AN E-BUSINESS FRAMEWORK DESIGN USING ENHANCED WEB 2.0 TECHNOLOGY

Karim Ouazzane, Muhammad Sajid Afzal, Hassan Kazemian
London Metropolitan University, 166-220 Holloway Road, London, U.K.

Jun Li
Department of Land Economy, University of Cambridge, 19 Silver Street, Cambridge, U.K.

Keywords: Request brokering, Service adaptors, Workflow management, Customized business logic, e-business.

Abstract: The main aim of this research is to develop a robust, reliable, efficient and novel framework by using Web 2.0 technology that will serve as a front and middleware collaboration model between data persistence logic and operational requests. This framework will serve as a mediation platform for request brokers. It will provide a high level of abstraction by encapsulating low level details of the system, such as request handling, request mediation, response handling and service loading. In order to overcome the hard coded service mapping with interface, there are no customizable business logic and no generic customized workflows problems. These are the most essential requirements of converting the Small and Medium Enterprises into one e-business platform swiftly.

1 INTRODUCTION

In the current era of state-of-the-art cutting edge technologies, business organizations are rushing to transform their companies into e-business. Presence of e-business brightens their chances of winning market shares by maximizing product availability and providing related services at convenient and cheap way to their customers.

However in this race, the e-business growth pendulum is only swinging one way. This is very easy to understand by observing today's business market. Due to a limited budget, small business organizations (e.g. local retail shops) cannot afford costly IT systems and associated maintenance/administration costs. But despite these financial constraints, they desperately need computation to survive and to compete with giant competitors by expanding their businesses.

Web 2.0 is a second generation design patterns and business model for web applications. The terms first coined by (O'Reilly, 2005). Every technology requires a clear and effective model for representing its components and the interaction between these components. Until recently, there has been no innovative Web 2.0 model which performs

- On-demand customization of business logic and user interfaces.
- Request management component.
- Dynamic operational request mapping with the interface.
- Encapsulation of standard business logic.

Unfortunately not much has been done by the research community in building a clear understanding of Web 2.0 and its application paradigm. Every technology requires a clear and effective model for representing its components and the interaction between these components. Until recently, there exists no such proper Web 2.0 framework with the features described above. As a result, a research conducted by (Omar, Abbas and Bendiab, 2007) proposed the model framework for defining Web 2.0 components and their relations as illustrated in Figure 1.

One of the major deficiency of the Web 2.0 model framework is none presence of request management mechanism. According to a research conducted by (Alur, Curpi and Malks , 2001) , request management

Ouazzane K., Afzal M., Kazemian H. and Li J..
AN E-BUSINESS FRAMEWORK DESIGN USING ENHANCED WEB 2.0 TECHNOLOGY.
DOI: 10.5220/0003417800970102
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 97-102
ISBN: 978-989-8425-56-0

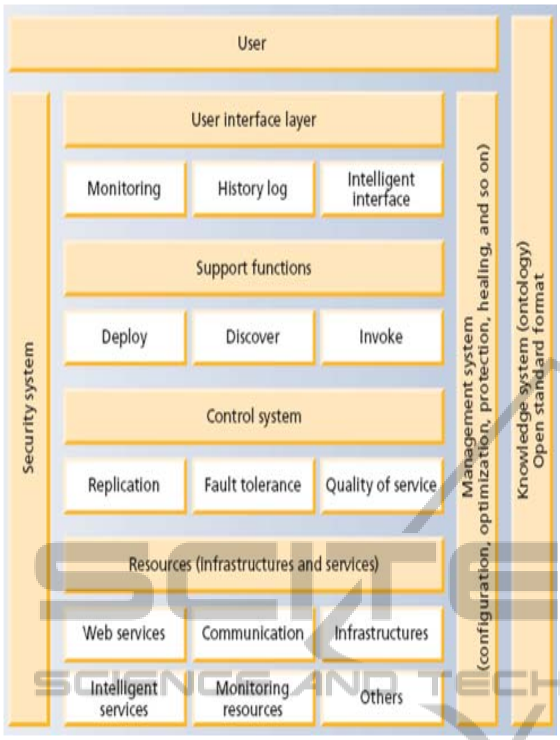

Figure 1: Model Framework by Omar, Abbas, & Bendiab (Omar, Abbas and Bendiab, 2007).

in web-based systems is a two step activity - Action Management and View Management.

