30
60. Holland JH (1996) Hidden order: how adaptation builds complexity. Addison Wesley Longman Publishing, MA
61. Hong D, Chiu DKW, Shen VY, Cheung SC, Kafeza E (2007) Ubiquitous enterprise service adaptations based on contextual user behavior. Inf Syst Front 9(4):343–358
62. Hou H, Hu M, Chen L, Choi JY (2011) An enhanced model framework of personalized material flow services. Inf Technol Manag 12(2):149–159
63. Hsu C, Cho J, Yee L, Rattner L (1995) Core information model: a practical solution to costly integration problems. Comput Ind Eng 28(3):523–544
64. Insaurralde CC, Seminario MA, Jimenez JF, Giron-Sierra JM (2008) Computer tool with a code generator for avionic distributed fuel control systems with smart sensors and actuators. IEEE Trans Syst Man Cybern Part C 38(3):431–445
65. Iyer B, Gottlieb R (2004) The four-domain architecture: an approach to support enterprise architecture design. IBM Syst J 43(3):587–597
66. Izza S (2009) Integration of industrial information systems: from syntactic to semantic integration approaches. Enterp Inf Syst 3(1):1–57
67. Janssen M, Kuk G (2006) A Complex adaptive system perspective of enterprise architecture in electronic government. In: Proceedings of the 39th Hawaii international conference on system sciences. IEEE Computer Society, Washington, DC, pp 71–72
68. Johansson B, Newman M (2010) Competitive advantage in the ERP system's value-chain and its influence on future development. Enterp Inf Syst 4(1):79–93
69. Kaetker S, Geihs K (1997) A Generic model for fault isolation in integrated management systems. J Netw Syst Manag 5(2):109–130
70. Kakousis K, Paspallis N, Papadopoulos G (2010) A survey of software adaptation in mobile and ubiquitous computing. Enterp Inf Syst 4(4):355–389
71. Kamogawa T, Okada H (2005) A framework for enterprise architecture effectiveness. In: Proceedings of 2005 international conference on services systems and services management. IEEE Computer Society, Washington, DC, pp 740–745
72. Kishore R, Zhang H, Ramesh R (2006) Enterprise integration using the agent paradigm: foundations of multi-agent-based integrative business information systems. Decis Support Syst 42(1):48–78
73. Koh S, Gunasekaran A (2006) A knowledge management approach for managing uncertainty in manufacturing. Ind Manag Data Syst 106(4):439–459
74. Kosanke K (2006) ISO standards for interoperability: a comparison. In: Konstantas D, Bourrières J-P (eds) Interoperability of enterprise software and applications. Springer, London, pp 55–64
75. Kulkarni V, Reddy S (2006) A model-driven architectural framework for integration-capable enterprise application product lines. In: Rensink A, Warmer J (eds) Model driven architecture—foundations and applications. Lecture Notes in Computer Science, vol 4066. Springer, Berlin, pp 1–12
76. Kumar S, Kadow BB, Lamkin M (2011) Challenges with the introduction of radio-frequency identification systems into a manufacturer's supply chain—a pilot study. Enterp Inf Syst 5(2):235–253
77. Lankhorst MM (2004) Enterprise architecture modelling—the issue of integration. Adv Eng Inform 18(4):205–216
78. Lê LS, Wegmann A (2005) Definition of an object-oriented modeling language for enterprise architecture. In: Proceedings of the 38th Hawaii international conference on system sciences. IEEE Computer Society, Washington, DC, pp 222
79. Lee J, Siau K, Hong S (2003) Enterprise integration with ERP and EAI. Commun ACM 46(2):54–60
80. Lee J, Barry H, Kantowitz B (2005) Network analysis of information flows to integrate in-vehicle information systems. Int J Veh Inf Commun Syst 1(1–2):24–43
81. Lenny S, Saad M (2006) Managing uncertainty in ERP-controlled manufacturing environments in SMEs. Int J Prod Econ 101(1):109–127
82. Li F, Li L, Jin C, Wang R, Wang H, Yang L (2012) A 3PL supplier selection model based on fuzzy sets. Comput Oper Res 39(8):1879–1884
83. Li L (2007) Supply chain management: concepts, techniques and practices. World Scientific, Hackensack
84. Li L (2011) Introduction: advances in e-business engineering. Inf Technol Manag 12(2):49–50
85. Li L (2011) Assessing the relational benefits of logistics services perceived by manufacturers in supply chain. Int J Prod Econ 132(1):58–67
