

Middleware is STILL Everywhere!!!

Middleware! It was somewhere, and now it is definitely everywhere. Over the past four decades, the term middleware has been tossed around, picked up, and investigated vigorously. Yet, if you ask 10 different people “what is middleware?” You will most likely get 10 different answers. It started as some additions on top of operating systems to facilitate complex applications development, moved to become data integration features, then became network applications facilitator, and eventually became an important component of every distributed environment, application, system and platform there is. To-date, if you examine any type of distributed system or application, you must find middleware or some middleware functionality involved. Although recently, the term itself is becoming less used, yet it still exists in mobile and sensor networks, service-oriented architectures, grid computing, cloud computing, online multi-player games, networked robotics, the Internet of things, and much more. So as is, middleware is really still everywhere and most likely will remain everywhere for a very long time.

MIDDLEWARE THROUGH TIME

Middleware as a term appeared in the early 1970s according to [1]. The main mention refers to middleware as program enhancements to the operating system that help reduce the complexity of applications development and offer features not generally available in the operating system. However, until the early 1990s, the term was not much in use. During that period, several notions relevant to middleware started to appear although they did not carry the name middleware. There were several programming environments that provided abstractions and extended libraries to facilitate applications development, and these by essence belong to the middleware era. In addition, some tools and integration techniques were also used that resemble middleware functionality. Following the two-tier architecture that was becoming very common since the 1980s (the client/server model in particular), the three-tier architecture appeared. In this model, a third layer was introduced to facilitate the operations between the clients and the servers. This layer is usually referred to as the application server [2]. However, examining the model closely shows that the middle layer is actually nothing but the middleware. One interesting view of middleware is that it is the slash in “client/server” [3]. In the 1990s, another issue was becoming of high importance, and that is the integration between legacy systems that cannot be rebuilt or replaced with new applications designed and built using new programming models and tools. The two entities could not be merged, and it is usually very hard to match the interfaces among the two. Therefore, one notion of middleware appeared to help solve this problem. Data integration middleware such as Common Object Request Broker Architecture (CORBA) [4] came into the picture in 1991 and provided an excellent solution to the application integration and data interfacing issues. CORBA offered data and process integration standards agreed upon by more than 700 companies worldwide, which resulted in the first concrete definition of middleware. From that moment, the term middleware sort of got tangled with CORBA and for many years to follow, if anyone mentions middleware, everyone automatically think CORBA. However, as the years passed, several issues and limitations appeared, and CORBA was not able to offer solutions for all types of distributed applications. For example, CORBA works mainly on LAN and mostly within a single organization. In addition, it imposes high overhead on the applications, which makes it unsuitable for grid, cloud, wireless sensor network (WSN), and mobile applications. Thus, the term middleware started to get free of its association with CORBA and began to cover a much wider range of tools and services to support the different distributed computing requirements.

DIFFERENT VIEWS OF MIDDLEWARE

The traditional definitions of middleware vary in how they approach the term [3, 5–14], yet many agree on something that sounds like the following definition. “Middleware is the software that acts like a {glue, intermediary, broker, middle man, interpreter, abstraction provider, consolidator, integrator, facilitator, or connector} between one or more applications and other application(s) or between one or more applications and the underlying infrastructure including the platforms and operating systems used.”