There are only two types of request management techniques that have been discovered so far. One is called 'conventional' and another is called 'request brokerage technique'.

Conventional request management technique is a technique that is commonly used in traditional web-based systems. In a conventional technique, request management is automatically performed by the application server.

This concept of request management mechanism on the basis of request broker's based system architecture becomes the centre of attraction for relevant research communities and still gaining popularity. Request brokerage in service oriented architectures is an active research area these days, such as in ACTS/ABS (Architecture for information brokerage service) (ACTS, 2000) and have managed to propose some stable models in which a request can be dispatched to a best suited service(s), transparent to a user, with the help of request brokers.

Web Services are a new emerging web programming paradigm based on the concept of Service-Oriented Architecture (SOA) (Bell and Michael, 2010). (Howard and Kerschberg, 2004) proposed a complete framework called Knowledge-based Dynamic Semantic Web Services (KDSWS). Another research attempted by (Beck, et al, 2000) considers the idea of next generation electronic brokerage for performing active and real-time functionalities. Request broker model proposed by (XiaoQin, LinPeng, and Minglu , 2004) is a concept of agent based web-services platform request brokers model which is being used in this research and is actually an extension of architecture for information brokerage service models proposed in ACTS/ABS.

A research conducted by (Zhao and Tong, 2007) has proposed a service composition model called 'A Dynamic Service Composition Model Based on Constraints'. This model is not a request broker based model, but is capable of handling complex situations. Problems of Web2.0 framework without request broker are illustrated in Figure 2.

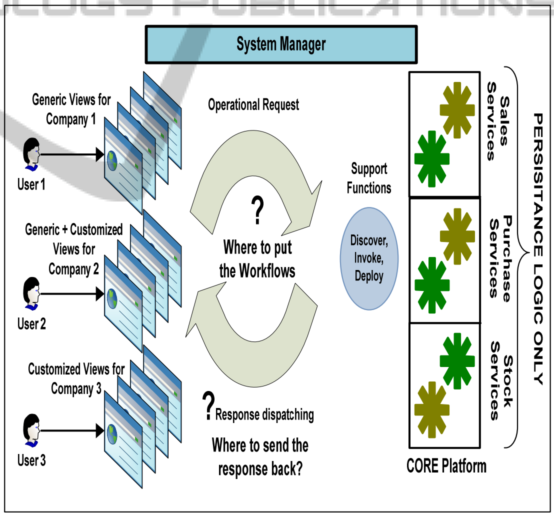

Figure 2: Application model framework without request broker.

A review of the current Web 2.0 retail systems that are built upon existing SOAW2 model framework has shown the lack of fundamental functionalities such as on-demand customization of business logic and user interfaces. A system like ZohoCRM has shown some capabilities of user interface customization but it is only limited to the customization of business reports (A.H.M. van Limburg, 2010). Furthermore, another important observation is that none of the current Web2.0 retail systems is capable of providing on-demand customization of business logic. This is because of the fact that applied SOAW2 model framework does not support explicit request management. Current SOAW2 model framework directly exposes user interfaces to business services that reside inside resource container. Also, it is crucial to note that the presence of supporting function between user interfaces and resource container should be ignored as they don't perform any special tasks with the exception of searching & binding of best suited services. This direct exposure led to a result in the form of user interfaces that contain concrete business services mapping instructions. Due to this fundamental shortcoming of the framework, developed systems do not provide on-demand customization of business logic at run-time. Moreover the framework does not provide any extra layer to secure this customized business logic. Therefore, an investigation is required to propose a new Web 2.0 framework that will overcome the current SOAW2 fundamental shortcomings (D. Gallula, 2009; K. Ducatel, et al, 2010).

2 THE PROPOSED FRAMEWORK

2.1 Fundamental Architecture

The proposed model framework is based on SOAW2 and contains fundamental changes in existing models framework. The fundamental changes include introduction of effective and intelligent request broker architecture and the replacement of supporting functions with service adapters, as shown in Figure 3.