86. Li L (2012) Effects of enterprise technology on supply chain collaboration: analysis of China-linked supply chain. Enterp Inf Syst 6(1):55–77
87. Li L, Su Q, Chen X (2011) Ensuring supply chain quality performance through applying the SCOR model. Int J Prod Res 49(1):33–57
88. Li L, Warfield J (2011) Perspectives on quality coordination and assurance in global supply chains. Int J Prod Res 49(1):1–4
89. Li L, Xu L, Jeng HA, Naik D, Allen T, Frontini M (2008) Creation of environmental health information system for public health service: a pilot study. Inf Syst Front 10(5):531–543
90. Li S, Xu L, Wang X, Wang J (2012) Integration of hybrid wireless networks in cloud services oriented enterprise information systems. Enterp Inf Syst 6(2):165–187
91. Liu B, Cao SG, He W (2011) Distributed data mining for e-business. Inf Technol Manag 12(2):67–79
92. Liu D, Deters R, Zhang WJ (2010) Architectural design for resilience. Enterp Inf Syst 4(2):137–152
93. Liu W, Du Y, Yan C (2012) Soundness preservation in composed logical time workflow nets. Enterp Inf Syst 6(1):95–113
94. Lyman D, Drew P (1995) Active database management of global data integrity constraints in heterogeneous database environments. In: Proceedings of the 11th international conference on data engineering. IEEE Computer Society, Washington, DC, pp 99–108
95. Ma J, Wang K, Xu L (2011) Modelling and analysis of workflow for lean supply chain. Enterp Inf Syst 5(4):423–447
96. McCarthy I, Menicou M (2002) A classification schema of manufacturing decisions for the GRAI enterprise modelling technique. Comput Ind 47(3):339–355
97. Mietzner R, Leymann F, Unger T (2011) Horizontal and vertical combination of multi-tenancy patterns in service-oriented applications. Enterp Inf Syst 5(1):59–77
98. Mirja P, Anton N, Kari L (2007) Managing information security in a business network of machinery maintenance services business—enterprise architecture as a coordination tool. J Syst Softw 80(10):1607–1620
99. Mowshowitz A (1997) Virtual organization. Commun ACM 40(9):30–37

100. Mukherji A (2002) The evolution of information systems: their impact on organizations and structures. Manag Decis 40(5): 497–507
101. Narman P, Holm H, Johnson P, Konig J, Chenine M, Ekstedt M (2011) Data accuracy assessment using enterprise architecture. Enterp Inf Syst 5(1):37–58
102. Ng JKC, Ip WH, Lee TC (1998) Development of an enterprise resources planning system using a hierarchical design pyramid. J Intell Manuf 9(5):385–399
103. Nikolaidou M, Alexopoulou N, Tsadimas A, Dais A, Anagnostopoulos D (2006) A consistent framework for enterprise information system engineering. In: the 10th IEEE international enterprise distributed object computing conference. IEEE Computer Society, Washington, DC, pp 492–496
104. Oh S, Yeon HY, Ahn J (2011) Cohesion and coupling metrics for ontology modules. Inf Technol Manag 12(2):81–96
105. O'Leary E, Kuokka D, Robert P (1997) Artificial intelligence and virtual organization. Commun ACM 40(1):2–59
106. Oreizy P, Gorlick MM, Taylor RN, Heimhigner D et al (1999) An architecture-based approach to self-adaptive software. IEEE Intell Syst 14(3):54–62
107. Panetto H, Molina A (2008) Enterprise integration and interoperability in manufacturing systems: trends and issues. Comput Ind 59(7):641–646
108. Papazoglou MP, Georgakopoulos D (2003) Service oriented computing. Commun ACM 46(10):25–28
109. Patankar AK, Adiga S (1995) Enterprise integration modelling: a review of theory and practice. Comput Integr Manuf Syst 8(1):21–34
110. Peristeras V, Tarabanis K (2000) Towards an enterprise architecture for public administration using a top-down approach. Eur J Inf Syst 9(4):252–260
111. Petrie CJ (1992) Enterprise integration modeling: proceedings of the first international conference. MIT Press, Cambridge
112. Piao C, Han X, Wu H (2010) Research on e-commerce transaction networks using multi-agent modelling and open application programming interface. Enterp Inf Syst 4(3):329–353
113. Puschmann T, Alt R (2004) Enterprise application integration systems and architecture—the case of the Robert Bosch Group. J Enterp Inf Manag 17(2):105–116
114. Qian XS (2007) Systems engineering. Shanghai Jiaotong University Press, Shanghai