In the different definitions we found of middleware, we can see how diverse and versatile the term and its underlying meaning. At the very early stages, it was considered a way to simplify some complex applications development issues such as adding specialized libraries to programming languages that support some specific functions. It was also part of some development tools that promote better utilization of some resources available such as the network interface cards. In such case, we can actually consider the network sockets as the middleware that allows us to access those cards without having to worry much about the lower-level details. Later on as data usage and application interfaces became a big issue, middleware came to the rescue by providing the necessary tools and features that allow for application and data integrations, provide better access models to data sources, and offer simplified high-level interfaces to different applications. As distributed systems started to pick up popularity, and development and research in that area became more significant, middleware played a major role in the process. IBM [15], as one of the very strong advocates of middleware, offered many middleware type solutions such as WebSphere. At IBM, several research and product-line projects are underway in distributed middleware, advanced enterprise middleware, event-based middleware, middleware services, and middleware for the Cloud. At Oracle, another advocate for middleware, the main middleware solution suite is the Oracle Fusion Middleware [16], which includes various tools such as Oracle WebLogic Suite, Oracle WebCenter, Oracle SOA Suite, Oracle Identity Management, and Oracle JDeveloper. Although it does not openly use the term middleware, Sun Microsystems offer the J2EE (Java 2 Platform, Enterprise Edition) development environment [17]. It provides various middleware functionalities such as distributed objects and remote method invocation for distributed application developers. Besides the industry, research in academia also covers middleware from many different angles and functionalities. For example, several conferences, workshops, symposiums, and journals were dedicated to middleware. One of the most important venues is the ACM Middleware Conference, which started in 1998 and ran its 12th edition in 2011 [18]. Another example is the international symposium on middleware and network applications (MNA) [19], which we offer this issue based on the best papers presented in its second edition. More examples can be found in the listing in [20].

WHY IS MIDDLEWARE IMPORTANT

Middleware has become an indispensable part of distributed computing. It is practically impossible to build large-scale distributed applications without involving middleware. By comparison, trying to build a distributed application without middleware is like trying to write a simple application on a personal computer without the operating system. As middleware developed, the main goals, designs, architectures, and features it provides have also evolved. Currently, we could summarize the main functionalities/features of middleware in the following points:

- Offering tools and functions to simplify the development of complex applications.
- Offering high-level abstractions and interfaces to facilitate application integration, reuse, and development.
- Hiding the heterogeneity of the underlying platforms and operating environments.
- Hiding the distribution and communication details in the underlying environment.
- Facilitating communications among the different distributed components of the infrastructure.
- Providing common services for general purpose functions needed by different applications to reduce development efforts and avoid duplication of services.

- Providing a common architecture to add new services and features without having to change the applications.
- Offering added-value features and nonfunctional properties such as security, reliability, and QoS.
- Supplying the necessary tools to enhance performance and increase the stability and scalability of the distributed applications.

OUR DEFINITION OF MIDDLEWARE

As you see, it is not easy to arrive at a concise definition of middleware, but let us try. Middleware is a lot like the middleman. In practical terms, a middleman could be a real estate agent, a stock broker, a translator, a driver, a salesman, a sports commentator, a handy man, and much more depending highly on where and how a middleman is needed. In the world of distributed computing, middleware could be found everywhere in the system. It is the underlying common features needed for communication, it is the services used to translate and integrate data from different sources, it is the functions used to offer common operational functionalities, it is the tools used to simplify application development, and it is the models used to facilitate reuse and application integration, it is the added values of the applications. So to put it in simple terms “*middleware connects any set of components in a distributed environment to offer better functionality.*” That component could be an application, a task within an application, a platform, a communication network, a piece of hardware (e.g. robot, sensor, microcontroller, etc.), a server, a client, a service, a grid node, and so on.

