

## A Review of Open Systems Architecture

**Authors:** Liping Gu

**Date:** 2016-06-08T00:00:00+00:00

### Abstract

This paper briefly introduces the requirements and advantages of open system architecture, analyzes its foundational elements in hardware, software, and network services, enumerates case studies in the defense, aviation, automotive, and healthcare industries, describes its role in library and information institutions, and summarizes the design principles for implementing open system architecture.

### Full Text

#### Review of Open System Architecture Research

**Author Affiliation:** National Science Library, Chinese Academy of Sciences, Beijing 100190, China

### Abstract

This article briefly introduces the needs and benefits of Open System Architecture (OSA), analyzes its foundations in hardware, software, and network services, enumerates case studies in military, aerospace, automotive, and medical industries, describes its functions in libraries and information institutions, and summarizes the design principles for implementing OSA.

**Keywords:** Component agent; Information and communication technology; Operating system; Open architecture; Open interface

### 1. Needs and Advantages of Open System Architecture

In the information society, network services that provide dynamic information represent the development trend of information service systems. Open system architecture offers multiple advantages for activities involving information, resources, and systems: it supports system analysis based on component methods,

facilitates system implementation using component-based approaches, helps describe system complexity, and supports system design [1]. From the perspective of enterprise information flow management, current network service systems allow businesses to link their business processes with technical processes through minimal changes to business interfaces, without altering the underlying operating system. For example, a clothing retail organization with 500 international chain stores frequently needs to update designs to keep pace with fashion trends, which may involve changing not only styles and colors but also fabrics, manufacturers, and deliverable products [2]. If the systems between retailers and manufacturers are incompatible, switching from one supplier to another becomes an extremely complex process. Practical open system architecture and product standards significantly reduce development time from standards to products. When systems define task module models with consideration for interoperability and interchangeability, subsequent system performance can achieve real-time capabilities, usability, and practicality advantages [3].

End users often encounter situations where completing a single task requires repeatedly searching through multiple databases or websites. With the development of computer hardware and software, many industries face system incompatibility issues that prevent information sharing between systems, resulting in substantial duplicated efforts. Open system architecture and universal industry standards are needed to resolve this problem of redundant resource construction [4]. For instance, travelers can purchase bus and tram tickets across Germany, book hotels and sightseeing tickets, and even reserve travel routes in other EU countries through the German railway's ticketing system [5]. Such systems save users from having to query various transportation prices, seat availability, and schedules when traveling from place A to place B, while also providing information on hotels, restaurants, sightseeing locations (including opening hours, discounts, pricing, and restrictions) at origin, destination, and transfer points, along with weather, holidays, exchange rates, and credit limits. Therefore, services built on open system architecture not only facilitate the use of their own services but also enable users to access services from other institutions, generating positive benefits for entire industries.

In open system architecture, each architectural element or component can be designed as an independent unit. If the component agent approach is adopted as the basic system module, utilizing its software component technology to build systems, components become reusable application data and public methods that other applications can access and manipulate [6]. In mechanical engineering, open architecture includes interface specifications for necessary hardware and software integration of new processes and their drive control, monitoring, diagnostics, service functions, and sensors, significantly saving repeated modeling and design efforts [7]. In fact, fierce market competition demands rapid product development, and virtual product development represents a new development model. Applying this architecture can reduce costs and improve development efficiency during the popularization of communication architectures, hardware devices, and software content.

