

E-Business Framework Using an Enhanced Web 2.0 Component Model

Karim Ouazzane¹, Sajid Afzal², Vassil Vassilev³, Kaushik C. Patel⁴, Althaff Irfan⁵

¹⁻⁵Faculty of Life Sciences and Computing, London Metropolitan University,
166 - 220 Holloway Road, London, N7 8DB, UK

¹k.ouazzane@londonmet.ac.uk; ²sajid.afzal@live.com; ³v.vassilev@londonmet.ac.uk; ⁴kcp@askdirect.co.uk;
⁵althaff_mohideen@yahoo.com

Abstract-One of a current research challenge is the development of efficient and secure technologies for next generation Web 2.0 based enterprise e-business solutions for the e-retail sector. In an effort to avoid the shortcomings of the request management component in SOAW2, this paper introduces an enhanced Web 2.0 framework called W2ASVB (i.e. Web 2.0 Architecture for Service and View Brokerage). The framework enhances SOAW2 with two fundamental improvements: an addition of a Request Broker function and a replacement of Supporting Functions with Service Adapters. A layered representation of the framework is also designed. This framework provides small retailers with an e-business transformation solution by customizing their e-businesses according to their individual needs.

Keywords- *e-business Framework; Request Broker; Business Work Flow*

I. INTRODUCTION

The wide-spread use of information systems, particularly Internet-based systems, has globally changed the way people do business. During the last two decades, the introduction of IT systems has presented new opportunities and offered challenges to many different kinds of businesses. Steven Ashton [1] commented: "People are looking to spend more on IT because they are looking to gain competitive advantage". New research at CISCO [2] concluded, "Businesses that embrace new technologies grow faster". This global trend towards introducing IT systems to the business sector has also had a deep impact on the British business community. Nick Watson [2] states "With British businesses contending with increasingly tight markets, stronger competition from overseas, and key labour shortages at home, companies are turning to strategic technology investments to help them improve operational efficiency, cut costs and drive business growth".

The rapid e-business transformation by large enterprises has widened the gap between large enterprises and small businesses in terms of business growth, with local retail shops as a worst affected area. Dowd [3] in the "High-Street Britain: 2015" research report highlighted the essence of small retail shops: "The vast majority of contributors agreed that all small shops are important to, and influenced by, economic, social and political trends. The small retail sector is a key driver of: entrepreneurship, employment, skills, local economies, innovation, and sophisticated business networks, as well as accessibility to vital goods and services, diversity, social inclusion and community activities". Therefore, the key problem is to support small businesses with sophisticated and cutting-edge technologies to develop next-generation Web 2.0 based enterprise e-business solutions by using the Internet as a platform for the electronic retail domain; this will facilitate the swift transformation of traditional businesses into e-businesses by fulfilling the needs of each individual business at an affordable price.

According to Simmonds [4], Internet usage statistics indicated that nearly 1.7 billion people have been connected with each other on the Internet at the time of the study, and this figure is rapidly increasing on a daily basis. Due to the huge popularity of online social network environments, the Web has moved into a second generation; i.e. it is now a self-contained operating platform. Fig. 1 illustrates the evolution of the Internet and the transition from Web to Web 2.0.

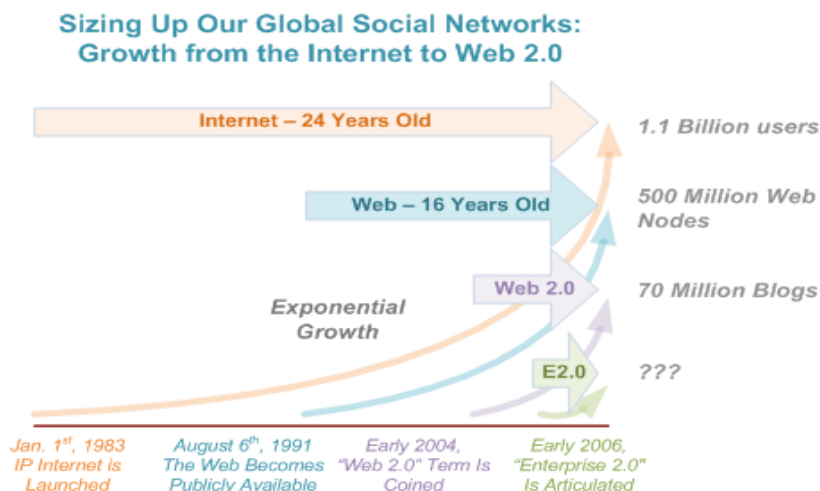


Fig. 1 Transition of Internet to Web 2.0 [5]

Every technology requires a clear and effective model for representing its components and the interaction between them. Unfortunately, there has been a little research undertaken to build a clear understanding of Web 2.0 and its application paradigm to accompany this level of usage. Until recently, no such model for Web 2.0 has existed. Research conducted by Omar & Taleb-Bendiab [6] proposed SOAW2 as the first model framework defining Web 2.0 components and their relationships. They applied the SOA approach to outline the important components of Web 2.0 and SOA technologies in five core and three support layers, as illustrated in Fig. 2.