WHAT COMES NEXT

Although in the past few years the emphasis on middleware – as a term – has slightly subsided, the technology itself is still alive and thriving. It is surviving within all distributed applications, services, and environments currently evolving. If we examine closely these new models, approaches, architectures, and tools, we can find “middleware” written all over them. In service-oriented computing, middleware functionality is present through the service-oriented architecture and as service-oriented middleware. Yet, with service-oriented computing challenges in development and operations were also present, and as the complexity of these models increased, the need emerged again for middleware solutions to help support them. Hence, the notion of service-oriented middleware came into existence [21]. In addition, in the grid and high-performance computing architectures, middleware played a key role as facilitating layers and models [22, 23]. As the complexity of these environments and their applications increases, middleware will continue to be the most important technology in the field for a long time. Web-based applications and services also require middleware support [24] and will continue to need it. Furthermore, the most recent advances in cloud computing also silently promote middleware as it is a vital component that makes cloud computing happen [25, 26]. In various types of large-scale distributed applications such as online multi-player gaming [27], robotics [28], and sensor networks [29], middleware is also present to facilitate communication, real-time support, QoS, and other reliability and scalability capabilities. At IBM, a “Middleware is Everywhere” campaign promoted middleware as an effective and feasible approach in distributed computing [30]. Many opposed the view, particularly Sun Microsystems [31], and yet over time others supported the statement. As middleware technologies evolved over the past four decades, it has secured a prominent position in the distributed computing era in all its varying stages and approaches. As for the future, researchers and industry key players agree that middleware will remain the driving force behind most distributed applications. It will offer the necessary tools and functionality to enhance the design and development process, enrich the operational environments, and enhance overall performance and value. IBM, as a strong advocate for middleware, stands by their products as the future middleware tools [32], whereas Microsoft affirms the same from their own perspective [33]. We also foresee that the trend will continue into the coming years, and middleware will be an essential component of all distributed computing systems regardless what name it may end up carrying.

MIDDLEWARE IN NUMBERS

Here we will try to provide a brief view of the research status of middleware in terms of published work. As it is common in any field and as interest rises in the field, more work gets published. The same applies to middleware. We used Google Scholar to find the total number of published work including the word “middleware.” The publication could be a conference/journal article, white paper, or patent. We covered several years (1 year at a time using the advanced search options) starting from the early mentions of middleware in scientific publications. Based on the search, we found that the term appeared around 1990 and publications increase as the years passed until it peaked in 2008 (see Figure 1). After 2008, a slight drop in the number of publications occurs, yet it does not really mean a decline in work in middleware. On the contrary, this decline occurred because of the use of other terms that inherently cover the same functionalities of middleware. One prominent example is cloud computing, where middleware is one of the main enabling technologies. Another term used is the Platform as a Service, which is also middleware in disguise. The volume of “cloud computing” publications in Google Scholar is shown in Figure 2. We covered the recent years, where cloud computing have picked up steam and became a strong field of research. As shown in Figure 2, there is a sharp increase in publications in the past 4 years and this reflects to the importance of middleware as it is the driving technology in cloud computing.

EXAMPLE APPLICATIONS USING MIDDLEWARE

This special issue on middleware and network application is an attempt to capture the current and optimistically the future of middleware. The second international symposium on middleware and network applications (MNA 2011) hosted several presentations on various aspects of middleware. It reflects on the growing interest in middleware and the huge impact of network applications. The symposium advocates for middleware and network application and tries to encourage the research communities to explore and discuss current and future advancements in the field. We tried to cover areas such as issues related to social-awareness, context-awareness, and power-awareness;

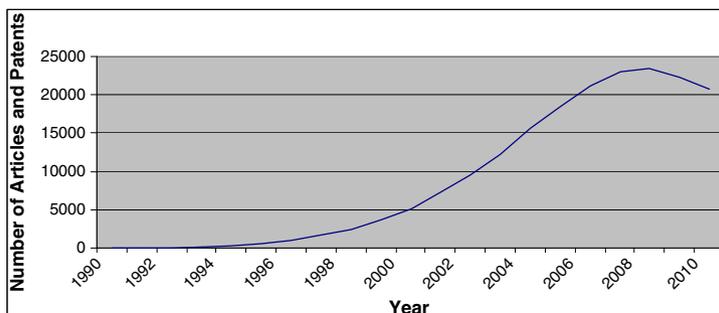


Figure 1. Appearance of “middleware” word on Google Scholar.

