Integration of Distributed Enterprise Applications: A Survey

Wu He and Li Da Xu, *Senior Member, IEEE*

***Abstract*—Many industrial enterprises acquire disparate systems and applications over the years. The need to integrate these different systems and applications is often prominent for satisfying business requirements and needs. In an effort to help researchers in industrial informatics understand the state-of-the-art of the enterprise application integration, we examined the architectures and technologies for integrating distributed enterprise applications, illustrated their strengths and weaknesses, and identified research trends and opportunities in this increasingly important area.**

***Index Terms*—Distributed enterprise applications, enterprise application integration, enterprise service bus, industrial informatics, industrial information integration engineering, Internet of things (IoT), middleware, enterprise systems, radio frequency identification (RFID).**

I. Introduction

A DISTRIBUTED enterprise application is usually defined as an enterprise application with multiple software modules or components running on two or more computers in an enterprise network [1]. Oftentimes the enterprise network is heterogeneous and is composed of diverse computers, devices, and operating systems. In industrial enterprise environments, many industry systems typically consist of numerous technologies, protocols, applications, and devices which are distributed across a network [2], [3]. As the industry environments become increasingly distributed and heterogeneous across multiple organizational and geographical boundaries in recent years, there are strong demands to integrate various distributed applications in order to enhance or increase enterprises' competitiveness. Particularly, integrating distributed industrial applications has attracted much attention in the industrial information system area. For example, according to IEC61499 specification [4], a distributed enterprise control system often consists of multiple applications that may be distributed among multiple networked devices. Oftentimes control processing application resides in a device, but output conversion application resides in another device. Sometimes a function of an application may be distributed to several devices and requires the cooperation of different parts to work properly. Additionally, the applications and devices may be developed or provided by different vendors [3] with different programming languages, formats, and protocols. Significant integration efforts are required to enhance and increase the interoperability of these applications and devices.

So far, many enterprises have invested heavily to integrate distributed enterprise applications due to the continuous mergers and acquisitions, joint venture, outsourcing, corporate restructuring, infrastructure upgrades, adoption of mobile devices, embedded devices, and wireless sensors. Integrating various applications owned by the enterprise can enhance or increase the enterprise's competitive advantages [5].

Distributed enterprise applications typically require their distributed components to interact with one another through certain remote communication mechanisms such as message-passing, and remote-invocations [6] in networking environments. However, distributed enterprise applications often have issues in communicating with one another due to issues with different formats and protocols. As distributed applications in an enterprise continue to grow, integrating distributed enterprise applications has become a challenging task. For example, many industry enterprises have trouble in integrating industry applications effectively. Such applications include computer-aided design system, engineering document management system, manufacturing execution system, and product data management system running on different hosts. For many industry enterprises, it is imperative for these industry systems to cooperate for achieving certain business objectives. To address integration problems, an IT solution named Enterprise Application Integration (EAI) has been developed to help achieve quality integration [5].

EAI encompasses technologies that enable distributed and heterogeneous applications to interact to one another across the network and help integrate many individual applications into a whole [5]. It consists of methods and tools to coordinate various applications and to support the integration of both intra-organizational and inter-organizational systems. EAI solutions support the integration of business processes and data across various enterprise applications [7]. With EAI, intra- or inter-enterprise application systems can be integrated effectively, and can ensure that different divisions, units or even different enterprises can cooperate with each other [8].

EAI is a key research area in Industrial Information Integration Engineering (IIIE). Broadly speaking, IIIE involves concepts and techniques that can be used to streamline or facilitate the process of integrating various industrial information sources [8]–[16]. In 1996, the International Electronics Manufacturing

Manuscript received December 23, 2011; accepted February 21, 2012. Date of publication March 06, 2012; date of current version December 12, 2014. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 71132008, the Changjiang Scholar Program of the Ministry of Education of China, and the U.S. National Science Foundation under Grant 1044845. Paper no. TII-11-1052.

W. He is with Old Dominion University, Norfolk, VA 23529 USA (e-mail: whe@odu.edu).

L. D. Xu is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with Old Dominion University, Norfolk, VA 23529 USA (e-mail: lxu@odu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2012.2189221

Initiative (iNEMI) revealed that many industry systems did not cooperate with each other [17]. Afterwards, substantial efforts were made to integrate disparate applications and systems in industry sectors. Nowadays many industries such as telecommunication, manufacturing, logistics, and energy have implemented and integrated distributed enterprise applications such as distributed control systems in factory automation, e-manufacturing systems, and distributed electronics production systems [3], [18]–[20]. For example, a factory management system may include an order processing application, a production scheduling application and a production control application which reside in three separate servers [21]. The three distributed applications then cooperate with each other to achieve their planned business purpose. The literature also shows that distributed enterprise applications sometimes perform better or required less resources than a large centralized software system [19] for some industrial applications such as manufacturing control software and manufacturing execution systems due to the increased demand for agility, flexibility and scalability, dispersed users in various locations, and unanticipated system change [1], [19].

In an effort to help readers understand the current status and future trends of integrating enterprise application in industries, we reviewed the past, present and future development related to the enterprise architectures and enabling technologies for integrating distributed enterprise applications. The main topics of interest to industrial informatics readers include distributed enterprise architectures, middleware technologies for integrating distributed enterprise applications in industries, and research trends and challenges involved in the integration of distributed enterprise applications. We hope this paper can help readers become more aware of the challenges and opportunities that exist in this increasingly important area and bring their expertise to help address research challenges for integrating various enterprise applications in industries.

The rest of this paper is organized as follows. Section II provides an introduction to the historical development of distributed enterprise application architectures. Section III provides a brief overview about the major distributed application integration technologies developed in the past decades. Recent research work and progress on enterprise application integration technologies are presented in Section IV. Future trends and research challenges are presented and discussed in Section V. Conclusions are given in Section VI.