Open system architecture originated from industry demands in the electronic parts supply chain. Beyond hardware, software and network services have also adopted similar development patterns and architectural designs. Large electronic systems often require subsystems from different suppliers and application categories to be integrated into the same hardware. Partitioned operating systems (OS) allow shared access to critical system resources for application integration systems, saving time and space needed for subsystem packaging [8]. Component-based open middleware architecture defines message passing and operation methods between software components at the application layer, aiming to ensure interoperability across multiple platforms and portability for future development technologies [9]. In the chemical industry, utilizing flexible and adaptive designs of open system architecture and predictive simulation tool development, along with the development trend of electronic maintenance (such as wireless communication, intelligent network services, and smart handheld computers), has shifted from pure technological advancement toward information standards for open system architecture in collection, communication, information storage, and identification system interoperability, gradually moving toward a “plug-and-play” basic operation mode under a global framework across different systems [11]. In this trend, the integrated environment of software development tools adopting open system architecture supports unified automation methods and professional functions, introducing a unified approach to embedded design processes for complex real-time control systems [12]. Provided distributed computing can be guaranteed, open systems can access and operate other systems through standard interfaces [13]. In the network services segment under hardware and software, open system architecture can promote semantic interoperability of semantic web technologies, using web service technology to build loosely coupled components and combining workflow technology to enrich required processing capabilities. An open architecture can effectively support key activities, from creating, retrieving, and post-processing to composing, classifying, and invoking algorithms, establishing new automated processing solutions for the knowledge lifecycle [14]. For a specific case, to fully fulfill the requirements of space tracking, prediction, and timely comprehensive understanding of polymer production processes [10], a geological feature network server (an open system architecture) with a three-tier architecture including client (GOCAD), application server (Apache Tomcat and Degree), and database (MySQL) can access three-dimensional geological models through web services to generate spatial applications of three-dimensional geographic models (compliant with ISO TC/211 - Geographic standards) [15].

The hierarchical layering of open system architecture across hardware, software, and networks, with open interfaces at each level, provides five advantages: easy partial component updates; partial component updates without affecting overall system operation; open interfaces meaning anyone can participate in overall system engineering, increasing industry competition and innovation; standardization of open interfaces promoting increasing market share of open systems within and across industries; and, based on the promotion effectiveness of open

system architecture in industries, obtaining intelligence on technical capabilities of other potential competitors through industry analysis.

From the perspective of computer hardware support, the Open Service Availability Framework (OpenSAF) combined with the Open Modeling and Simulation Architecture (OpenMSA) and Open System Architecture for Modeling and Simulation (OSAMS) features extensibility, parallelism, and distributed characteristics, supporting real-time business-oriented system applications, providing scalable high-performance execution of proposed cognitive architectures for multi-core computers [16]. From software to network services, semantic web and agent technologies represent the latest developments in Information and Communication Technology (ICT) for monitoring and maintenance in manufacturing, power, chemical, and other industries, which is a common development direction for multiple industries [17]. This industry diffusion trend is supported by mechanisms that transmit and collect data from different sources through multifunctional and open system architecture, filtering and processing through algorithms originally developed for real-time system monitoring, with customized open system architecture capable of handling data archiving and extended archiving, even managing visualization methods for this data [18].

Data flow, information flow, and knowledge flow management are separate issues. What is emphasized here is that the substantive significance of open system architecture is ensuring the possibility of free flow of data, information, and knowledge streams. This possibility varies across industries under different restrictive specifications and technical safeguard measures.

### 3. Industry Cases of Open System Architecture

Open system architecture relates to industry standards and specifications, but different patterns exist across domains: the military field follows government-established standards; the aviation industry follows standards set by upstream giants and downstream manufacturers; the automotive field involves a few industry oligarchs jointly establishing standards for self-implementation; and the healthcare field gradually forms industry standards based on individual practices. Additionally, different open system architectures warrant mutual reference.

#### 3.1 Open System Architecture in Military Systems

Experienced troubleshooters require training at component manufacturing factories to become more familiar with components. When a cannon breaks down, the solution is not to purchase, ship, or cast a new one, but to troubleshoot or replace parts. Familiarity with components is crucial because weapons no longer need to be sent to repair factories but can be maintained, repaired, and restarted on-site. This is especially true for large weapons, particularly ships and carrier-based aircraft.