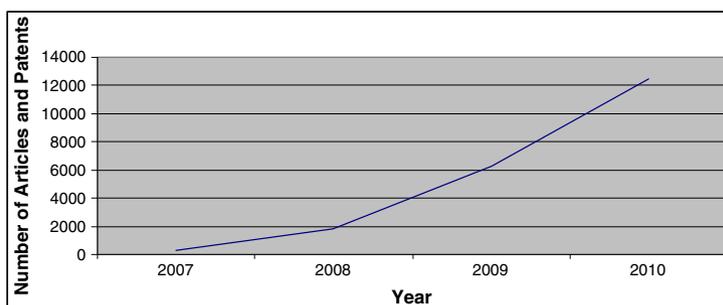


Figure 2. Appearance of “cloud computing” in Google Scholar.

autonomocity; security; scalability; evaluation; and software engineering techniques. In addition, we targeted domain specific middleware such as web-services, robotics, sensor networks, clusters, the grid, and the cloud. To further enrich the topics, we also included network applications and the issues involved that usually rely on middleware technologies such as cloud applications, emergency response network applications, ubiquitous and pervasive applications, mobile applications, distributed educational applications, remote health monitoring, and many more.

It is practically impossible to cover even a small percentage of the work being done in middleware, yet we could give a few examples. Using the selected papers from MNA for this issue and several other examples from other research projects, we will demonstrate some of the new advances in middleware technologies.

One way to make use of middleware is to assist in enhancing and optimizing network operations and performance. Including monitoring, reorganization, and adaptive features within the network can greatly improve its operations. These features are more easily added as middleware services performed by centralized or distributed components. For example, in [34], the authors opt to utilize mobile agents to monitor and tune the network performance on the grid. This approach allows the agents to find performance problems and resolve them on the spot, thus enhancing over all network performance. Another example is to introduce a decentralized cognitive plane to facilitate network self-management to ensure meeting QoS requirements [35]. In this issue, we offer another example in this direction where specific features are introduced to the network to optimize network performance [36]. Although the approach is not explicitly described as middleware, the general architecture and model is within the concept. The main goal of this optimization process is to reorganize the nodes in the network in an opportunistic manner to minimize the total communication distances and maximize available bandwidth. As a result, power consumption will be reduced, and effectively, the carbon foot print of the network will be reduced.

The second accepted paper in this issue covers wireless sensor networks by introducing a meta-data layer [37]. The authors propose separating the configuration properties from the runtime components. In this manner, the layer will offer flexible configurations of the wireless sensor network, while keeping the operational components independent. The main goals achieved using this approach are: improving the support for concurrent component use, optimizing resources utilization, and reducing the efforts to configure the network. Other works in middleware for wireless sensor networks have been done over the past decade and [29] offers a glimpse into some examples in service-oriented middleware for wireless sensor networks. Examples of these middleware approaches include SStreaMWare, Sensor Web, USEME, OASiS, and B-VIS.

Research in robotics also benefited from the advancements of middleware technologies. The introduction of middleware functionality has enhanced the development process, allowed for added-value features, and also helped separate common functionalities from operational requirements. In [38], a middleware approach is proposed to facilitate the development of complex real-time applications for robots. Using this middleware features to handle communication requirements and resolve issues such as deadlocks are made available. Applications to allow multiple robots to communicate and collaborate together can be developed easily. Another area where middleware is useful for robotics is for integrating robotic systems with enterprise and information systems. One example is using a middleware to integrate industrial robotic arms with a BCI2000 software [39]. In [40], another article in this issue, a middleware is introduced to support QoS and high-performance requirements of networked robotics. Nerve, a lightweight middleware, is introduced to handle the robots networking requirements such that changes in the application requirements can be included in an easier manner and without highly affecting the real-time performance of the robots. The middleware is based on a data distribution service, which is a publish/subscribe standard for real-time applications.