II. Historical Development: Distributed Enterprise Application Architectures

Distributed enterprise application architectures have undergone an extensive evolution. Early generation enterprise applications were built on centralized mainframes [22], [23]. As the capacity of personal computers increased, many applications and tasks were moved to the user's computers to better satisfy the business or processing needs. As a result, first-generation distributed enterprise applications were mainly developed using a simple two-tier client/server networked infrastructure [23], [24]. As the amounts of data continue to increase, a three-tier architecture became popular in enterprise application development in the mid-late 1990s [22]. On a three-tier architecture, software components are divided into three layers: a user interface layer, an application processing layer (middle tier), and a database management and processing layer [22]. Middleware such as CORBA are often deployed to the middle tier to integrate distributed enterprise applications including independently developed applications [22], [23]. In addition, transaction processing (TP) monitors often run on the middle tier for scalability, workload, and resource balancing needs [22].

As Web applications become widespread, the three-tier architecture was extended by adding web clients (mainly browsers) and web servers (e.g., Apache) [14]. Furthermore, the three-tier architecture continues to evolve and subsequently the multitier architecture appears. For example, additional tiers are often introduced between client and other layers such as data layer for security, workload and resource balancing, and performance monitoring [22], [23].

III. Integration of Distributed Enterprise Applications: The Past

This paper mainly concerns application integration on heterogeneous platforms. The research community came up with the EAI solution to help achieve quality application integration. EAI provides ways to integrate heterogeneous applications on different systems and platforms [8].

Integration can be studied through different dimensions including integration scope, integration point of view (views from users, designers and developers), integration layer and integration level [22]. Furthermore, intra-enterprise integration can be divided into horizontal and vertical integration [21], [22]. Inter-enterprise integration includes business to business and business to customer integration [22]. Integration levels can be at hardware, platform, syntactical, and semantic levels [22]. From the technology perspective, many researchers found it useful to examine the system integration by focusing on different layers [25]. Below is a brief introduction to integration on different layers.

A. Communication Layer Integration

Integrating distributed applications requires those separate applications to be able to communicate with each other and to exchange data and information. For example, an application may need to know the current status and operations of a remote application in order to perform certain tasks such as scheduling. Typically, protocols such as HTTP and IIOP are used to facilitate information exchange among different applications [25].

B. Data Integration

Research on data integration mainly deals with moving or federating data between different types of data sources [22], [26]. Data integration involves a lot of data mapping and conversion among different elements including data source schema, targeted data schema, and the mapping relationship between data source schema and targeted data schema [25], [27]. The drawback of data integration is that developers will have to understand and maintain the underlying schemas regularly to address any changes [27].

C. Business Logic Integration

Integration at this layer includes integration in the sublayers such as basic coordination, functional interfaces, business

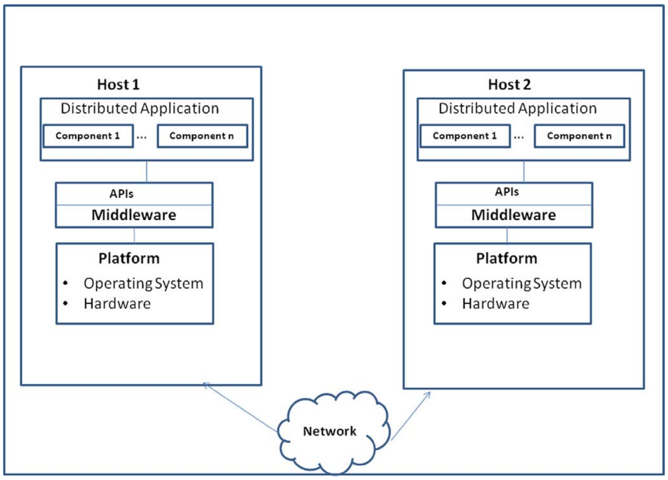

Fig. 1. The use of middleware in distributed applications.

protocol and policies, and nonfunctional properties [25]. A traditional way for the application integration involves complex system and network programming [25], [27]. Thus, maintaining the traditional integrated enterprise system was quite difficult to developers. To make the application integration at the business logic level easier, the research community mainly focuses on the development of middleware technologies. A number of middleware technologies have been developed to build and integrate distributed enterprise applications [28] in the past two decades. Middleware is designed to help developers and make distributed systems design less challenging [28], [29]. By adopting the middleware, we can easily separate applications from the underlying operating systems, hardware, and network environments [2]. Due to the advantages brought by middleware, the literature has witnessed an extensive use of middleware technologies in industrial environments [3], [6], [21]. As an important integration technology, middleware technologies are often used by industrial enterprises to integrate new applications, emerging technologies and legacy applications [8]. Typically, a middleware for communication comprises two types of remote communications: message-passing, and/or remote-invocation [6]. Additionally, middleware can provide functions to ensure reliability, scalability, and performance to enterprise systems [28], [29]. Fig. 1 illustrates the use of middleware in distributed applications.

There are many types of middleware such as RPC-based middleware, message-oriented middleware, event-based middleware, database middleware, transaction processing (TP) monitors, security middleware, agent-based middleware, and service oriented middleware [22], [30], [31]. In addition, some large companies such as SAP implement their own custom middleware as part of their application solution. Each of these technologies and approaches has their own advantages and disadvantages. As custom middleware has limited system portability, interoperability, and configurability [2], our review mainly focus on general purpose middleware technologies. Our review shows that the most commonly used general purpose middleware technologies at the business logic layer for integrating distributed enterprise applications include the RPC-based (remote invocation) middleware and message-oriented middleware [6], [29]. Below is a brief introduction to some of the major middleware technologies.

