SEWOS: A FRAMEWORK FOR SEMANTIC WEB OPERATING SYSTEM.

A. M. Riad
Head of information system dep.
Faculty of computers and information sciences.
Mansoura university, Egypt
amriad2000@yahoo.com

Hamdy K. Elminir
Head of communication dep
Misr Academy for Engineering & technology.
Hamdy_elminir@hotmail.com

Mohamed Abu ElSoud
Computer Science Dept.
Faculty of computers and information sciences.
Mansoura university, Egypt
mohamed_hossieny@yahoo.com

Sahar. F. Sabbeh
Assistant Lecturer
Alzarka Higher institute for administration & computer sciences
Damietta, Egypt
M_sabbeh@yahoo.com

ABSTRACT-- The core of our work is to present a SOA-based framework of a web operating system that makes uses of web semantics in its underlying file system. "SEWOS", as it is named, moves user's traditional operational desktop environment into a web browser accessed by user any time anywhere. The usage of semantic web techniques provides better organization, annotation, navigation and searching capabilities. Usage data, user's personalized settings along with file system ontology are all used to provide user a desktop metaphor and an experience that almost mimics traditional desktop.

Index Terms: Web Operating System, Semantic, Ontology, Service Oriented Architecture

1. INTRODUCTION

The tremendous growth in web as a source of information led to an increased interest in research of applications and trends that make use of web data trying to reach their ultimate potential. In this context, researchers investigated the ability to extend the functionality of traditional web-based applications to enable users to interact with applications in much the same way as they do with desktop applications. Web operating systems were developed to provide users with an environment that pretty much resembles traditional desktop environment through web browser. They represent an advance in web utilities as they aim to provide better operational environments by moving users' working environment within web site including managing his/her files, install his own applications. But, yet the bulk of the available systems disregarded the underling file system and the employment of semantic web techniques that are used to provide a well defined structure of resources as well as easier information retrieval and indexing.

Web operating system can be defined as a virtual desktop on the web, accessible via a browser as an interface designed to look like traditional operating system with multiple integrated built-in applications that allow the user to easily manage and organize his data from any location [1]. Web operating system provides users with traditional operating system applications as services available for user to access transparently without any prior knowledge about where service is available, the cost or constraints [2]. In web operating system, applications, data files, configurations, settings and access privileges reside remotely over network as services accessed by web browser

which is used for input and display purposes [3].

As mentioned earlier, web Operating system has become an interesting field of research and many attempts have been made for building web operating systems. WOS [2-8], the first known web-based operating system that provided a platform that enabled user to benefit from computational potential of the web. WOS provided users with plenty of tools through using a virtual desktop using the notion of distributed computing by replicating its services between multiple interacting nodes to manipulate user requests. WOS consists of three major components, graphical user interface, resource control unit which processes user request and finally a remote resource control unit which manages requests passed from other nodes.

The interest in web operating systems and their applications on academic communities resulted in VNet which was developed at the university of Houston and considered an access point to campus resources. VNet included variety of services that support students such as Desktop, admin management, contact management, file management services, calendar and scheduling services, report generation services, … etc [9].

based on the earlier work of WOS WEBRES was developed. WEBRES investigated the aspects of resource sharing that wasn't addresses in WOS and presented the notion of resource set which makes resources persistent rather that bounded to a specific user[10].

G.H.O.S.T (http://g.ho.st/vc.html), EyeOS "www.eyeos.com" and DesktopTwo "www.desktoptwo.com " are examples of systems that were built based on the trends of web operating systems. They mimic the look, feel and functionality of the desktop environment of an operating system. Moreover, they present variety of applications such as: File management, Address book, Calendar and text editing applications.

As mentioned earlier, taking semantics into consideration is an ongoing research field. Many attempts have been made to integrate semantic techniques to enhance traditional desktop environments [11-13].

Through annotating user files and resources, traditional desktop functionalities can be enhanced enabling retrieval of resources that are semantically related. HyperSD [14], a semantic desktop browser that comes with wrappers that allow importing regular desktop objects, such as files, person records, calendar entries, etc… into the semantic desktop accessed through a web browser.

Likewise, semantic techniques were used to provide better services by web operating systems. In this context, many attempts were made, DBin [15,16], a platform that integrates Semantic Web technologies with specific task modules and integrates a desktop service as user interface to build "Semantic Web communities". DBin accommodates a number of experimental modules to deal with specific kind of metadata (i.e. audio metadata extraction and desktop integration..etc).

The work in [17] integrated semantic web techniques towards building a social webtop which is the Web-based

counterpart to a desktop system on a personal computer (PC), a social webtop uses the Web for data storage and provides specific applications and for moving data process applications onto the Web.

Traditional system development process includes different parts that are strongly dependent of each other. The change of one function requires the whole system to be rebuilt which is expensive and time consuming causing many necessary process improvements never to be done. With SOA the services are encapsulated to make them autonomous where services can be easily changed, deleted, or added without influencing the whole system [18-21].