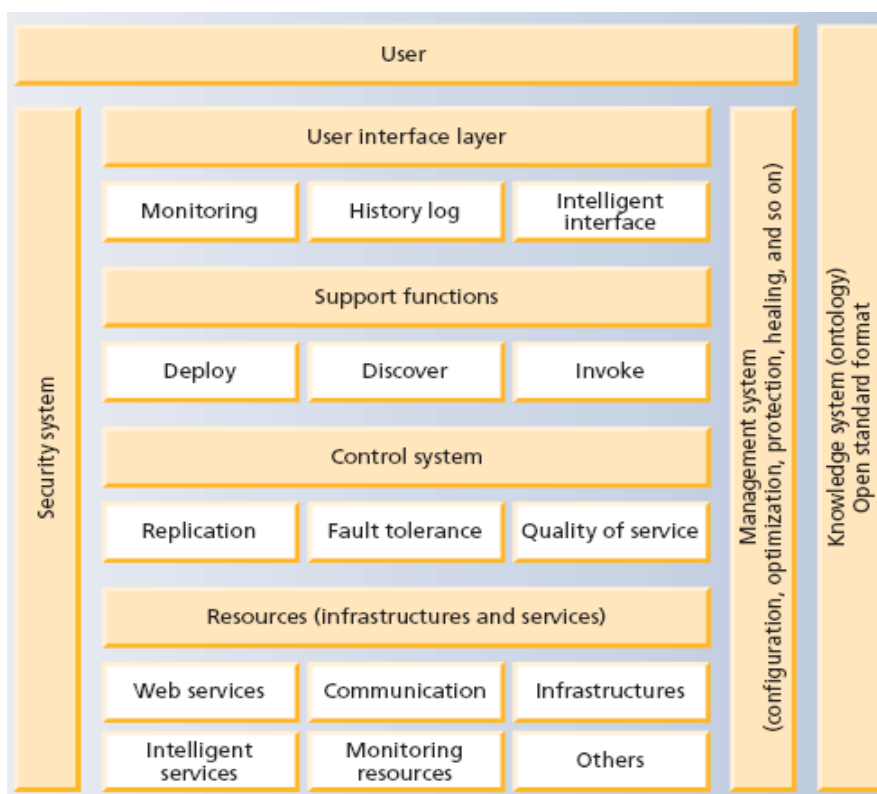


Fig. 2 SOAW2 Model Framework by [6]

The Resources layer in SOAW2 acts as a container for core services such as financial services, stock services, etc. These services play a part in building further services. The Control system layer represents a placeholder for a model manager which, monitors and controls the services residing in the Resources layer. The Support function layer contains a standard SOA facility for service providers to deploy services. The User layer represents an end-user (i.e. consumer) who interacts with the system via the user-interface layer. The remaining three support layers encapsulate the overall management and support capabilities of the model framework. Fig. 3 illustrates the collaboration between different components of the SOAW2 model framework in action. SOAW2 itself is not a new model framework; rather, it is an outcome of research activity that analyzed pre-existing models of Web 2.0 systems in depth.

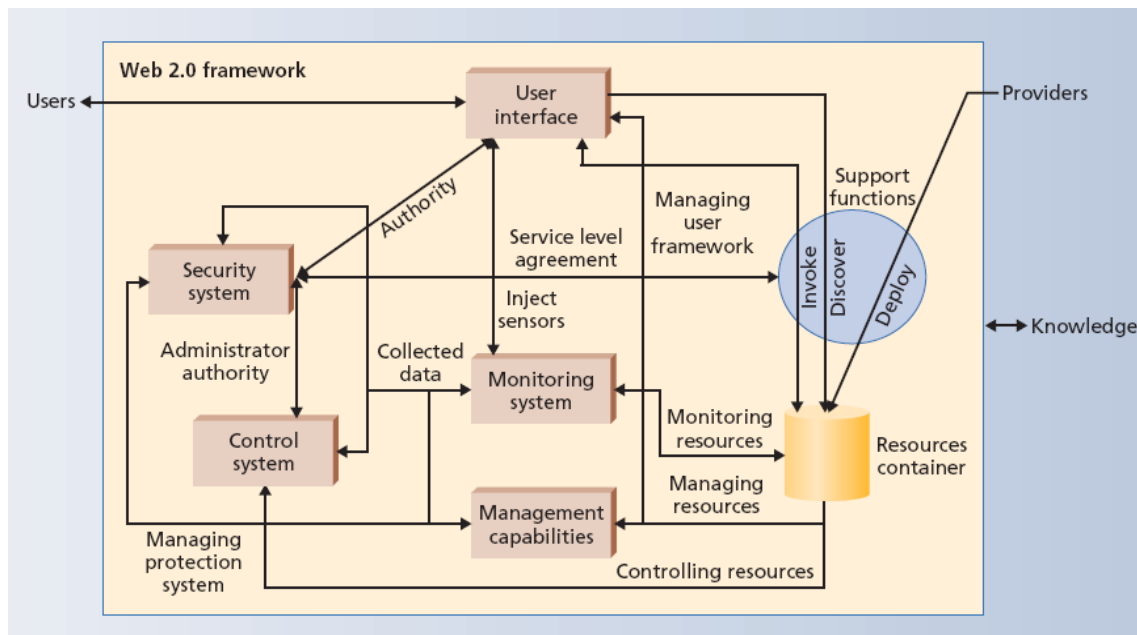


Fig. 3 SOAW2 in Action [6]

Recently, some implementations of the framework have appeared on the market of Web 2.0 tools. MerchantOS [7] is a Web 2.0-based POS system developed by a U.S.-based company. The system is completely Web-based and does not require installation of any additional software. The toolset offers transformation of manual POS, Inventory Control and Customer Relations business procedures into a full e-business solution. But MerchantOS only offers standard business procedures and a user interface to run the POS system. It does not offer any customization facilities, such as adapting standard business logic to company-specific business logic, or customizing the standard user interfaces into the company-specific user interfaces. The PHP Point of Sale [8] system is the outcome of the dedication of the open source software developer community to help small businesses in their effort to achieve e-business transformation. This system implements industry-specific business procedures for sales, purchase and stock management activities, and supplies standard user interfaces. This POS system is not feasible for small retailers who require tailored business logic and/or user interface customization. Zoho [9] is one prominent Web 2.0 company that is progressively making contributions to the wide endorsement of Web 2.0 platforms for e-Business. They have systematically transformed a number of desktop applications into Web 2.0 components and offer ZohoCRM, which is an affordable on-demand customer relationship management system with integrated POS. Despite offering valuable features for user access control and customized business reporting, ZohoCRM does not facilitate on-demand customization of its business logic and user interfaces.