1) Distributed Computing Environment (DCE): DCE was designed and developed to support distributed applications in heterogeneous environments in early 1990's [32], [33]. DCE technology includes multiple components such as RPC, Directory Services, Time Service, and many other Services [32], [33]. DCE is supported by many different platforms and legacy operating systems. A main advantage of DCE is that the DCE RPC facility enables software components on different platforms to interact [22]. Compared to traditional networking programming methods, RPC is relatively simpler to code [32]. However, DCE does not have strong support mechanisms for object-oriented languages because RPC is designed to be procedural since the beginning [33]. As a result, DCE has lost its popularity in recent years [33].

2) Distributed Component Object Model (DCOM): DCOM is a technology developed by Microsoft to improve communication among software modules or components on the network [32]. Using DCOM, two objects on two separate computers are able to call each other's methods. DCOM supports object-oriented languages like C++ and Java. DCOM is well supported by Windows platforms [33]. However, DCOM only supports a few non-Windows operating systems, which limits the use of DCOM for heterogeneous networks [33].

3) Common Object Request Broker Architecture (CORBA): CORBA was invented in the early 1990s. Different from DCOM, CORBA is platform and language independent. CORBA supports a variety of systems and programming languages and has ability to integrate legacy software applications [33]. Object Request Broker (ORB), a core component of CORBA, is responsible for passing client side requests to applications on remote servers [22], [33]. CORBA provides specifications to help developers develop distributed applications across diverse platforms and environments [22]. CORBA had applications in many domains including telecommunications, finance, medicine, and manufacturing [35]. However, CORBA was complicated and difficult to use properly. As a result, the interest in CORBA has declined sharply [33] and CORBA is no longer popular in the middleware marketplace. But some strengths of CORBA have been incorporated into technologies such as J2EE and Web services [33].

4) Java Remote Method Invocation (RMI): Java RMI was released by Sun around 1997 and Java RMI supports the development of distributed applications using Java language [36]. Java RMI offers developers a way to implement distributed Java applications. For example, a Java application on a computer can invoke the methods of other remote Java objects on different computers [22], [36]. As Java is platform-independent, Java RMI-based applications can be executed on many different computing platforms [36]. However, RMI heavily relies on Java language and does not work easily with other types of programming languages such as C++ [36].

5) Message Oriented Middleware (MOM): MOM relies on messages to enable communication between separate systems [6]. Using the message passing and queueing mechanism, MOM is able to carry information and action requests between distributed applications [6], [25]. MOM provides strong support for asynchronous communications [6]. Main disadvantages of MOM include limited support mechanisms for scalability, lack of industrial standards, and poor portability [26]. MOM has been used successfully in some industrial systems such as integrated manufacturing systems [20], [25].

D. Presentation Layer Integration

The integration in the presentation layer mainly focuses on the user interface (UI) integration [27]. Presenting integrated and consistent view for users is the main goal of UI integration [37]. So far, the research community has not done enough work at the presentation level [27]. A recent example of user interface integration is Web mashup such as integrating Google Map with other applications [37]. Portlet is another UI integration technology and has the technical potential to help produce more customizable and flexible portal applications [37]. Further work on effective standardization and instructional design at the presentation level is needed for effective user interface integration to take place [27].

IV. Recent Research on the Technology Integration of Distributed Enterprise Applications

In this section, we give a brief introduction to recent research work on the integration of distributed enterprise applications technologies. Both perspectives from developers and users are examined respectively. From the developers' perspective, enabling technologies for system developments including J2EE, .Net framework, and other web-based technologies are discussed; from the users' perspective, integration-related studies based on grid computing and cloud computing are introduced.

A. Enabling Technologies for System Development

1) Java 2 Enterprise Edition (J2EE): J2EE has emerged as a leading platform for developing enterprise applications [38]. J2EE contains a number of technologies such as "Enterprise Java Beans, RMI, Java Server Pages (JSP), and XML" [38]. Particularly, the Enterprise JavaBeans (EJB) provides a simplified method to develop component-based distributed applications over heterogeneous environments [22], [25]. J2EE has been used extensively in industrial systems. For example, J2EE was used as a system framework to integrate supply chain alliance enterprises' information systems [39].

2) Microsoft's .Net Framework: The .Net Framework provides support for objects developed by different Microsoft programming technologies to communicate with each other [34]. It allows programmers to create programs in different languages and allows their execution on different runtime systems and environments [34]. The .NET framework also provides a way for an object on one computer to call objects on a different computer using remoting infrastructure [40]. As .Net runs only on Windows systems, .Net technology has been rarely used in heterogeneous environments [34], [40].

3) Web Services, Service-Oriented Architecture (SOA) and Enterprise Service Bus (ESB): As traditional middleware such as CORBA and DCOM are typically used for Intranet applications and often has trouble to cross firewall boundaries [22], [42], web services have been created to support the integration of Internet applications in recent years [42]. A Web service usually contains various functions that can be organized as a single entity and can be published to the Web as a new service [41], [42]. Web services use the HTTP protocol transport information and thus can easily pass requests through firewalls [22], [41], [42].

Web services are building blocks for constructing Web-based distributed applications and they are being considered by many researchers as a suitable middleware for Internet-related application integration [41], [42]. Both J2EE and .Net can be used as programming tools by developers to create web services. Web services play an important role for integrating different middleware systems [41], [42]. As different middleware has different advantages, many enterprises have used various middleware over the years. However, middleware technologies and products from different vendors often have trouble to easily interoperate [43]. As a result, enterprises have to find other ways to integrate these different middleware systems. In some cases, ad-hoc techniques such as adapters were used or developed to support the integration needed by enterprises [2], [43]. But oftentimes, the integration is hard due to its complexity, lack of technical expertise and costs [2]. Web Services and its underlying principle named Service-oriented architecture are considered good solutions for such middleware-to-middleware interaction [2], [43].