115. Qian Y, Jin B, Fang W (2011) Heuristic algorithms for effective broker deployment. Inf Technol Manag 12(2):55–66
116. Quartel D, Steen M, Lankhorst M (2012) Application and project portfolio valuation using enterprise architecture and business requirements modelling. Enterp Inf Syst 6(2):189–213
117. Radha M (2005) Maintaining enterprise software applications. Commun ACM 48(11):75–79
118. Railsback S (2001) Concepts from complex adaptive systems as a framework for individual-based modeling. Ecol Model 139(1): 47–62
119. Repoussis P, Paraskevopoulos D, Zobolas G, Tarantilis CD, Ioannou G (2009) A web-based decision support system for waste lube oils collection and recycling. Eur J Oper Res 195(3):676–700
120. Richardson G, Jackson B, Dickson G (1990) A principle-based enterprise architecture: lessons from texaco and star enterprise. MIS Q 14(4):385–403
121. Rohloff M (2005) Enterprise architecture—framework and methodology for the design of architectures in the Large. In: European conference on Information systems ECIS 2005, AIS, pp 113–126
122. Ross AM, Hastingg DE, Joyce MW (2004) Multi-attribute tradespace exploration as front end for effective space system design. J Spacecr Rocket 41(1):20–28
123. Sanchez R (1999) Modular architectures in the marketing process. J Mark 63(5):92–111
124. Sepehri M (2012) A grid-based collaborative supply chain with multi-product multi-period production-distribution. Enterp Inf Syst 6(1):115–137
125. Shafiei F, Sundaram D (2004) Multi-enterprise collaborative enterprise resource planning and decision support systems. In: Proceedings of the 37th Hawaii international conference on System sciences. IEEE Computer Society, Washington, DC, pp 80228
126. Shan S, Wang L, Wang J, Hao Y, Hua F (2011) Research on e-government evaluation model based on the principal component analysis. Inf Technol Manag 12(2):173–185
127. Sharp J, Irani Z, Desai S (1999) Working towards agile manufacturing in the UK industry. Int J Prod Econ 62(1):155–169
128. Shen C, Chou C (2010) Business process re-engineering in the logistics industry: a study of implementation, success factors, and performance. Enterp Inf Syst 4(1):61–78
129. Siegel M, Madnick S (1991) Context interchange: sharing the meaning of data. ACM SIGMOD Rec 20(4):77–78
130. Smith D, O'Brien L, Barbacci M, Coallier F (2002) A roadmap for enterprise integration. In: Proceedings of the 10th international workshop on software technology and engineering practice. IEEE Computer Society, Washington, DC, pp 94–102
131. Stamatelopoulos F, Maglaris B (1999) Performance and efficiency in distributed enterprise management. J Netw Syst Manag 7(1):47–71
132. Tan WA, Shen W, Xu L, Zhou B, Li L (2008) A business process intelligence system for enterprise process performance management. IEEE Trans Syst Man Cybern Part C 38(6): 745–756
133. Tan W, Xu Y, Xu W, Xu L, Zhao X, Wang L, Fu L (2010) A methodology toward manufacturing grid-based virtual enterprise operation platform. Enterp Inf Syst 4(3):283–309
134. Thacker RM (1989) A new CIM model: a blueprint for the computer-integrated manufacturing enterprise. Society of Manufacturing, Dearborn
135. The Open Group (2003) The open group architecture framework (Version 9.1 Enterprise Edition). Available at http://www.opengroup.org/architecture/togaf/#download. Accessed on 20 Mar 2012
136. Tiwari M, Mondal S (2002) Application of an autonomous agent network to support the architecture of a holonic manufacturing system. Int J Adv Manuf Technol 20(12):931–942
137. Tyree J, Akerman A (2005) Architecture decisions: demystifying architecture. IEEE Softw 22(2):19–27
138. van Sinderen M, Almeida J (2011) Empowering enterprises through next-generation enterprise computing. Enterp Inf Syst 5(1):1–8
139. Vernadat F (1996) Enterprise modeling and integration: principles and applications. Chapman & Hall, London
140. Versteeg G, Bouwman H (2006) Business architecture: a new paradigm to relate business strategy to ICT. Inf Syst Front 8(2):91–102
141. Vinoski S (2002) Middleware "Dark Matter". IEEE Internet Comput 6(5):92–95
142. Waldner JB (1992) CIM: principles of computer-integrated manufacturing. Wiley, Chichester