A market review of the current Web 2.0 retail systems that are built on the existing SOAW2 model has revealed a lack of essential functionalities that would be desirable in a framework on this scale, such as on-demand customization of business logic and business-tailored user interfaces. The fundamental reason for this drawback is the lack of explicit request management capabilities in the middle layers of the SOAW2 model framework. According to Dorn, Rainer and Hrastnik [10], request management in web-based systems is a two-step activity, comprising of Action Management in the resource container and View Management at the front-end. Action Management involves resolving the incoming request and dispatching it to an appropriate service (i.e. service binding and execution). The current SOAW2 model framework directly exposes the user interfaces to business services that reside inside the resource container. The need for supporting functions between the user interfaces and the resource containers is completely ignored as they do not perform any business-related tasks, except the searching and binding of the business services. This direct exposure resulted in the direct mapping of user interfaces to business services without any logical control. Due to this fundamental shortcoming of the SOAW2 framework, developed systems do not provide on-demand customization of business logic at run-time. Also, the current framework does not support extra customization of the business logic. Further investigation is required to develop a more adequate Web 2.0 model framework that will overcome the fundamental shortcomings of SOAW2.

The practical implication of these shortcomings is the current frameworks' inability to handle complex scenarios. On the other hand, the request brokering technique has proved very successful in many complex scenarios, as in Hitachi [11] and XiaoQin [12] researches. For example, Howard and Kerschberg [13] proposed a complete framework called KDSWS (Knowledge-based Dynamic Semantic Web Services). Their approach is based on a multi-agent system in which web services have been converted into semantic web services. Four separate agents support the KDSWS's brokerage: the broker, classifier, discoverer and selector. This framework still does not have facilities for mapping the response back to a user-defined interface. Furthermore, the developed model is not applicable in situations where a request demands execution of more than one service, or requires a partial collaboration of different services in order to fulfil multiple user needs. The request broker model proposed

by [12] is a concept of an agent-based web-services platform. The scope of the proposed brokerage model is limited to handling e-commerce operations only. The request broker provides automated assistance to help consumers find feasible offers and at the same time assists the operator in proposing different offers to the consumers. The request broker model proposed in this research is an extension of the Architecture for Information Brokerage Service models proposed in [1]. Another extension is the CORBA-based mediation platform for efficient brokerage proposed in [14]. The design is primarily based on the concept of the complete mediation platform originating in [15]. The proposed model also complies with TINA and OMG specifications. The elegance of this model's design is demonstrated in its dynamic profile and profile rule management techniques. Unlike [13], this model has an applied profile management philosophy on both user and service provider ends.

In conclusion, none of the Web 2.0 retail systems built on the SOAW2 model framework seems to exhibit all the characteristics that are needed to further improve the applicability of the framework to relevant e-business tasks. Amongst other issues, the major factor that causes this weakness is the absence of a request management component in the middle layers of the architecture. A review of popular brokerage techniques has shown that the request management component has proven its strategic importance and promises successful application in realistic business scenarios. Therefore this can be considered a candidate for fundamental changes in the SOAW2 model framework. It has also been observed that existing request broker techniques are more service-centric than view-centric; i.e., they are primarily focused on routing a user request to the best matching destination service, but are incapable of routing the service output back to the best-matching user interface destination. Hence, existing request broker techniques need to be integrated more fully into the model framework through a separate request management component. This is also the main motivation behind this paper.

II. BUSINESS CASE

This section provides a high-level description of retail industry practices for the purpose of the framework development.

A. Core Infrastructure

In a typical e-Business scenario, the software tools transform the business activities into services. These services contain the implementation of both business data persistence and business workflow logics specific to that business. The software developers connect the user interfaces (i.e., views) with the corresponding services to make a system capable of performing operational requests. The user initiates the operational request, as illustrated in Fig. 4.

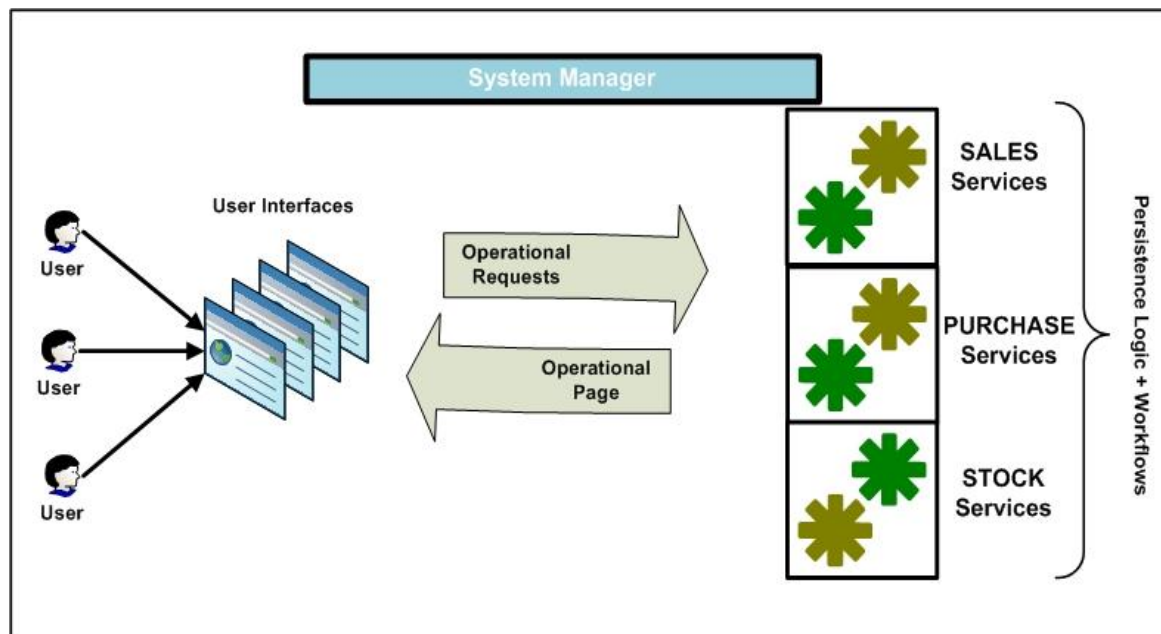


Fig. 4 A typical e-business workflow

As discussed earlier, the scenario for developing an enterprise system that not only holds the business data and services but also provides sharing and customization of the standard business logic is highly desirable. Such a system could be produced by a framework that not only encapsulates business logics but also provides tools for mapping the requests and responses in both directions. Furthermore, such a framework would allow merchant businesses to share the same infrastructure by merchant businesses. Fig. 5 illustrates such an enterprise system.