SOA is a recent trend in integrating heterogeneous systems and different middleware systems [43]. SOA provides detailed guidelines and specifications to explain how services can be described, discovered and/or used [8], [22]. In SOA, developers organize and package different software applications as services. Each service includes an interface which specifies the operations available and types of messages it can handle [8], [43]. In industrial systems, SOA has been widely adopted and applied to transportation systems, manufacturing execution system, healthcare medical system, electronic power application system, electronics production systems, factory ordering systems, etc., [8], [20], [44]–[46].

Service-oriented integration depends on Enterprise Service Bus (ESB). ESB is able to work across different middleware products and standards to implement enterprise-wide SOA [2], [8]. ESB can shield from different protocols (e.g., CORBA IIOP, J2EE RMI) [2] to implement the smooth data flow and exchange among different application systems [47]. Thus, SOA-based ESB is often viewed as a new middleware technology by many researchers and practitioners. An example is that an electronic power application system was integrated by using ESB in Northern China [47]. Fig. 2 illustrates an SOA-oriented integration environment using ESB. In summary, Web services, SOA and ESB provide a promising and valuable framework for inter-enterprise integration.

B. Users' Integrated Solutions

The selection of a solution to integrate specific distributed enterprise applications depends on many factors such as the variety of resources and the complexity of businesses. Adequate computing solutions should be considered to support the system-level integration of distributed applications. In this section, we briefly discuss two key integration solutions from the users' perspective: grid computing and cloud computing.

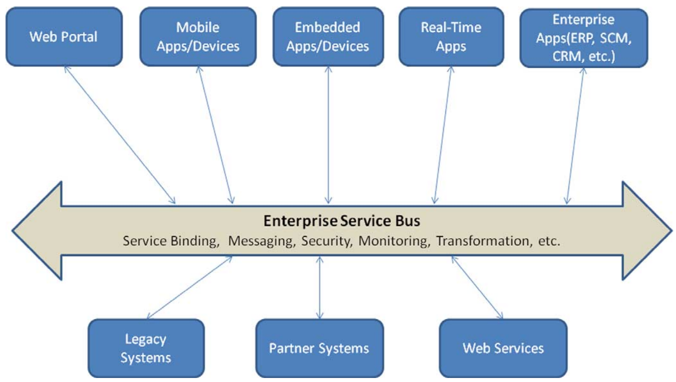

Fig. 2. An SOA-oriented integration environment using ESB.

1) Grid Computing: Grid computing has been adopted to solve some complex integration problems. For example, a .Net-based enterprise grid computing system was developed to construct enterprise desktop grids and applications for users [48]. Key issues related to grid computing are heterogeneity, reliability, reconfiguration, scheduling, and security. For example, Goel and Sobolewski [49] discussed the security issue of a grid system and developed a trust model for the system evaluation. Currently, the tested grid systems or applications are still limited to the low-level event responses and strong couplings are not adequately involved in the integration of distributed components.

2) Cloud Computing: Cloud computing has been used to support the integration of distributed applications recently. An e-HUB was developed as a SaaS platform to support the integration of distributed enterprise [50]. Enterprise applications can be plugged into the platform as needed. It is advantageous if the e-HUB evolves into a new platform where e-services can be easily deployed and integrated [50]. Delic and Riley [51] argued the enterprise knowledge architecture has evolved from enterprise stack to grid computing and will evolve to an enterprise knowledge cloud in a few years. Despite recent advances in grid computing and cloud computing, so far the achievements in these areas have been fragmented. Few enterprises have successfully applied grid computing or cloud computing to support high-level enterprise operation and application integration.

V. Future Perspectives

A. Trends

Integrating of various industry applications are an ongoing task for industry enterprises that are adopting new technologies and embedded devices. Some new trends in this area include the following.

1) As web services, SOA and ESA are being increasingly used to integrate existing and legacy applications, it is becoming increasingly important to ensure Quality-of-Service (QoS) for effective integration [52]. As different web service applications often have different QoS requirements and may cause conflicts between each other for resources such as bandwidth and processing time, it is necessary to develop empirically tested QoS integration models to check the QoS parameters, examine the defined service levels and quality, identify issues, and implement mechanism for dynamically selecting QoS-aware web services [53]–[55]. We expect to see models like trusted service cooperation and integration model [49]–[51] to be deployed in industrial enterprises.
2) The amount of data in industrial applications increases exponentially. There is an increasing need to integrate Online Analytical Processing (OLAP), knowledge discovery, data mining functions and data sources for decision support, information integration, and other business needs [56]–[58]. Application-specific middleware such as data mining middleware [56] will be increasingly developed and deployed for industrial information system integration.
3) The semantic web technology and the popular social networking technology are still in their infancy in terms of their use in industrial applications [59], [60]. The integration of semantic web technology and the social networking technology with sensor data is expected to grow in industrial applications [61] and to add more values to customers and partners [59] in industrial settings. Various ontology approaches for semantic integration and interoperability [22] will become mature and more applicable in industrial environments.
4) Mobile applications, embedded systems and smart embedded devices have been increasingly deployed in industry enterprises [61], [62]. This creates new opportunities and challenges to build communication and interoperability between industrial systems and embedded systems or devices [62]. Due to its strong support for both autonomy and interoperability, SOA approaches have the capabilities to implement communication and data exchange between embedded systems/devices and various applications [62]. We expect to see that industry applications are increasingly integrated with services running on networked, resource-limited mobile devices and smart embedded devices using SOA approaches. On the other hand, techniques such as embedded Web servers [63], gateway implementation and XML [64], [65] and TCP/IP-based protocol application programming [63], [64] will continue to be used to exchange data with a variety of heterogeneous Networked Control Systems (NCS) and commercial Programmable Logic Controllers (PLC) deployed in the industry over the years [63].