The emergence of military standards resulted from parts replacement needs in

the mechanical age, which also serve as demonstration standards for commercial and public industries. With standard specifications, component production is not limited to fixed factories and production teams. Of course, production capability depends on manufacturing processes (precision) and licenses (franchises), but this approach is highly efficient in industrial engineering and administrative management. In the electronic tactical era, this demand not only expands to various industries related to military industry but also gradually becomes an important business philosophy and industrial law.

The U.S. Air Force began funding the Small Business Innovation Research (SBIR) program in 1998 to integrate small businesses into the military-industrial system. Its technology platform is based on the Condition-based Intelligent Equipment Management System (ICEMS), which includes an open system architecture applicable to all types of equipment or systems (such as engines) and a set of tools for heterogeneous data integration in distributed systems, processes, and asset management [21]. For example, the software requirements for wireless technology on naval carrier-based aircraft involve programmable, multi-mode, multi-band, networked, robust, mobile, resilient systems with synchronized voice, data, and video information assurance; the U.S. Navy's Space and Naval Warfare Systems Command (SPAWAR) once commissioned Motorola to design a radio digital modular platform, a plan that included detailed evaluation, analysis, and demonstration of key attributes such as reconfigurability, architectural openness, software programmability, modularity, portability, and scalability [22]. Subsequently, the U.S. Navy required systems to adopt open system specifications and standards, implementing proper design and development schemes with sufficient open interfaces, services, and support format specifications [22].

Open system architecture adopted by U.S. naval vessels includes relevant models and standard interfaces, enabling the latest commercial equipment and related technologies such as structures, electrical systems, mechanical systems, and other shipboard systems to be upgradable. By developing an open system architecture and standard interface framework, ship vitality and combat effectiveness can be extended and enhanced [23]. Ship systems require component replacement, upgrading, enhancement, and reuse, necessitating an open system architecture to create a "plug-and-play" environment including industry standard interfaces, protocols, and network variables [24].

### 3.2 Open System Architecture in Aviation Industry

Avionics development or upgrading is often extremely expensive. Generating new product descendants from the same product family can save time and money. Currently, U.S.-developed radar systems consist of a layered open system architecture from underlying hardware and software to partitioned combat systems, middleware, communication fabric, and computer platform applications. This framework includes a component library for expanding application domains, making hardware independence relatively easy and enabling easy

maintenance and upgrading of infrastructure [19]. The U.S. Navy's Tomahawk missile weapon control system uses an open system architecture based on Fiber Distributed Data Interface (FDDI) and Computer Software Configuration Items (CSCIs) as its weapon control system interface (CCS) [20], thereby achieving weapon control and development.

The aviation industry represents the most complex large-scale systems engineering, including standards for complete aircraft, engines, and ground support infrastructure. Each subsystem includes countless switches and standard operating procedures, requiring multi-person synchronization even with advanced electronic communications. Taking Boeing's Integrated Vehicle Health Management (IVHM) architecture as an example, even providing asset management for ground facilities requires creating an open system architecture for condition-based maintenance (OSACBM) as a universal tool. Using extensible and efficient modules, it simplifies the IVHM integration software framework, primarily handling integration of software models, algorithms, data, communication, and embedded processors; integrating IVHM control usage, communication, mission, flight, maintenance, and other major vehicle systems [26,27]. Through extensive maintenance and usage management experience, the aviation industry's software development tool chain can support collaborative development of complex systems with multiple commercial partners, solving development and deployment challenges through alliance-based approaches.

### 3.3 Open System Architecture in Automotive Industry

Contrary to Boeing's proprietary open system architecture, the Automotive Open System Architecture (AUTOSAR) developed by the automotive manufacturing industry is an open, standardized software architecture jointly established by major global automakers. On one hand, it maintains stability among existing partnerships; on the other hand, it addresses development challenges for critical systems such as brake-by-wire or steer-by-wire [28] by establishing similar cooperation conventions to avoid unnecessary duplicate specification development.