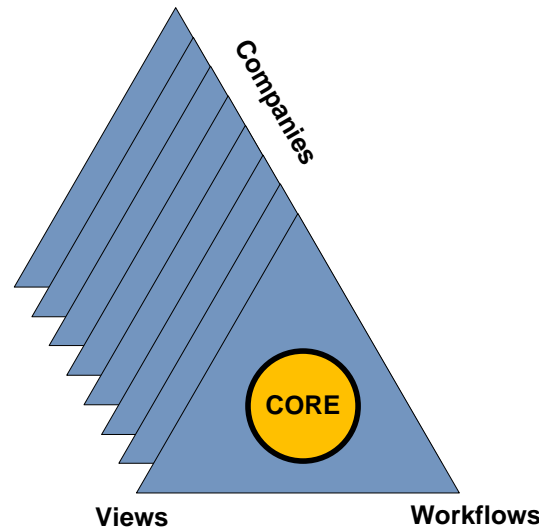


Fig. 5 Multi-company e-business system over single enterprise infrastructure

The core of this enterprise system is referred to as a core platform. To build up this core platform, in the first phase the workflows that are part of each company business need to be separated and segregated into two categories, namely generic and customized workflows. Subsequently, the core business data persistence logic that is embedded inside these workflows needs to be extracted and will become part of the core platform. During the next phase, the views that are part of each company’s workflow need to be separated and segregated into two categories, namely generic views and customized views. This separation of views, workflows and core business logic makes the system three dimensional, as shown in Fig. 5.

B. Evaluating SOAW2 Model Framework on the Core Platform

As a starting point we will evaluate the existing SOAW2 model framework. Fig. 6 illustrates the problem solution within the existing SOAW2 model framework. As discussed earlier, the SOAW2 framework does not support explicit request mediation. Due to this limitation, every company that runs its e-business on the system requires its own dedicated set of views with explicit references to the required services embedded inside them. Fig. 6 illustrates an instance where the users of Company 1 and Company 2 are using replica copies of the same generic views, while the users of Company 3 have their own set of customized views. As a result, the enterprise system does not offer any sharing of user interfaces. In addition, the system scalability is a major concern because each time a new company is registered on the system, the service provider has to allocate a dedicated set of views to this new company.

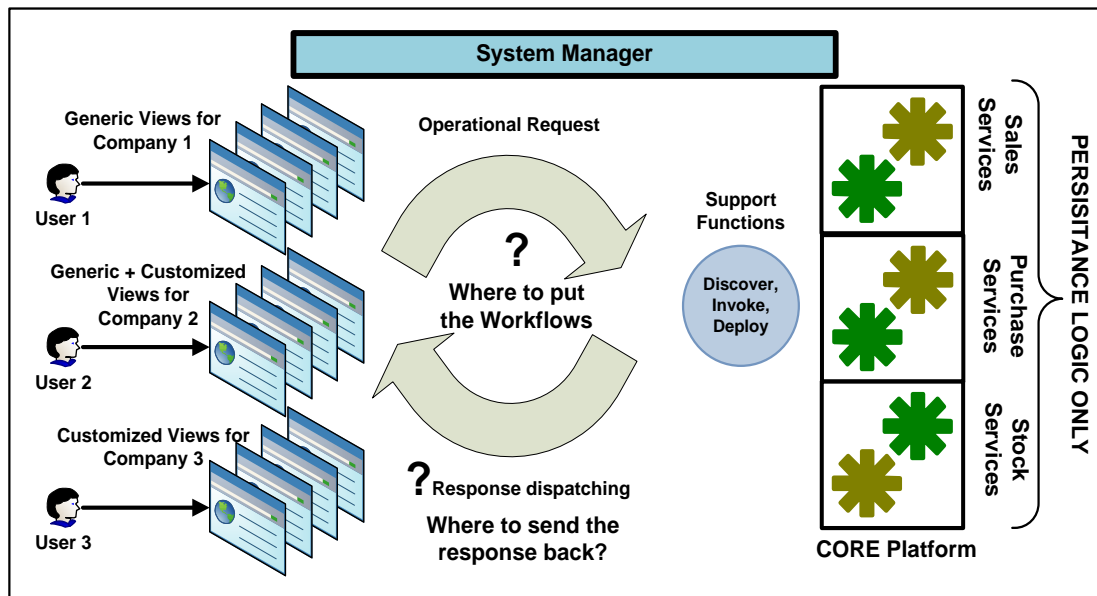


Fig. 6 SOAW2 framework solution

III. AN EXTENSION OF SOAW2

This section describes an enhanced Web 2.0 model framework called W2ASVB (Web 2.0 Architecture for Service and View Brokerage). The enhanced model framework is based on SOAW2 and contains two fundamental additions: the introduction of an effective request broker and the replacement of supporting functions with service adapters. Fig. 7 illustrates the enhanced W2ASVB model framework.