B. Some Research Challenges

1) As more and more industrial enterprise are adopting Web services to integrate various applications and devices, security challenges become more prominent. For example, security risk may arise in various stages of the industrial process such as the diagnosis, troubleshooting, and maintenance stages [3] because information exchange and device access have to go through multiple networks now.
2) Many industrial enterprises have "distributed real-time control systems (DRCS)" which interact with sensors and actuators over the network [65]. Generally, DRCS needs to be more reliable, robust, and efficient than data-centric applications [65]. Particularly, industrial enterprises such as power plants have strict performance constraints and response time requirements on their industrial systems and applications [65], [66]. For example, many industrial

automation and control systems collect data from heterogeneous sensors and require real-time analyses for the data collected from a variety of heterogeneous sensors [66]–[69]. As DRCS is playing an increasingly crucial role in critical industrial operations and transactions, new integration techniques such as real-time distribution middleware, distributed real-time Java [6] and real-time SOA solutions [66], [67] are needed to address performance concerns, and mitigate integration risks such as data quality issues, accountability, and security issues [70].

3) In practice, the integration of enterprise applications and devices may collapse due to various unanticipated changes [70]. Thus, research on the integration reliability including data and information reliability is highly valued by all industrial enterprises [66]. New guidelines and methods such as task load balance, fault-tolerant and message scheduling and transaction mechanisms needed to be further developed to ensure the reliability, robustness, maintainability, and rapid diagnosing of integrated industrial applications and devices in various environments.

4) As new technologies and devices are constantly introduced into industrial systems, user interface integration for industrial systems poses many new problems [27] like various interface types, standards, definitions, and service interfaces (functional, discovery, binding, etc.) [71]. Interface integration requires a good understanding of various applications, devices and enterprise-wide integration requirements. Currently, there is a lack of conceptual modeling techniques that can effectively help analysts comprehend enterprise-wide integration requirements [72].

5) In recent years, new technologies such as wireless sensor networks (WSNs), Internet of Things (IoT) and Radio Frequency Identification (RFID) have been deployed to industrial systems such as logistics systems, material flow systems, and supply chain management systems [73]. These new technologies made the integration of industrial applications, devices and various interfaces more difficult. There is a lack of services, guidelines, quality assurance, and standard architecture for allowing interactions of heterogeneous devices, sensors, aggregators, actuators, and diverse applications while increasing reusability and reducing various privacy and security concerns and issues [74]. For example, researchers identified a lack of services which can effectively connect users to appropriate sensor networks as the amount of data sources in sensor network increases [75]. Deployment of built-in user-related services on physical devices still need further research work to ensure the expected functionality and performance [76]. Various middleware solutions and architectures (e.g., publisher and subscriber architecture, message bus architecture, ontology architecture) for integrating industrial applications, RFID, WSN, and IoT have been proposed [22], [62], [66], [75]–[83]. However, these solutions and architectures are typically designed for respective domains. The scalability and customization of these middleware solutions and architectures are still challenging issues [84].

VI. Conclusion

EAI is concerned with the interaction and communication of various applications on heterogeneous environments [22]. So far, many techniques have been developed to integrate distributed enterprise applications. In this paper, we have surveyed the current status of the integration of distributed enterprise applications and discussed the research directions. As enterprises continue to add and/or deploy more applications, integrating distributed enterprise applications becomes inevitable for enterprises that need to achieve competitive advantages. As industry enterprises are increasingly adopting multitier client/server, Internet and service-oriented architectures for their enterprise applications and industrial devices [20], [35], [85], we have to address the need and requirement for interoperability in the industrial enterprise environment [21]. Currently, there are still many research challenges such as user interface integration, reliability, performance management and security risk management [3], data mining for distributed applications [86], [87], architecture design and optimization for wireless, RFID, and real-time sensor network [75], [88], and control architecture for the sensor, actuator and control service implementation [71]. These challenges will have to be effectively addressed or overcome in order for industrial systems to become more effective [8].

References

[1] V. S. Tanenbaum, *Distributed Systems, Principles and Paradigms*. Englewood Cliffs, NJ: Prentice Hall, 2002.

[2] S. Vinoski, "Integration with web services," *IEEE Internet Comput.*, vol. 7, no. 6, pp. 75–77, Nov.-Dec. 2003.

[3] Y. Xu, R. Song, L. Korba, L. Wang, W. Shen, and S. Lang, "Distributed device networks with security constraints," *IEEE Trans. Ind. Informat.*, vol. 1, no. 4, pp. 217–225, Nov. 2005.

[4] "Function blocks for industrial-process measurement and control systems," 2000, IEC TC65/WG6, IEC-TC65/WG6 Committee.

[5] D. S. Linthicum, *Enterprise Application Integration*. Essex, U.K.: Addison-Wesley Longman Ltd., 2000.

[6] P. B. Val, M. Garcia-Valls, and I. Estevez-Ayres, "Simple asynchronous remote invocations for distributed real-time Java," *IEEE Trans. Ind. Informat.*, vol. 5, no. 3, pp. 289–298, Aug. 2009.

[7] K. Qureshi, "Enterprise application integration," in *Proc. IEEE 2005 Int. Conf. Emerging Technologies*, Islamabad, Pakistan, Sep. 17–18, 2005, pp. 340–345.

[8] L. Xu, "Enterprise systems: State-of-the-art and future trends," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 630–640, Nov. 2011.

[9] Z. D. Zhou, R. Valerdi, and S. M. Zhou, "Guest editorial," *IEEE Trans. Ind. Informat.*, vol. 8, Special Section on Enterprise Systems, no. 3, p. 630, Aug. 2012.