In recent years, the increasing software complexity required by the automotive industry has demanded significant time and cost to manage and develop software modules in vehicles. A core aspect of AUTOSAR's standardized software architecture supporting distributed processing embedded platforms is implementing virtual bus functionality for Electronic Control Units (ECUs) (communication applications between software components) [29]. To reduce costs of developing new vehicle functions, the standardized software architecture promoted by the automotive industry plays an important role in reliability analysis processes, where fault injection schemes help developers address and solve implementation issues while enhancing environmental applicability [30]. AUTOSAR's standardized software architecture and software system scheme for automotive electronic control-defined vehicle diagnostic protocols can improve reusability and portability while reducing software development lifecycle [31].

### 3.4 Open System Architecture in Healthcare

Network, sensor, and embedded device technologies can assist in improving healthcare and reducing social costs of population aging. However, challenges include system usability, reliability, and safety. Open system architecture can combine software and hardware from many vendors to provide information needed for medical assistance [32]. However, due to different interfaces and transmission protocols, communication between devices from different manufacturers is often impossible. In medical fields, multiple automated devices for patient monitoring and treatment lack configuration and coordination. Therefore, if a universal data and communication model exists to describe transmission equipment and critical parameters, device types, communication specifications, etc., it can be compatible with different platforms and system devices while maintaining a framework for special plug-in interfaces, enabling simple plug-ins to have different uses on different devices [33].

On the other hand, wireless networks combined with image sensors to develop various remote sensing applications represent an emerging industry. As an open system architecture supporting wireless transmission image processing, it can support any sensor or wireless network application instance, accelerating development of complex and efficient algorithms required for practical applications [34]. This has pioneering significance not only in healthcare devices but also in remote education, spatial intelligence management (home, entertainment, public places), and remote control fields (such as seabed and space monitoring assistance).

## 4. Role of Open System Architecture in Libraries and Information Institutions

Currently, existing library automation systems, resource link servers, electronic resource management systems, and digital resource preservation and management systems have adopted open system architecture to some extent, using multi-tier client/server architecture where component modularity can be added, modified, and customized as needed. However, deep-level issues such as data migration and reconstruction, along with core code openness and development capabilities, are rarely discussed.

Real-time interactive virtual classrooms, which have not yet been considered and integrated, have three priorities for open system architecture: how to support integration of human-machine interfaces and personal mobile devices in classrooms; how to handle deployment of interaction interfaces with other learning systems under different circumstances; and how to manage interactivity and transmission functions required for cross-cultural learning in different contexts and countries [35].

Furthermore, multi-agent patterns have been fully applied in process control, mobile robots, information retrieval, and other fields. Open system architec-

ture can provide network dynamic agent independence and serve as a universal information retrieval and management system architecture [36]. However, developing such systems requires automated testing, while developing and maintaining large amounts of test case software is both expensive and error-prone. On a standard interface of open system architecture, service data definitions, functions, and domain knowledge provided by semantic models can create rules to simulate causes and conditions of Software Under Test (SUT) [37], partially solving this engineering problem.

In addressing business processes, objects, and resources in libraries and information institutions, the Computer Integrated Manufacturing Open System Architecture (CIMOSA) modeling framework has not yet been adopted. Analyzing institutional environments through data flow diagrams (DFDs) or forming Unified Modeling Language cases [38], the question lies in how to practice and detect the so-called degree of openness.

## 5. Design Principles for Implementing Open System Architecture

An important advantage of open system architecture is development of complex products, as product development requires many software components that can be reused across platforms. Component-based open system architecture enables synergy of product component data and operation methods [42]. Once participants and designers propose challenges requiring both overall grasp and detailed clarification, open system architecture 梳理 helps recognize commonalities and differences between library and information systems and other industry systems. More fundamental issues exist in execution problems of data analysis, application, management, and preservation. The solution approach takes the library community's pioneering KB as a foundation for deeper description in the future.