Last, but definitely not least, middleware has played an important role in the advancement of distributed and online educational systems. Several applications and systems have been designed and used that could have been very difficult to implement and use without the use of middleware. I-Minds [41, 42] utilizes middleware and multiagent technologies to facilitate a virtual classroom. The middleware allowed for rapid development of the system and also helps the runtime operations where multiple users are collaborating online or within a classroom using the system. Another example in this area is the introduction of middleware services to support voice-enabled learning

services [43]. In this paper, included in this issue, the distributed interactive learning application is supported by a service-oriented architecture (middleware), which enables cross-platform multi-channel access to Internet-based learning. The design, implementation, and operational issues are discussed and evaluated.

The application domains discussed in this issue give an example of the wide range of application domains, operating environments, and target distributed platforms, where middleware has become essential. It supports functionalities, not only for the applications but also for the underlying communication infrastructure, the platforms, and operating environments. The various advantages gained by including or utilizing middleware support are great and outweigh the design and execution overhead imposed. In addition, the current research has come up with various ways to reduce this overhead and/or offset it with additional gains in performance, availability, and QoS. Furthermore, we expect the future to offer even more advances in this regard such that using middleware will become a standard model rather than just a preferred approach.

CONCLUSION

In this issue, we offer a glimpse of the current innovations in middleware and network technologies. As illustrated by the articles in the issue [36], [37], [40] and [43], the role of middleware is not always explicit, yet closely examining any distributed or network application, the middleware functionalities will be there at some level. It is important to keep in mind the broad definition of middleware and the diverse roles it could take within a distributed environment. It is generally the glue connecting different resources to satisfy certain requirements. It offers heterogeneity abstractions, uniform interfaces, data, application and system integration, facilitates application development and reuse, and allows for adding common features and added-value properties. The four articles selected for this issue offer a range of application domains where middleware has become essential. We expect future distributed systems development to rely more on middleware technologies in their various flavors and types. However, we also acknowledge that many of these technologies may no longer carry the name “middleware” as the current trend has started to move towards different ways of naming it. However, the names do not matter much as the approaches all relate to each other in one way or another and satisfy the same set of goals. We truly hope that this issue will help increase the interest in middleware technology and offer useful information and research directions to researchers in the field.

As middleware evolved over time, it is clear that it also secured a prominent position as an important enabling technology for distributed applications. IBM earlier campaigned for “Middleware is Everywhere,” now we see that middleware is defiantly affirming this fact, and it is everywhere. The vast contributions of the middleware research communities and the industry has allowed middleware to thrive and evolve. As a result, it is now an indispensable part of distributed computing and will continue to be so for a very long time. The future promises even more advancement in this field and exciting additions that will further enhance the distributed applications development and operations. As the current trends in grid and cloud computing evolve, middleware technologies will also evolve. However, the growth could be much faster and more drastic if the industry and research communities could come up with a set of guidelines that would unify the efforts and facilitate integration and reuse in a more fluid way. Some may suggest having specific standards to govern the field, yet because of the huge diversity of the distributed computing era and the vast spectrum of requirements, we believe it may be extremely difficult to arrive at a set of standards that would be satisfactory for all. The CORBA community successfully offered their standards and they worked very well. However, they were limited to a specific set of application requirements and facilitated a specific target, namely data/application integration within specific types of environments. Applying the same model, it may be possible to create some standards that govern specific areas of the field such that the developers and researchers will have something to start from. Yet, these should be flexible enough and independent from any particular technology or organization to allow for further developments and advancements in the field. So, to sum it all up, middleware is still and will remain everywhere.