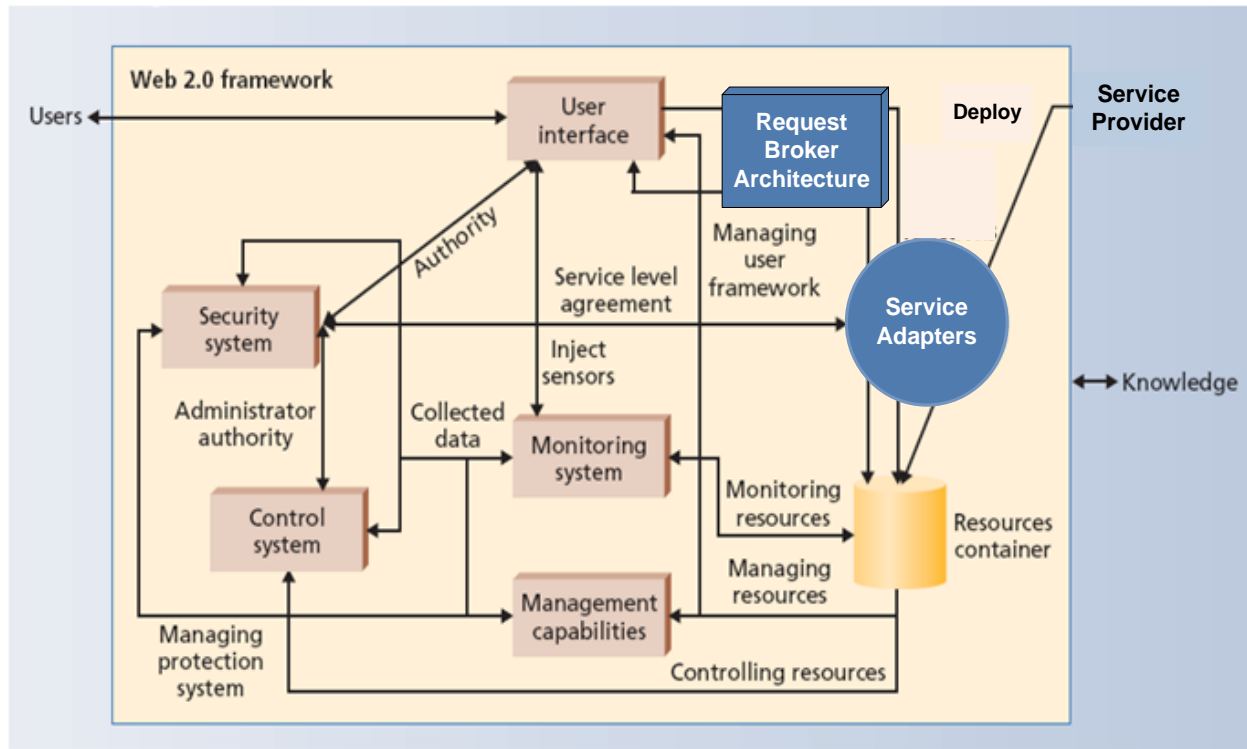


Fig. 7 W2ASVB - A proposed extension of SOAW2 framework

The new request broker architecture is an improved version of the application controller model proposed by Alur, Curpi, and Malks (2003). It contains improved request brokers that are responsible for action and view management as part of the W2ASVB model framework. They provide on-demand request routing between the user interfaces and the services of the core platform. This way, the knowledge layer of the model framework is shared between the service provider and the users. This means that the knowledge in the model will evolve through the experiences of both users and the service provider.

The second fundamental addition in the proposed W2ASVB model framework is the replacement of the supporting functions with service adapters which act as the backend interfaces of the core platform. The reasons for this are as follows:

1. The discover and invoke supporting functions are no longer needed in the new model since these are now the broker's responsibility.
2. Supporting functions such as deploy, invoke, and discover, which were useful in situations where one type of service is published in the resource container from more than one service provider and a user request needs to be resolved to the best matching service is now mapped by the broker and is no longer necessary as a separate function.

The service adapter is a new type of lightweight service that contains an implementation of a workflow. In the SOA tradition, services are relatively large, shared, intrinsically loosely coupled units of functionality, and have no embedded calls to each other. Web Services is a version of SOA that runs over the Internet and provides services to users transparent to their locations. But in our framework, the generic workflows must be shared among different retailers, which make the web services inappropriate technology. In our framework, the workflows are modeled as service adapters. As a user request is processed, the adapters connect to the core platform to execute the workflow implemented by the corresponding service.

The W2ASVB framework as shown in Fig. 8 has four main layers: presentation, request management, operational, and core service layers.

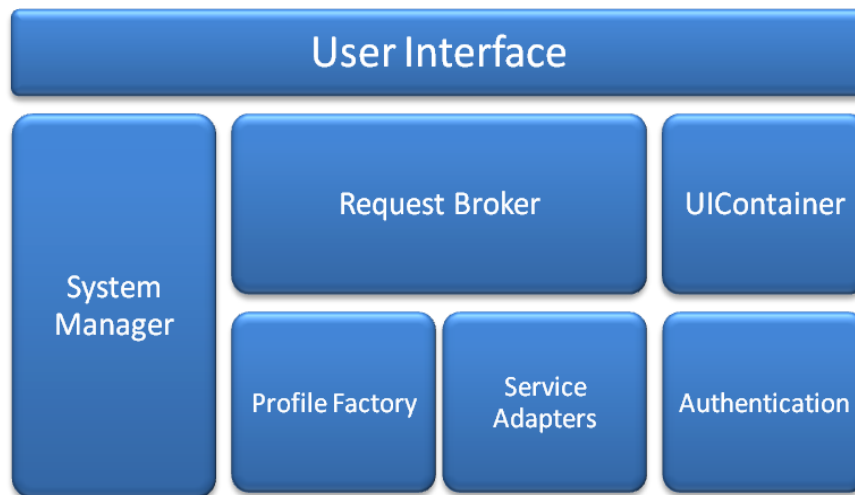


Fig. 8 Framework Layer Representations

The Presentation layer is the top-most layer and includes the user-interfaces. Internally this component is divided into two sub components: generic user interfaces and customized user interfaces. The generic user interface component holds the generic set of user interfaces shared between different e-Business systems on the same core platform. The customized user interface component holds a set of user interfaces that are customized by some of the systems; and these are not shared amongst the systems. The Request management layer consists of request brokers and UI containers. The request broker is responsible for brokering user requests (received via the presentation layer) to the service adapters, and then brokering the responses back to the users via the presentation layer. The User-interface container serves as a data container that holds the response data. The presence of the System manager component in this layer enables control and administration. The Operational layer consists of a Profile factory, Service adapters, and Authentication components. The Profile factory is responsible for holding the profiles of the client companies. The Authentication component is part of system security and provides assistance to the system manager in authenticating and authorizing users and their respective locations.

The final layer is the Core services layer which consists of the core platform components. This layer provides access to core business logic and persistence services to the components that exist in the operational layer, assisting them in the accomplishment of their functions. For example, at one end, it facilitates the service adapter in making business services-related calls for data storage and retrieval, while at the other end it assists the authentication component in validating user credentials, such as user id, password, and profile roles. In addition, it assists the profile factory component in retrieving company-related information from the database, such as location details, employee details, addresses, contact numbers, etc.

IV. BROKERING USER REQUESTS IN W2ASVB FRAMEWORK

As a key concept in the proposed W2ASVB model framework, request brokers are semi-autonomous objects equipped with all the capabilities required to handle action and view management for user requests. In other words, they can be considered agents that provide a mediation mechanism for the core platform services. They have been classified as semi-autonomous because they can complete user requests without assistance from other brokers, but are controlled by a managing authority, the System manager. When a user request is received, it is queued in a waiting area to be dispatched by the broker.