The positive significance of open system architecture is achieving human-centered design. Middleware products on open platforms follow design principles that recognize product target users, develop user-friendly software products, and complete customer-centered design methods and finished products based on product usability, functionality, and operability [43].

The negative crisis of open system architecture lies in lack of strategic thinking and system monitoring and risk analysis. These factors are easily overlooked in architectural design: whether open system architecture projects guide developers to build an open system; whether there is capability to evaluate open system engineering; and whether system performance can be guaranteed [44]. Since open systems must allow reasonable interoperability, portability, and scalability, open system architecture actually requires management mechanisms as support; otherwise, it can easily be misunderstood, making component expansion and function enhancement become a bottomless pit of future cost and manpower additions. In other words, open system architecture is actually a constraint on the entire system.

Addressing this negative crisis requires optimizing integrated system framework design. As system response is an important part of integrated manufacturing processes, logistics, organization, and information subsystems, combining Quality Function Deployment (QFD), Value Engineering (VE), and Failure Mode and Effects Analysis (FMEA) facilitates control of product cost, quality, and reliability [45]. Even open system architecture requires certain management mechanisms as support.

In recent years, product lifecycles in knowledge-intensive industries have changed significantly. Facing this challenge and managing increasingly complex product development and service management are necessary to maintain continuous business growth [40]. Open system architecture can primarily maintain cooperation between virtual enterprises (VE), including information parallelism, project operating conditions, electronic document management, product structure and configuration management, access control, restriction marking, work groups, user analysis, intellectual property and version management, workflow management, distributed databases, application integration schemes, and application launch procedures [41]. Facing increasing embedded intelligent technologies and evolving user dynamic relationships, research-intensive libraries and information institutions need to provide more value-added services to parent institutions and partners.

## References

- [1] Chi Y L, Cai J G, Wu Z Y. Platform for an Open Architecture Controller Based on a General Operation System with a Hard Real-time Extension[J]. International Journal of Production Research, 2003, 41(12): 2839-2850.
- [2] Ruentex Garment Management[EB/OL]. [2010-08-16]. <http://www.ruentex.com.tw/garment/>.
- [3] Brueck T M, Gerstberger R L. Emerging Standards for Water Utility Computerization[J]. Journal of American Water Works Association, 1993, 85(4): 124-128.
- [4] Ong M H, West A A, Monfared R P, et al. Application of Enterprise Modeling Technique for Specifying a Component-based System[J]. Journal of Engineering Manufacture, 2005, 219(9): 649-664.
- [5] Startseite[EB/OL]. [2010-09-22]. <http://www.bahn.de/p/view/index.shtml>.
- [6] Rosenman M, Wang F J. A Component Agent Based Open CAD System for Collaborative Design[J]. Automation in Construction, 2001, 10(4): 383-397.
- [7] Pritschow G, Kramer C. Open System Architecture for Drives[J]. CIRP Annals-Manufacturing Technology, 2005, 54(1): 375-380.
- [8] Leiner B, Schlager M, Obermaisser R, et al. A Comparison of Partitioning Operating Systems for Integrated Systems[C]. In: Proceedings of the 26th International Conference on Computer Safety, Reliability and Security. 2007: 342-355.