REFERENCES

1. Web: Google Answers, <http://answers.google.com/answers/threadview?id=268580>, viewed, November 2011.
2. Web: Ramirez AO. "Three-Tier Architecture," in *LINUX Journal*, Issue 75, July 2000. <http://www.linuxjournal.com/article/3508>, viewed, November 2011.
3. Web: Middleware Resource Center, <http://www.middleware.org/whatis.html>, viewed, November 2011.
4. Keller A. Service-based systems management: using CORBA as a middleware for intelligent agents. *Proc. IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, L'Aquila, Italy, October 1996.
5. Web: Answers.com, <http://www.answers.com/topic/middleware>, Viewed November 2011.
6. Web: Business Dictionary, <http://www.businessdictionary.com/definition/middleware.html>, viewed, November 2011.
7. Web: Dictionary.com, <http://dictionary.reference.com/browse/middleware>, viewed, November 2011.
8. Web: DiMaggio L. What is middleware? In plain English, please., REDHAT Magazine. <http://magazine.redhat.com/2008/03/11/what-is-middleware-in-plain-english-please/>, viewed, November 2011.
9. Web: FOLDOC: Free On-Line Dictionary of Computing, <http://foldoc.org/middleware>, viewed, November 2011.
10. Web: SearchSOA, <http://searchsoa.techtarget.com/definition/middleware>, viewed, November 2011.
11. Web: TechTerms.com, <http://www.techterms.com/definition/middleware>, viewed, November 2011.
12. Web: The Free Dictionary, <http://www.thefreedictionary.com/middleware>, viewed, November 2011.
13. Web: WeboPedia, <http://www.webopedia.com/TERM/M/middleware.html>, viewed, November 2011.
14. Web: WikiPedia, <http://en.wikipedia.org/wiki/Middleware>, viewed, November 2011.
15. Web: IBM M. <http://www.ibm.com/search/csass/search?q=middleware&cc=us&en=utf&co=us&sn=mh&lang=en&lo=any> and <http://www-935.ibm.com/services/us/en/it-services/middleware-services.html>, viewed, November 2011.
16. Web: Oracle Fusion Middleware, <http://www.oracle.com/us/products/middleware/index.html>, viewed, November 2011.
17. Web: J2EE, <http://www.java.com/en/download/faq/techinfo.xml>, viewed, November 2011.
18. Web: ACM/IFIP/USENIX 12th International Middleware Conference, <http://www.middleware-conference.org/history.html>, viewed, November 2011.
19. Web: International symposium on Middleware and Network Applications, <http://mna2011.middleware-tech.net/>, viewed, November 2011.
20. Web: WikiCFP, Middleware, <http://www.wicicfp.com/cfp/call?conference=middleware>, viewed, November 2011.
21. Al-Jaroodi J, Mohamed N. Service-oriented middleware: a survey. *The Journal of Network and Computer Applications*, Elsevier 2012; **35**(1):211–220.
22. Al-Jaroodi J, Mohamed N, Jiang H, Swanson D. Middleware infrastructure for parallel and distributed programming models on heterogeneous systems. *The IEEE Transactions on Parallel and Distributed Systems – Special Issue on Middleware Infrastructures* 2003; **14**(11):1100–1111.
23. Engelmann C, Ong H, Scott SL. Middleware in modern high performance computing system architectures. In *Computational Science – ICCS*, Vol. 4488/2007, Lecture Notes in Computer Science, 2007; 784–791.
24. Tolkstdorf R, Nixon LJB, Bontas EP, Nyumen DM, Liebsch F. "Enabling real world semantic Web applications through a coordination middleware," in the Semantic Web: Research And Applications, Lecture Notes in Computer Science 2005; **3532/2005**:9–72.
25. Azeez A, Perera S, Gamage D, Linton R, Siriwardana P, Leelaratne D, Weerawarana S, Fremantle P. Multi-tenant SOA middleware for cloud computing. *Proc. IEEE 3rd International Conference on Cloud Computing (CLOUD)*, Miami, Florida, USA, July 2010; 458–465.
26. Caron E, Desprez F, Loureiro D, Muresan A. Cloud computing resource management through a grid middleware: a case study with DIET and eucalyptus. *Proc. International Conference on Cloud Computing*, Bangalore, India, September 2009; 151–154.
27. Hsiao T-Y, Yuan S-M. Practical middleware for massively multiplayer online games. *IEEE Internet Computing* 2005; **9**(5):47–54.
28. Mohamed N, Al-Jaroodi J, Jawhar I. A review of middleware for networked robots. *The International Journal of Computer Science and Network Security* 2009; **9**(5):139–148.
29. Mohamed N, Al-Jaroodi J. A survey on service-oriented middleware for wireless sensor networks. *Service Oriented Computing and Applications*, Springer 2011; **5**(2):71–85.
30. Web: Leclerc A. "Middleware Is Everywhere!" Article on Gantthead.com, posted September 2000. <http://www.gantthead.com/content/articles/11413.cfm>, viewed, November 2011.
31. Web: Schwartz J. "Middleware is history," from J Steven Perry, The Blog, Posted January 2006. <http://jstevemperry.wordpress.com/2006/01/24/jonathan-schwartz-middleware-is-history/>, viewed, November 2011.
32. Web: GEELAN J. "The Future of Middleware & Tools," in JDJ, Q & A with the 5 General Managers of IBM's Software Group, posted April 2004. <http://java.sys-con.com/node/44622>, viewed, November 2011.
33. Web: Kawasaki B. "The Future of Middleware?" Paul Somers Connected Systems Blog, posted January 2011. <http://paulsomers.blogspot.com/2011/01/future-of-middleware.html>, viewed, November 2011.
34. Valliyammai C, Selvi ST, Kumar RS, Pradeep E, Naveen K. "Multi-agent based network performance tuning in grid," in *Information Processing and Management, Communications in Computer and Information Science* 2010; **70**:297–304.