Fig. 9 illustrates the above principles, tracing the execution of the requests generated by three separate users. Both User 1 and User 2 use the generic interfaces; User 3 uses only the specific interfaces while User 2 uses both the generic and the specific interfaces. Request R4 is generated by the generic user interfaces and is mapped by the broker to the generic service adapter, while request R2 is generated by the customized user interface and is mapped to the customized service adapter instead.

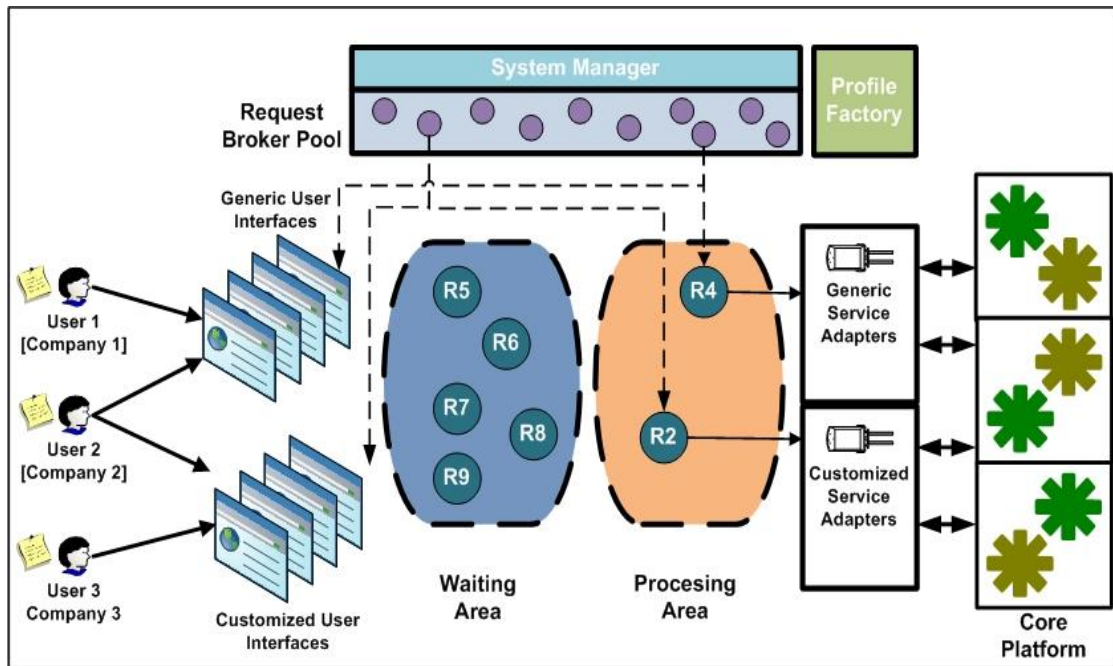


Fig. 9 W2ASVB in action – Example 1

The system manager continually checks the waiting area, and as soon as a request arrives, it allocates one of the request brokers to it. The allocated request broker transfers the request from the waiting area into a processing area, and begins analysis of the request header to ascertain the source details, such as the name of the user interface from where the request is being generated, and the action it requires. Once the source and required action are identified, the request broker initiates a search to match a suitable service adapter within the user session profile. The user session profile is a profile carried by the users, and is allocated to each user by the system manager when they first log in as illustrated in Fig. 10.

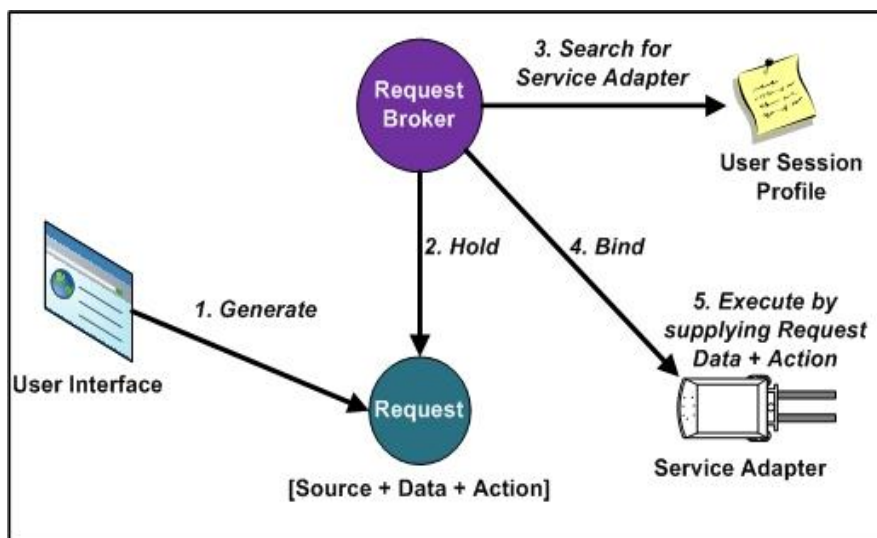


Fig. 10 Request brokering in action

On successful match, the request broker binds the relevant service adapter (either generic or customized) and executes the service with the data and the action part of the request. The services are independent of their usage scenario and can be employed by any user. At the same time, they can be conveniently implemented using a suitable component technology on the same application server since they are in fact part of the core infrastructure. This is one of the major features of the proposed W2ASVB model framework that promotes multi-company sharing of service adapters and allows implementation of the framework using entirely server-side software component technology. Fig. 11 illustrates the flow of control in response brokering.