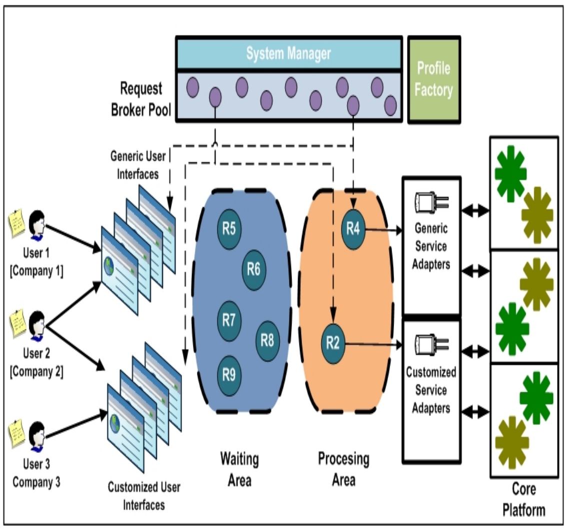

Figure 3: The proposed model framework.

By taking into consideration the shortcomings of request management component in Web2.0 framework and a need of effective and intelligent request brokerage mechanism to handle complex on-demand sharing and customization problem, a new architecture of the framework is proposed which overcomes the inadequacies of the existing Web2.0 framework. The proposed model framework integrates request broker architecture in between user interface and resource container. This will increase the action and view management of the model framework. It might also provide on-demand request routing between user interfaces and services of standard platform. The second important component in this proposed model framework is a replacement of supporting functions with service adapters. This component could avoid any conflict with SOA principles. Also, in this architecture workflows are modeled as service adapters.

The supporting functions in the new model framework are replaced with service adapters. Service adapter is a new concept of light weight services and contains implementation of workflows. In SOA tradition, services are relatively large, shared, intrinsically loosely coupled units of functionalities, and have no embedded calls to each other.

It is debateable that proposed model framework requires sharing of generic workflows among different companies (i.e. users) then why can't generic workflow be modelled as web services. The simple answer of this question is that, in proposed framework, workflows are actually sequential calls to the services of core platform. Therefore, if workflows get modelled as web services, it will be a violation of loosely coupled and no embedded calls principle of SOA.

To overcome this limitation and to avoid any conflict with SOA principles, all Workflows including generic and customized are modeled as service adapters. On a user processing request, these adapters are connected with the core platform to execute the modeled workflows.

2.2 A Layered Representation of the Framework

Layered representation of the framework shown above is divided into four layers namely presentation, request management, operational and core service layers.

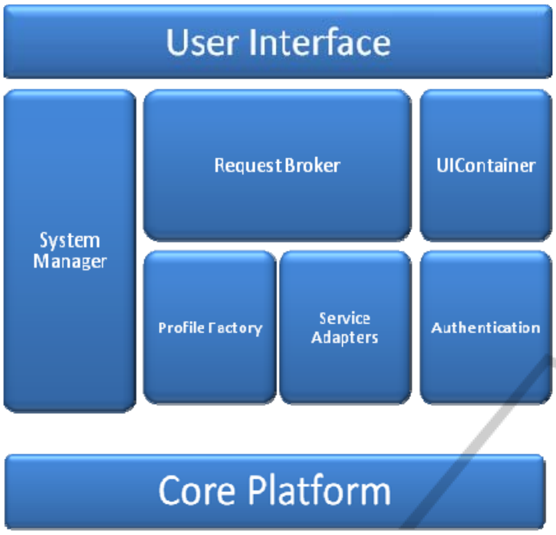

Figure 4: Framework Layered Representation.

2.2.1 Presentation Layer

Presentation layer comes at the top and it consists of a user interface component that is internally divided into two sub components – generic user interfaces and customized user interfaces; Generic user interface component holds the generic set of user interfaces. These user interfaces shared across the interested companies. Customized user interface component holds the set of user interfaces that are customized for some companies. Presentation layer of the system is directly exposed to users thus serves as a gateway to the system. The users use it for sending data processing requests to the system.

2.2.2 Request Management Layer

Request management layer consists of request brokers and UI container components. Presence of the system manager component on this layer indicates a control and administration. Request broker component on this layer is responsible for brokering user's requests (received via presentation layer) to service adapters and then brokering back the responses to the users via presentation layer. UI container serves as a data storage that holds the response data.

2.2.3 Operational Layer

Operational layer consists of profile factory, service adapters and authentication components. Profile factory component is responsible for holding the profiles of the client companies. Authentication component is part of a system security and provides assistance to the system manager in authenticating and authorizing users and their respective locations.