35. Manzalini A, Deussen PH, Nechifor S, Mamei M, Minerva R, Moiso C, Salden A, Wauters T, Zambonelli F. Self-optimized cognitive network of networks. *Computer Journal, Oxford Journals* 2011; **54**(2):189–196.
36. Sami H, Marimuthu P. Optimizing network Performance and Carbon Offset through Opportunistic Reclustering; **24**(16):1927–1939.
37. del Cid PJ, Hughes D, Michiels S, Joosen W. Applying a Meta-Data Level for Concurrency in Wireless Sensor Networks; **24**(16):1953–1962.
38. Finkemeyer B, Kröger T, Wahl FM. A middleware for high-speed distributed real-time robotic applications. In *Robotic Systems for Handling and Assembly*, Vol. 67/2011. Springer Tracts in Advanced Robotics, 2011; 193–212.
39. Henderson A. A design for a middleware communications layer between an industrial robotic arm and the BCI2000 software package. *Proc. Florida Conference on Recent Advances in Robotics (FCRAR)*, Jacksonville, Florida, USA, May 2010.
40. Martinez J, Romero-Garces A, Bandera JP, Marfil-Robles R, Bandera A. A DDS-based middleware for quality-of-service and high-performance networked robotics, **24**(16):1940–1952.
41. Soh L-KXL, Zhang X, Al-Jaroodi J, Jiang H, Vemuri P. “*I-MINDS: An Agent-Oriented Information System for Applications in Education*,” *Lecture Notes in Artificial Intelligence (LNAI 3030) Series Agent-Oriented Information Systems*. Springer-Verlag, 2004.
42. Soh L-K, Khandaker N, Jiang H. I-MINDS: a multiagent system for intelligent computer-supported collaborative learning and classroom management, in the *International Journal of Artificial Intelligence in Education* 2008; **18**(2):119–151.
43. Álvarez García VM, Paule-Ruiz MP, Riestra-González M, Pérez Pérez JR. Voice interactive classroom: best practices and design strategies, **24**(16):1963–1973.

JAMEELA AL-JAROODI
*Faculty of Information Technology,
UAE University, Al Ain,
United Arab Emirates*
E-mail: jaljaroodi@gmail.com

NADER MOHAMED
*Faculty of Information Technology,
UAE University, Al Ain,
United Arab Emirates*
E-mail: nader.m@uaeu.ac.ae