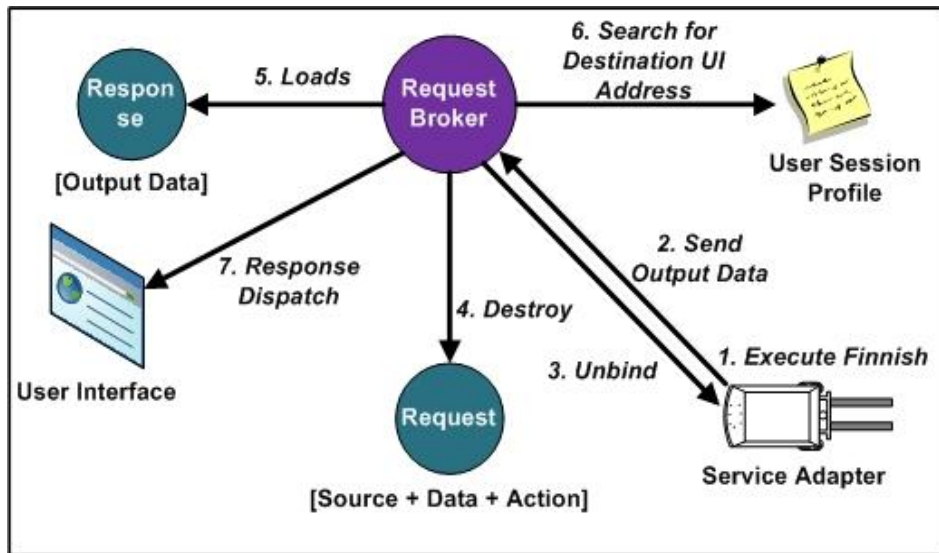


Fig. 11 Response brokering in action

On completion of the execution of the bound service adapter, the request broker unbinds the service adapter and loads the output data generated in the response to the execution of the request. It then searches the user session profile again to find the user interface destination reference. Using the matching interface, it dispatches the response data back to the user together with the output data generated. At the end of response brokering, the request broker also releases all the holding resources and makes itself available to the system manager to be allocated to another request.

The user interfaces that generate the request and receive the response data are independent of the company usage scenario; therefore any user can use them. This is a second major feature of the proposed W2ASVB model framework that promotes multi-company sharing of user interfaces (again, except customized versions because they are company-specific implementations and only shared by the users of that company).

Fig. 12 shows a second example that depicts a change in the mapping algorithm. In this case, request R5 (generated by the customized user interface) is mapped to a generic service adapter and request R6 (generated by the generic user interface) is mapped to a customized service adapter.

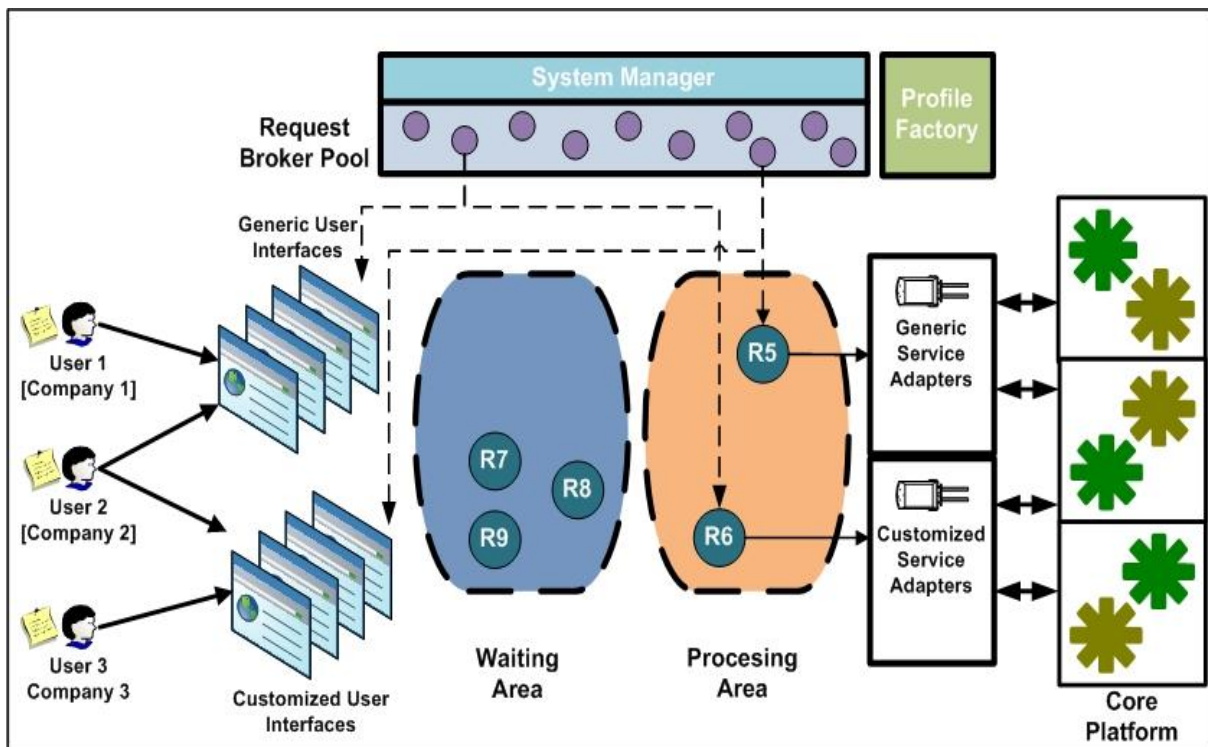


Fig. 12 W2ASVB in action – Example 2

The change in the broker mapping rules shown on Fig. 12 is not caused by the user session profile, since the users continue

using the same interfaces. It is rather determined dynamically by the broker's matching algorithm. The two examples illustrate the flexibility of the framework broker, which allows for much more fine-grained customization of the framework than SOAW2.

V. CONCLUSION

The paper addresses the issues of constructing an efficient framework for utilizing Web 2.0 component technology, which will facilitate the transformation of small retailers' businesses into e-Businesses. The proposed model framework is an extension of the well-known SOAW2 framework. It provides users with an option to customize their e-businesses according to their individual needs using additional profiling and customization. This is achieved by introducing an extra layer in the architecture that supports the brokering of both the user requests for services and the service responses generated in return.

The above model can be implemented after the SOA principles, but, unlike the SOAP-based SOA systems, it can leverage from the use of lightweight RESTful Web services [16] instead. This would allow the entire framework to be implemented as a one server-side application using enterprise technologies such as Java EE. This implementation can get further advantage from the use of JSON as a standard format for encapsulating request data [17] and NoSQL technologies [18] for managing persistent data directly by the services of the core infrastructure.