143. Wang C, Xu L (2008) Parameter mapping and data transformation for engineering application integration. Inf Syst Front 10(5):589–600
144. Wang HJ, Wu H (2011) Supporting process design for e-business via an integrated process repository. Inf Technol Manag 12(2):97–109
145. Wang X, Wang H, Wang H, Zhang L, Cao X (2011) Constructing a decision support system for management of employee turnover risk. Inf Technol Manag 12(2):187–196

146. Warfield JN (2007) Systems science serves enterprise integration: a tutorial. Enterp Inf Syst 1(2):235–254
147. Wegmann A, Preiss O (2003) MDA in enterprise architecture? The living system theory to the rescue. In: Proceedings of the 7th IEEE international enterprise distributed object computing conference. IEEE Computer Society, Washington, DC, pp 2–13
148. Weill P (2007) Innovating with information systems: what do the most agile firms in the world do? Available at http://www.iese.edu/en/files/6_29338.pdf. Accessed on 20 Mar 2012
149. Wu SL, Xu L, He W (2009) Industry-oriented enterprise resource planning. Enterp Inf Syst 3(4):409–424
150. Xiao S, Lin Z, Peng-fei Y (2008) Semantic SOA-based enterprise information system integration technology. In: International conference on smart manufacturing application 2008. IEEE Computer Society, Washington, DC, pp 534–537
151. Xu L (1987) Toward escape from the limitations of economic systems analysis: introduction of dimensionality. Syst Res 4(4):243–250
152. Xu L (1992) Systems characteristics in information systems design. Syst Res 9:67–78
153. Xu L (1995) Systems thinking for information systems development. Syst Pract 8:577–589
154. Xu L (2000) The contribution of systems science to information systems research. Syst Res Behav Sci 17(2):105–116
155. Xu L (2011) Enterprise systems: state-of-the-art and future trends. IEEE Trans Ind Inf 7(4):630–640
156. Xu L (2011) Information architecture for supply chain quality management. Int J Prod Res 49(1):183–198
157. Xu L, Li L (1989) Complementary opposition as a systems concept. Syst Res 6(2):91–101
158. Xu L, Liang N, Gao Q (2008) An integrated approach for agricultural ecosystem management. IEEE Trans Syst Man Cybern Part C 38(2):1–10
159. Xu L, Liu H, Wang S, Wang K (2009) Modelling and analysis techniques for cross-organizational workflow systems. Syst Res Behav Sci 26(3):367–389
160. Xu L, Swanson GA, Samuelson K (2009) Systems science and enterprise integration, technological economics and the theory of material flow. Syst Res Behav Sci 26(2):123–127
161. Xu L, Tan WA, Zhen H, Shen W (2008) An approach to enterprise process dynamic modeling supporting enterprise process evolution. Inf Syst Front 10(5):611–624
162. Xu L, Wang C, Luo X, Shi Z (2006) Integrating knowledge management and ERP in enterprise information systems. Syst Res Behav Sci 23(2):147–156
163. Yang Z, Zhang JB, Gay R, Zhuang L, Lee HM (2005) Building a semantic-rich service-oriented manufacturing environment. In: Ngu A, Kitsuregawa M (eds) Web information systems engineering. Lecture Notes in Computer Science, vol 3806. Springer, Berlin, pp 623–632
164. Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26(3):276–292
165. Zachman JA (1997) Enterprise architecture: the issue of the century. Database Program Des
166. Zachman JA (2008) John Zachman's concise definition of the enterprise framework. Available at http://test.zachmaninternational.com/index.php/the-zachman-framework. Accessed on 20 Mar 2012
167. Zang C, Fan Y (2007) Complex event processing in enterprise information systems based on RFID. Enterp Inf Syst 1(1):3–23
168. Zdravković M, Panetto H, Trajanović M, Aubry A (2011) An approach for formalising the supply chain operations. Enterp Inf Syst 5(4):401–421
169. Zhang WJ, Lin Y (2010) On the principle of design of resilient systems—application to enterprise information systems. Enterp Inf Syst 4(2):99–110
170. Zhang Y, Li Z, Xu L, Wang J (2011) A new method for automatic synthesis of tolerances for complex assemblies based on polychromatic sets. Enterp Inf Syst 5(3):337–358
171. Zhou X, Chaudhry PE, Zhu Z, Li F (2005) Development of a financial management system in a CIMS environment. Int J Prod Res 43(12):2603–2613