- [9] Ahn M K, Park Y W, Jee T Y. Component Based Open Middleware Architecture for Autonomous Navigation System[C]. In: Proceedings of the SPIE. 2007: 656120.
- [10] Kralis A, Pladis P, Kanellopoulos V, et al. Design, Simulation and Optimization of Polymerization Processes Using Advanced Open Architecture Software Tools[C]. In: Proceedings of the 20th European Symposium on Computer Aided Process Engineering. 2010: 955-960.
- [11] Arnaiz A, Iung B, Adgar A, et al. Information and Communication Technologies Within E-maintenance[J]. E-Maintenance, 2010: 39-60.
- [12] Baxter M J, Bass J M, Browne A R, et al. Development Framework Approach to Heterogeneous System Design for Control Systems[J]. Control Engineering Practice, 1996, 4(2): 229-238.
- [13] Mukhopadhyay D, Dalezman B. Designing Open Systems with Case[J]. Information Systems Management, 1995, 12(1): 26-34.
- [14] Pitikakis M, Houstis C, Vasilakis G, et al. A Knowledge Management Architecture for 3D Shapes and Applications[C]. In: Proceedings of the 10th Panhellenic Conference on Advances in Informatics (PCI '05). Berlin, Heidelberg: Springer-Verlag, 2005: 360-370.
- [15] Poulio J, Badard T, Desgagne E, et al. Development of a Web Geological Feature Server (WGFS) for Sharing and Querying of 3D Objects[C]. In: Proceedings of the 2nd International Workshop on 3D Geo-Information. 2008: 115-130.
- [16] Steinman J S, Lammers C N, Valinski M E. A Proposed Open Cognitive Architecture Framework[C]. In: Proceedings of the 2009 Winter Simulation Conference. 2009: 1325-1335.
- [17] Campos J. Development in the Application of ICT in Condition Monitoring and Maintenance[J]. Computers in Industry, 2009, 60(1): 1-20.
- [18] Skok S, Sturlic I, Matica R. Multipurpose Open System Architecture Model of Wide Area Monitoring[C]. In: Proceedings of 2009 IEEE Bucharest PowerTech Conference. 2009: 2060-2065.
- [19] Nelson J A. Net Centric Radar Technology & Development Using an Open System Architecture Approach[J]. IEEE Aerospace and Electronic Systems Magazine, 2011, 26(5): 34-37.
- [20] Baehr J S. The Advanced TOMAHAWK Weapons Control System for Submarines[J]. Naval Engineers Journal, 1995, 107(3): 35-43.
- [21] Nelson H W. Distributed Control System (DCS) Network Architecture for a Shipboard Machinery Control System (MCS)[C]. In: Proceedings of Instrumentation, Systems and Automation Conference. 2002, 434: 119-131.

- [22] Christensen E, Miller A, Wing E. Waveform Application Development Process for Software Defined Radios[C]. In: Proceedings of IEEE 21st Century Military Communications Conference (MILCOM 2000). 2000: 231-235.
- [23] Rendek P, Jaw L, Smith G. Successful Trending and Diagnostics Technology Transition[C]. In: Proceedings of the 52nd ASME TurboExpo 2007. 2007: 823-827.
- [24] Vasilakos J, Devries R, Tompkins K T. Total Ship Open Systems Architecture[J]. Naval Engineers Journal, 2000, 112(4): 59-69.
- [25] Walden D D, Blanshan K M, Kranz G N. Technology Roll Lessons Learned on Embedded Avionics Platforms[J]. IEEE Aerospace and Electronic Systems Magazine, 2000, 15(5): 19-23.
- [26] Swearingen K, Majkowski W, Bruggeman B, et al. An Open System Architecture for Condition Based Maintenance Overview[C]. In: Proceedings of 2007 IEEE Aerospace Conference. 2007: 4161-4168.
- [27] Dunsdon J, Harrington M. The Application of Open System Architecture for Condition Based Maintenance to Complete IVHM[C]. In: Proceedings of 2008 IEEE Aerospace Conference. 2008: 4015-4023.
- [28] Hladik P E, Deplanche A M, Faucou S, et al. Adequacy Between AUTOSAR OS Specification and Real-time Scheduling Theory[C]. In: Proceedings of International Symposium on Industrial Embedded Systems (SIES' 07). 2007: 225-233.
- [29] Jo H C, Piao S, Cho S R, et al. RTETemplate Structure for AUTOSAR Based Embedded Software Platform[C]. In: Proceedings of IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications 2008 (MESA2008). 2008: 233-238.
- [30] Lanigan P E, Narasimhan P, Fuhrman T E. Experiences with a CANoe-based Fault Injection Framework for AUTOSAR[C]. In: Proceedings of 2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2010: 569-574.
- [31] Fang Z, Wu F, Zhang Y G, et al. Development of Diagnostic Communication Module on KWP2000 According to AUTOSAR[J]. Advanced Materials Research, 2011, 171-172: 462-467.
- [32] Wang Q, Shin W, Liu X, et al. I-living: An Open System Architecture for Assisted Living[C]. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC' 06). 2006: 4268-4275.
- [33] Weigand C. VITAL: Use and Implementation of a Medical Communication Standard in Practice[C]. In: Proceedings of Computers in Cardiology. 2005: 319-322.
- [34] Hengstler S, Aghajan H K. WiSNAP: A Wireless Image Sensor Network Application Platform[C]. In: Proceedings of the 2nd International Conference