This research model is currently in the very early stages of development, which is why it has some limitations that require future research, such as a request management algorithm to control the request brokers' pool size at run-time. Taking into consideration the highly operational and business-centric nature of the system, there is a need for serious investigation of this issue. Existing data mining algorithms would be useful, but a careful review is required to develop an improved version that not only performs the run-time statistical calculations of incoming requests, but also uses its own knowledge base to determine the pool size. Implementing this framework by using standard web service orchestration can be further researched. In this case, the framework will inherit all the existing web service features, and these framework services can be used as external services.

REFERENCES

- [1] Steven Ashton, "Project ABS (Architecture for information Brokerage Service)." Retrieve from: <ftp://ftp.cordis.europa.eu/pub/infowin/docs/fr-206.pdf>, 2000.
- [2] Nick Watson, "Controlling the Business Growth Speed from Financial Angle of View: Using Cash Flow System Dynamics Model," *IEEE conference on First International Workshop on Database Technology and Applications*, vol. 1, pp. 610-613, 2009.
- [3] Dowd, J., "High Street Britain 2015. London: All Party Parliamentary Small Shops Group (APSSG)." Retrieve from: http://news.bbc.co.uk/1/shared/bsp/hi/pdfs/15_02_06_highstreet.pdf, 2006.
- [4] J. Simmonds, Y. Gan, M. Chechik, S. Nejati, B. O'Farrell, E. Litani, and J. Waterhouse, "Runtime Monitoring of Web Service Conversations," *IEEE Trans. Services Computing*, vol. 2, pp. 223-244, 2009.
- [5] Hinchcliffe, "Web 2.0 Continues Its Move to the Workplace," *The Enterprise 2.0 Conference*, Santa Clara, CA. Retrieve from: <http://www.zdnet.com/article/the-top-enterprise-web-2-0-stories-of-2007/>, 2007.
- [6] Omar, W., Abbas, A., and Bendiab, T., "SOAW2 for Managing the Web 2.0 Framework," *IEEE Journal on Computer Society*, vol. 9, pp. 30-35, 2007.
- [7] MerchantOS, "Introduction to MerchantOS." Retrieve from: <http://www.merchantos.com/try-it/2/>, 2007.
- [8] PHPPointOfSale, "Peoria Plumbing." Retrieve from: <http://demo.phppointofsale.com/index.php>, 2007.
- [9] ZohoCRM. Retrieve from: <http://crm.zoho.com/crm/ShowHomePage.do>, 2010.
- [10] Dorn, J., Rainer, A., and Hrastnik, P., "Toward Semantic Composition of Web Services with MOVE," *IEEE International Conference on Enterprise Computing*, vol. 3, pp. 67, 2009.
- [11] Hitachi, L., "Proposal of Application Architecture in Electronic Commerce Service between Companies," *International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems*, vol. 1, pp. 46, 1999.
- [12] XiaoQin, H., LinPeng, H., Lin, C., and Minglu, L., "Design and Implementation of an Agent-Based Web Services Platform for Electronic Commerce. Services Computing," *IEEE International Conference on Services Computing (SCC'04)*, pp. 643-646, 2004.
- [13] Howard, R. and Kerschberg, L., "A Knowledge-based Framework for Dynamic Semantic Web Services Brokering and Management. Database and Expert Systems Applications," *15th International Workshop on (DEXA'04)*, pp. 174-178, 2004.
- [14] Koerner, E., Aghoutane, N., Ercan, G., Strick, L., Wunderlich, W., Funabashi, M., et al., "Efficient Brokerage in a CORBA-Based Service Mediation Platform," *The Fourth International Symposium on Autonomous Decentralized Systems*, pp. 284, 1999.
- [15] Tothezan, I., Athanassiou, E., Alzon, P., and Karetos, G., "Enterprise Modelling of Information Brokerage and Retail Services," *1st International Enterprise Distributed Object Computing Conference (EDOC '97)*, pp. 13, 1997.
- [16] Fielding, Roy T.; Taylor, and Richard N., "Principled Design of the Modern Web Architecture," *ACM Transactions on Internet Technology (TOIT)*, New York: Association for Computing Machinery, vol. 2 (2), pp. 115-150, 2001.
- [17] ECMA International, "The JSON Data Interchange." Retrieve from: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>, 3rd ed., 2013.
- [18] Ryan Betts, "How JSON sparked NoSQL," *InfoWorld*. Retrieve from: <http://www.infoworld.com/article/2608293/nosql/how-json-sparked-nosql-and-will-return-to-the-rdbms-fold.html?page=2>, 2014.



Prof. Karim Ouazzane is currently a full professor of computing and knowledge exchange at London Metropolitan University. He is deputy director of Intelligent Systems Research Centre and Knowledge Transfer Partnership Director at London Metropolitan University. He carried out research in collaboration with industry through a number of research schemes such as EPSRC, KTP, EU Tempus. Karim published around 70 papers, 3 chapters in books mainly in the area of AI and in visual speech recognition, and he is the author of 3 patents. Some of his work was nominated for awards such as Bafta and Radar ones and was recently commended by the UK Queen in recognition of the novel Intelligent Keyboard, a project he carried out as part of KTP programme in partnership with Essex Disability.



Dr. Sajid Afzal is part time lecturer in London Metropolitan University. Dr. Afzal received his PhD from London Metropolitan University, London, UK in 2013.



Dr. Vassil Vassilev is a senior Lecturer at London Metropolitan University and Member of British Computer Society. Reader in Computer Science and former Project Manager at General Electric Capital Services with interests and publications in Semantic Web, Web Services and Web Analytics.



Kaushik C. Patel is the Director and the founder of ASK electronics with his two brothers in 1983. He oversees the multi-branded store in central London, online operations as well as five retail outlets across two dedicated brands, Toshiba and Sony. Patel is a partner of London Metropolitan University who worked on developing a CRM system which had a great impact on his e-business activities.



Dr. Althaff Mohideen is a lead investigator in internet engineering with University of Aberdeen. He was formerly attached to the University of Westminster as lecturer. Dr. Mohideen received his PhD from Nagaoka University of Technology, Japan.