2.2.4 Core Service Layer

This layer is core services layer and it consists of core platform components. It provides the access of core business and persistence services to the components that exist within operational layer. Moreover, it assists operational layer components in the accomplishment of their required functionalities. For example, on one end, it facilitates service adapter in making business services related calls for data storage and retrieval, while on the other end it assists authentication component in validating the user credentials such as user ID, password & roles. In addition, it also provides service to the profile factory component for retrieval of company related information from database such as location details, employee details, addresses and contacts numbers.

3 HYPOTHETICAL MODEL IN ACTION

Any user request for data processing can be divided into two sections, data part and action part. Data part contains the *data* that requires processing. Whereas the action part describes the required operation on a given data. On arrival, a user request gets queued up in a waiting area. The system manager consistently checks the waiting area and as soon as a request arrives, it will be allocated to one request broker. This allocated request broker moves the request from waiting area into processing area and starts analysis of a request header to find out the source details, such as, the name of user interface from which the request is being generated and the action it requires. On identification of source and action, it starts searching to find out the name of matching service adapter in a user session profile. User session profile is a profile carried by each user and it is initially allocated to them by the system manager when they first log in.

On successful match, request broker binds the relevant service adapter (i.e. either generic or customized) and executes it by providing data and action part of the request. This binding of service adapter is called 'request brokering'. The service adapter only requires data and action to be executed. Hence, they are services that are independent of their usage scenario and can be used by any user. This is one of the main feature of the proposed Web2.0

Architecture for Service and View Brokerage (W2ASVB) model framework and it promotes the idea of multi-company sharing service adapters with the exception of customized ones as they are company-specific implementations and hence confidential. Figure 5 illustrates the flow of control in request brokering mechanism.

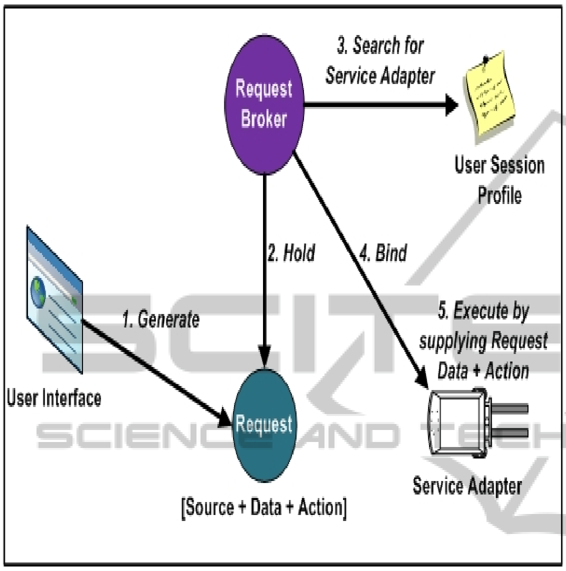

Figure 5: Request brokering in action.

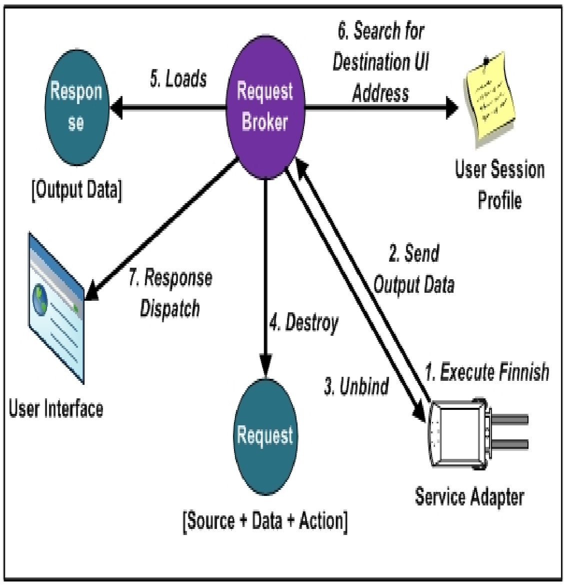

Figure 6: Response brokering in action.