[10] W. Viriyasitavat, L. Xu, and A. Martin, "SWSpec: The requirements specification language in service workflow environments," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 631–638, Aug. 2012.

[11] L. Xu, W. Viriyasitavat, P. Ruchikachorn, and A. Martin, "Using propositional logic for requirements verification of service workflow," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 639–646, 2012.

[12] J. Guo, L. Xu, G. Xiao, and Z. Gong, "Improving multilingual semantic interoperation in cross-organizational enterprise systems through concept disambiguation," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 647–658, Aug. 2012.

[13] Y. H. Yin, J. Y. Xie, L. Xu, and H. Chen, "Imaginal thinking-based human-machine design methodology for the configuration of reconfigurable machine tools," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 659–668, Aug. 2012.

[14] L. Xu, C. Wang, Z. Bi, and J. Yu, "AutoAssem: An automated assembly planning system for complex products," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 669–678, Aug. 2012.

[15] L. Li, R. Ge, S. M. Zhou, and R. Valerdi, "Guest editorial integrated healthcare information systems," *IEEE Trans. Inform. Technol. Biomed.*, vol. 16, no. 4, pp. 515–517, 2012.
[16] Y. H. Yin, Y. J. Fan, and L. Xu, "EMG and EPP-integrated human-machine interface between the paralyzed and rehabilitation exoskeleton," *IEEE Trans. Inform. Technol. Biomed.*, vol. 16, no. 4, pp. 542–549, 2012.
[17] "Roadmap for board assembly," in *NEMI Technology Roadmaps 1996 Edition*. Herndon, VA, USA: INEMI, Dec. 1996, .
[18] S. Chen, "Open design of networked power quality monitoring systems," *IEEE Trans. Instrument. Meas.*, vol. 53, no. 2, pp. 597–601, Apr. 2004.
[19] A. Lüder, A. Klostermeyer, J. Peschke, A. Bratoukhine, and T. Sauter, "Distributed automation: PABADIS versus HMS," *IEEE Trans. Ind. Informat.*, vol. 1, no. 1, pp. 31–38, Feb. 2005.
[20] I. M. Delamer and J. L. M. Lastra, "Service-oriented architecture for distributed publish/subscribe middleware in electronics production," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 281–294, Nov. 2006.
[21] A. Kalogeras, J. Gialelis, C. Alexakos, M. Georgoudakis, and S. Koubias, "Vertical integration of enterprise industrial systems utilizing web services," *IEEE Trans. Ind.Informat.*, vol. 2, no. 2, pp. 120–128, May 2006.
[22] S. Izza, "Integration of industrial information systems: From syntactic to semantic integration approaches," *Enterprise Inform. Syst.*, vol. 3, no. 1, pp. 1–57, 2009.
[23] V. Matena, S. Krishnan, L. DeMichiel, and B. Stearns, *Applying Enterprise JavaBeans™: Component-Based Development for the J2EE™ Platform*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2003.
[24] A. Sinha, "Client-server computing," *Communications of the ACM*, vol. 35, no. 7, pp. 77–98, Jul. 1992.
[25] B. Benatallah and H. R. Motahari-Nezhad, "Service oriented architecture: Overview and directions," *Advances in Software Engineering* A. Ferro and E. Boerger, Eds., vol. 5316/2008, Lecture Notes in Computer Science, pp. 116–130, 2008.
[26] D. Chen, G. Doumeingtsb, and F. Vernadatc, "Architectures for enterprise integration and interoperability: Past, present and future," *Comput. Ind.*, vol. 59, no. 7, pp. 647–659, 2008.
[27] F. Daniel, J. Yu, B. Benatallah, F. Casati, M. Matera, and R. Saint-Paul, "Understanding UI integration: A survey of problems, technologies, and opportunities," *IEEE Internet Computi.*, vol. 11, no. 3, pp. 59–66, May-Jun. 2007.
[28] P. Bernstein, "Middleware: A model for distributed systems services," *Commun. ACM*, pp. 86–98, 1996.
[29] S. L. Ooi and M. T. Su, "Integrating enterprise application using message-oriented middleware and J2EE technologies," in *Proc. Int. Conf. Comput. Informat.*, Jun. 2006, pp. 1–5.
[30] W. Emerich, "Software engineering and middleware: A roadmap," in *Proc. Conf. Future of Software Eng.*, Limerick, Ireland, 2000, pp. 117–129.
[31] Q. Chen, J. Yao, and R. Xing, "Middleware components for e-commerce infrastructure: An analytical review," *J. Issues in Inform. Sci. Inform. Technol.*, vol. 3, pp. 137–146, 2006.
[32] "Open Software Foundation," in *OSF DCE Application Development Guide*, 1.0.2 ed. Cambridge, MA: Open Software Foundation, 1994, Revision 1.0.
[33] Recommendations for Using DCE, 1998., DCOM, and CORBA Middleware, Sponsored by: DISA/JIEO Center for Computer Systems Engineering (JEXF).
[34] Microsoft Corporation, "DCOM technical overview," 2011. [Online]. Available: http://technet.microsoft.com/en-us/library/cc722925.aspx, Available:
[35] S. Vinoski, "CORBA: Integrating diverse applications within distributed heterogeneous environments," *IEEE Commun. Mag.*, vol. 35, no. 2, pp. 46–55, Feb. 1997.
[36] J. Maassen, R. van Nieuwpoort, R. Veldema, H. E. Bal, T. Kielmann, C. Jacobs, and R. Hofman, "Efficient Java RMI for parallel programming," *ACM Trans. Prog. Lang. Syst.*, vol. 23, no. 6, 2001.