on Testbeds and Research Infrastructures for the Development of Networks and Communities. 2006: 86-91.

[35] Suo Y, Miyata N, Morikawa H, et al. Open Smart Classroom: Extensible and Scalable Learning System in Smart Space Using Web Service Technology[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(6): 814-828.

[36] Julian V, Carrascosa C, Soler J. A Multiagent System Architecture for Retrieving and Showing Information[J]. Knowledge Organization, 2000, 27(1-2): 11-16.

[37] Bai X Y, Hou K J, Lu H, et al. Semantic-based Test Oracles[C]. In: Proceedings of the 35th Annual IEEE International Computer Software and Applications Conference (COMPSAC). 2011: 640-649.

[38] Cuenca L, Ortiz A, Vernadat F. From UML or DFD Models to CIMOSA Partial Models and Enterprise Components[J]. International Journal of Computer Integrated Manufacturing, 2006, 19(3): 248-263.

[39] Nam S H, Ryu K R, Kwon K E, et al. Architecture of Flexible and Reliable Control & Monitoring System for Distributed Machining Systems[C]. In: Proceedings of International Conference on Smart Manufacturing Application (ICSMA2008). 2008: 205-210.

[40] Salminen V, Pillai B. Methodology on Product Lifecycle Challenge Management for Virtual Enterprises[C]. In: Proceedings of Processes and Foundations for Virtual Organizations, IFIP TC5/WG5.5 4th Working Conference on Virtual Enterprises (PRO-VE' 03). 2003: 501-508.

[41] Kazi A S, Hannus M. Functional Requirements for Inter-enterprise Intranet Services[C]. In: Proceedings of the 4th International Working Conference on the Design of Information Infrastructure Systems for Manufacturing: Global Engineering, Manufacturing and Enterprise Networks (DIISM2000). 2000: 54-60.

[42] Wang F J, Mills J J. The Component-oriented Approach Towards Complex Product Development[C]. In: Proceedings of the 4th International Working Conference on the Design of Information Infrastructure Systems for Manufacturing: Global Engineering, Manufacturing and Enterprise Networks (DIISM2000). 2000: 83-94.

[43] Kobayashi T, Miyamoto H, Komatsu M. Human-Centered Design Approach for Middleware[J]. Fujitsu Scientific and Technical Journal, 2009, 45(2): 195-201.

[44] O' Neill T, Leaney J. Risk Management for an Open CBS Project[C]. In: Proceedings of the 8th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS2001). 2001: 45-53.

[45] Chin K S, Lam J, Chan J S F, et al. A CIMOSA Presentation of an Integrated Product Design Review Framework[J]. International Journal of Com-

puter Integrated Manufacturing, 2005, 18(4): 260-278.

**Author Contact:** gul@mail.las.ac.cn

*Note: Figure translations are in progress. See original paper for figures.*

*Source: ChinaXiv – Machine translation. Verify with original.*