On completion of execution of bonded service adapter, request broker unbinds the service adapter and loads the output data (if any) which are being generated as a response of the execution of response object. It then starts searching again the user session profile to find the destination user interface address. On successful match, it dispatches the response data back to the user along with output data. This unbinding of service adapter and mediation of response back to the user is called ‘response brokering’, as shown in Figure 6. Since the user interfaces that generates request and receives response data are independent of company usage scenario, they can therefore be used by any user. This is another major feature of the proposed model framework and it promotes the novel idea of multi-companies sharing user interfaces. Figure 6 illustrates the flow of control in response brokering mechanism. At the end of response brokering, request broker releases all the holding resources and makes itself available to the system manager to be allocated to another request.

Figures 5 and 6 illustrate four examples of request mappings. In the figures it is clear that user 1 of company 1 uses generic user interfaces only. Whereas user 2 of company 2 uses some generic and some customized user interfaces; finally user 3 of company 3 uses customized user interfaces only. Figure 5 depicts mapping of request R4 (i.e. generated by generic user interfaces) to generic service adapter and request R2 (i.e. generated by customized user interface) to customized service adapter. On the other hand, Figure 6 depicts changes in the scenario where request R5 (i.e. generated by customized user interface) is mapped to generic service adapter and request R6 (i.e. generated by generic user interface) is mapped to customized service adapter. However, the mapping shown in the examples above are not the boundaries of the framework, it is also capable of supporting other possible mappings which reflects its universality.

4 CONCLUSIONS

The investigation started with a question of building up a model framework by using Web 2.0 technology, which enables small retailer to achieve e-business transformation. The model framework provides them with an option to customize their e-businesses according to their individual needs. With reference to customization and intelligent request management, it can be concluded that the designed system with the application of the new proposed model framework has proven its capabilities of handling such complex situations.

Initial version of proposed framework architecture is presented along with the problem description; it is evident that the framework requires an effective model that serves as a front & middleware collaboration. This model framework provides an intelligent feature, that is interfacing for sharing core business logic among all small businesses as well as providing customization facility without affecting other businesses.

For future work it is worthwhile exploring the need of demand-and-supply algorithm to control the request brokers' pool size at run-time. Given the highly operational and business-centric nature of the system, the authors recommend the use of current data mining algorithms to achieve an improved version that not only performs the run-time statistical calculation of incoming requests but also uses its own knowledge base to decide on the pool size.

REFERENCES

ACTS Project (2000). *Project ABS (Architecture for information Brokerage Service) Final Report*. Avaliable:ftp://ftp.cordis.europa.eu/pub/infowin/docs/fr-206.pdf.

A.H.M. van Limburg, (2010), *Towards innovative business modeling for sustainable eHealth applications*, Second International Conference on eHealth, Telemedicine, and Social Medicine

Alur, D., Curpi, J., & Malks, D (2001). *Core J2EE Patterns: Best Practices and Design Strategies*. USA: Sun Microsoft Press. 60-65 .

Beck, et al (2000). *Active and Real-Time Functionalities for Electronic Brokerage Design*, International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems ,

Bell and Michael (2010). *SOA Modeling Patterns for Service-Oriented Discovery and Analysis*. UK: Wiley & Sons. 390

D. Gallula (2009), *Enriching the E-learning experience in the framework of Web 2.0 using Usability 2.0 (in Hebrew),* Dept. of Information Sciences, Bar-Ilan University, Ramat-Gan, Israel.

Howard, R., & Kerschberg, L. (2004). *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.

K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.C. Burgelman (2010), *Scenarios for Ambient Intelligence in 2010,*Available:ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf. Last accessed 10th Jan 2011

O'Reilly, T. (2005). *What is Web 2.0?* . Available: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Last accessed 4th Dec 2010

Omar, W., Abbas, A., & Bendiab, T. . (2007). *SOAW2 for Managing the Web 2.0 Framework*. IT Professional - IEEE,. 1 (1), 30-35 .

Rezaiian Fardoii, S., Monfared, M.A.S. (2008), *Design of an Intelligent Customer Relationship System Using Evidential Reasoning Approach*, Int. Conf ICDM , IMT Ghaziabad, India,

XiaoQin, Lin Peng and Minglu, (2004) , *Design and Implementation of an Agent-Based Web Services Platform for Electronic Commerce. Services Computing*, IEEE International Conference on (SCC'04), pp. 643-646

Zhao, H., and Tong, H. (2007). *A Dynamic Service Composition Model Based on Constraints*. Sixth International Conference on Grid and Cooperative Computing (GCC 2007) , pp. 659-662 .