[37] F. Bellas, "Standards for second-generation portals," *IEEE Internet Comput.*, vol. 8, no. 2, pp. 54–60, Mar.-Apr. 2004.
[38] R. Johnson, "J2EE development frameworks," *Computer*, vol. 38, no. 1, pp. 107–110, Jan. 2005.
[39] H. Zhang, B. Zhang, and B. Liu, "Information integration solutions based on J2EE for supply chain enterprises," in *Proc. Int. Conf. Manage. Serv. Sci. (MASS)*, Wuhan, China, 2011, pp. 1–4.
[40] S. Khan, K. Qureshi, and H. Rashid, "Performance comparison of ICE, HORB, CORBA and dot NET remoting middleware technologies," *Int. J. Comput. Appl.*, vol. 3, no. 11, pp. 15–18, 2010.
[41] J. Roy and A. Ramanujan, "Understanding web services," *IT Profess.*, vol. 3, no. 6, pp. 69–73, 2001.
[42] M. D. Hanes, S. C. Ahalt, and A. K. Krishnamurthy, "A comparison of Java RMI, CORBA, and web services technologies for distributed SIP applications," in *High Performance Embedded Computing Annual Workshop (HPEC)*, 2002.
[43] S. Baker and S. Dobson, "Comparing service-oriented and distributed object architectures," in *Proc. Int. Symp. Distrib. Obj. Appl.*, 2005, pp. 631–645, LNCS 3760: p.
[44] Y. H. Yin and J. Y. Xie, "Reconfigurable manufacturing execution system for pipe cutting," *Enterprise Inform. Syst.*, vol. 5, no. 3, pp. 287–299, 2011.
[45] L. Duan, W. N. Street, and E. Xu, "Healthcare information systems: Data mining methods in the creation of a clinical recommender system," *Enterprise Inform. Syst.*, vol. 5, no. 2, pp. 169–181, 2011.
[46] M. Dumitrache, S. Dumitra, and M. Baciu, "Web services integration with distributed applications," *J. Appl. Quantitative Methods*, vol. 5, no. 2, pp. 223–233, 2010.
[47] R. Xu, J. Bai, and Y. Wang, "The research and implementation of power application system integration based on enterprise service bus," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst. (ICIS)*, Beijing, China, 2010, pp. 466–469.
[48] A. Luther, R. Buyya, R. Ranjan, and S. Venugopal, "Alchemi: A .NET-based enterprise grid computing system," in *Proc. 6th Int. Conf. Internet Comput. (ICOMP'05)*, Las Vegas, NV, USA, June 27–30, 2005.
[49] S. Goel and M. Sobolewski, "Trust and security in enterprise grid computing environment," in *Proc. IASTED Int. Conf. Commun., Netw., Inform. Security*, New York, NY, USA, Dec. 10–12, 2003.
[50] D. Concha, J. Espadas, D. Romero, and A. Molina, "The e-hub evolution: From a custom software architecture to a software-as-a-service implementation," *Comput. Industry*, vol. 61, no. 2, pp. 145–151, 2010.
[51] K. A. Delic and J. A. Riley, "Enterprise knowledge clouds: Next generation KM systems?," in *Proc. Int. Conf. Inform., Process, Knowl. Manage.*, 2009, pp. 49–53.
[52] B. S. Farroha and D. L. Farroha, "Policy-based QoS requirements in a SOA enterprise framework—An investigative analysis," in *Proc. IEEE Military Commun. Conf.*, 2007, pp. 1–7.
[53] J. Ai, J. , J. Gao, J. Yu, and Z. Zhao, "A concept for QoS integration in trusted and autonomic service cooperation," in *Proc. 1st Int. Conf. Inform. Sci. Eng. (ICISE2009)*, 2009, pp. 3922–3925.
[54] X. Wang and X. Xu, "DIMP: An interoperable solution for software integration and product data exchange," *Enterprise Inform. Syst.*, vol. 6, no. 3, pp. 291–314, 2012.
[55] R. Mietzner, F. Leymann, and T. Unger, "Horizontal and vertical combination of multi-tenancy patterns in service-oriented applications," *Enterprise Inform. Syst.*, vol. 5, no. 1, pp. 59–77, 2011.
[56] V. T. Ravi and G. Agrawal, "Integrating and optimizing transactional memory in a data mining middleware," in *Proc. Int. Conf. High Perform. Comput. (HiPC)*, 2009, pp. 215–224.
[57] B. Liu, S. G. Cao, and W. He, "Distributed data mining for E-business," *Inform. Technol. Manage.*, vol. 12, no. 1, pp. 1–13, 2011.
[58] E. Xu, M. Wermus, and D. B. Bauman, "Development of an integrated medical supply information system," *Enterprise Inform. Syst.*, vol. 5, no. 3, pp. 385–399, 2011.
[59] J. G. Breslin, D. O'Sullivan, A. Passant, and L. Vasiliu, "Semantic web computing in industry," *Comput. Ind.*, vol. 61, no. 8, pp. 729–741, 2010.
[60] G. Governatori and R. Iannella, "A modeling and reasoning framework for social networks policies," *Enterprise Inform. Syst.*, vol. 5, no. 1, pp. 145–167, 2011.
[61] M. Kaplan and M. Haenlein, "Users of the world, unite! The challenges and opportunities of social media," *Business Horizons*, vol. 53, pp. 59–68, 2010, S..
[62] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Trans. Services Comput.*, vol. 3, no. 3, pp. 223–235, Jul.-Sep. 2010.
[63] A. Jestratjew and A. Kwiecien, "Performance of HTTP protocol in networked control systems," *IEEE Trans. Ind. Informat.*, 2012, DOI: 10.1109/TII.2012.2183138.
[64] T. Sauter and M. Lobashov, "How to access factory floor information using internet technologies and gateways," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 699–712, Nov. 2011.
[65] S. Eberle, "Adaptive internet integration of field bus systems," *IEEE Trans. Ind. Informat.*, vol. 3, no. 1, pp. 12–20, Feb. 2007.

[66] L. Du, C. Duan, S. Liu, and W. He, "Research on service bus for distributed real-time control systems," in *Proc. 6th IEEE Joint Int. Inform. Technol. Artif. Intell. Conf.*, 2011, pp. 401–405.

[67] T. Cucinotta, A. Mancina, G. F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo, and F. Rusinà, "A real-time service-oriented architecture for industrial automation," *IEEE Trans. Ind. Informat.*, vol. 5, no. 3, pp. 267–277, Aug. 2009.

[68] T. Cucinotta, L. Palopoli, L. Abeni, D. Faggioli, and G. Lipari, "On the integration of application level and resource level QoS control for real-time applications," *IEEE Trans. Ind. Informat.*, vol. 6, no. 4, pp. 479–491, Nov. 2010.

[69] X. Liu, Q. Wang, S. Gopalakrishnan, W. He, L. Sha, H. Ding, and K. Lee, "ORTEGA: An efficient and flexible online fault tolerance architecture for real-time control systems," *IEEE Trans. Ind. Informat.*, vol. 4, no. 4, pp. 213–224, Nov. 2008.

[70] R. Gleghorn, "Enterprise application integration: A manager's perspective," *IT Professional*, vol. 7, no. 6, pp. 17–23, 2005.

[71] A. Pohl, H. Krumm, F. Holland, I. Luck, and F.-J. Stewing, "Service-orientation and flexible service binding in distributed automation and control systems," in *Proc. 22nd Adv. Inform. Netw. Appl. Workshops*, 2008, pp. 1393–1398.

[72] N. Bolloju, "Conceptual modeling of systems integration requirements," *IEEE Software*, vol. 26, no. 5, pp. 66–74, Sep.-Oct. 2009.

[73] S. Kumar, B. Kadow, and M. Lamkin, "Challenges with the introduction of radio-frequency identification systems into a manufacturer's supply chain—A pilot study," *Enterprise Inform. Syst.*, vol. 5, no. 2, pp. 235–253, 2011.

[74] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "Role of middleware for internet of things: A study," *Int. J. Comput. Sci. Eng. Survey (IJCSES)*, vol. 2, no. 3, pp. 94–105, 2011.

[75] C. Gadea, B. Ionescu, and D. Ionescu, "Real-time collaborative intelligent services for sensor networks," in *Proc. ICCC-CONTI 2010*, May 2010, pp. 511–516.

[76] G. Candido, A. W. Colombo, J. Barata, and F. Jammes, "Service-oriented infrastructure to support the deployment of evolvable production systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 759–767, Nov. 2011.

[77] F. Gandino, B. Montrucchio, M. Rebaudengo, and E. R. Sanchez, "On improving automation by integrating RFID in the traceability management of the agri-food sector," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2357–2365, Jul. 2009.

[78] Y. Tian, J. V. Geiger, H. B. Su, S. V. Kumar, and P. R. Houser, "Middleware-based sensor web integration," *IEEE J. Sel. Topics Appl. Earth Observations and Remote Sensing*, vol. 3, no. 4, pp. 467–472, Dec. 2010.

[79] S. Haller, S. Kanouskos, and C. Schroth, "The Internet of Things in an enterprise context," in *Future Internet Systems (FIS)*. New York, NY, USA: Springer, 2009, vol. 5468, pp. 14–28, LCNS.

[80] C. Lee and C. Chung, "RFID data processing in supply chain management using a path encoding scheme," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 5, pp. 742–758, May 2011.

[81] M. Kranz, P. Holleis, and A. Schmidt, "Embedded interaction interacting with the Internet of things," *IEEE Internet Comput.*, pp. 46–53, Mar./Apr. 2010.

[82] T. Perumal, A. R. Ramli, C. Y. Leong, K. Samsudin, and S. Mansor, "Middleware for heterogeneous subsystems interoperability in intelligent buildings," *Autom. Construction*, vol. 19, no. 2, pp. 160–168, 2010.

[83] L. Xu, "Information architecture for supply chain quality management," *Int. J. Prod. Res.*, vol. 49, no. 1, pp. 183–198, Jan. 2011.

[84] H. Panetto and A. Molina, "Enterprise integration and interoperability in manufacturing systems: Trends and issues," *Comput. Ind.*, vol. 59, no. 7, pp. 641–646, 2008.

[85] G. Cândido, A. W. Colombo, J. Barata, and F. Jammes, "Service-oriented infrastructure to support the deployment of evolvable production systems," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 759–767, Nov. 2011.

[86] D. M. Chiang, C. Lin, and M. Chen, "The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres," *Enterprise Inform. Syst.*, vol. 5, no. 2, pp. 219–234, 2011.

[87] L. Duan and L. Xu, "Business intelligence for enterprise systems: A survey," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 679–687, Aug. 2012.

[88] S. Li, L. Xu, X. Wang, and J. Wang, "Integration of hybrid wireless networks in cloud services oriented enterprise information systems," *Enterprise Inform. Syst.*, vol. 6, no. 2, pp. 165–187, 2012.

Wu He received the B.S. degree in computer science from DongHua University, Shanghai, China, in 1998, and the Ph.D. degree in information science from the University of Missouri, Columbia, MO, USA, in 2006.

His research interests include enterprise applications, data mining, and knowledge management.

Li Da Xu (M'86–SM'11) received the M.S. degree in information science and engineering from the University of Science and Technology of China, Hefei, Anhui Province, in 1981, and the Ph.D. degree in systems science and engineering from Portland State University, Portland, OR, USA, in 1986.

He serves as the Founding Chair of IFIP TC8 WG8.9 and the Founding Chair of the IEEE SMC Society Technical Committee on Enterprise Information Systems.