Seeing that SOA enhances the process of web development and since semantic web techniques provides better organization, navigation, searching and retrieval, in developing our framework we could never disregard these two ongoing techniques as we believe that it will enhance web operating system functionality. In our work we present the merge of semantic web and web operating system under the term semantic web operating system presenting SOA-based framework. The proposed system uses ontology for annotating user resources as well as employing semantic web techniques for enhancing user personal experience.

This paper is organized as follows; the next section presents SEWOS architecture. In section 3 a detailed description of the proposed architecture is provided. Section 4 outlines personalization services provided for our system. Our conclusion and future work is presented in section 5.

2. THE PROPOSED ARCHITECTURE

As stated earlier, Web operating systems have the features and functionality of traditional desktop operating system. However, Web operating systems typically transfer applications to web server where user can manage his resources through virtual desktop using web browser. At the start of our research we had three main interests which we tried to satisfy.

1- Moving from fully personalized familiar desktop on PC to a virtual remote desktop, is a hard task, as users will accept nothing less than traditional desktop which they have been accustomed to. Thus, user data, preferences as well as sessions must be maintained ensuring that user will always has a personal experience that resembles his fully personalized traditional pc environment.
2- Semantic web technology plays a significant role in today's web as well as desktop systems [22-23]. That's why we thought that it was only a matter of time before semantic web techniques thrust in the research of web operating system.
3- A service-oriented architecture (SOA) is seen as the next evolutionary step in building web-based applications as it provides a set of principles of governing concepts used during phases of systems development. As in n-tier architectures SOA separates

presentation/applications, services and data into layers preventing dependency between layers.

In our work, we tried to merge the semantic web with web operating system utilizing the notion of SOA to support our architecture.

2.1 SEWOS ARCHITECTURE

SEWOS is SOA-based architecture that shows the underlying semantic file system of our semantic web operating system. SEWOS consists mainly of three layers, application layer, service layer and data layer as depicted in Figure 1. SEWOS architecture Application layer contains both user interface (portal) and application manager which in turn includes set of applications: file manager, word processing, spread sheets, web search and to-do list. These applications will be covered more details later in the next section.

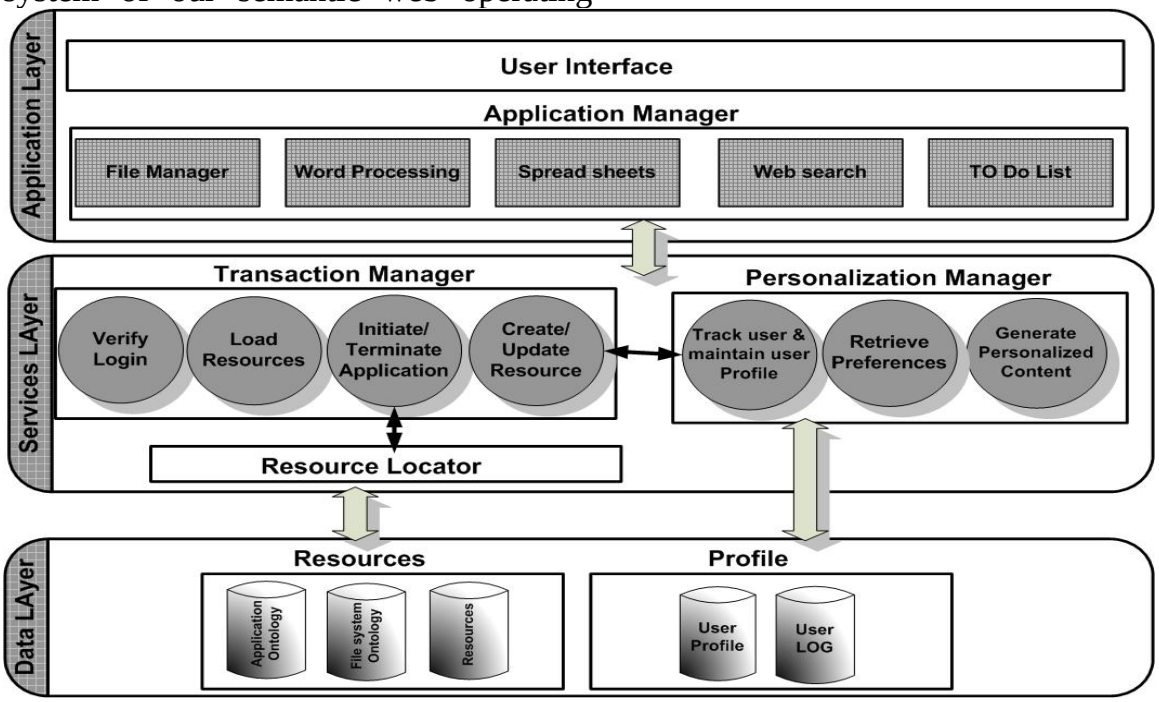

Fig.1 SEWOS Architecture

The second layer is service layer which includes transaction manager as well as personalization manager. Transaction manager controls user requests and works in correlation with both application layer and data layer in order to provide a virtual desktop. Personalization manager is responsible for generating a personalized desktop making use of user log, preferences and profile. Resource locator is used to locate where resources reside. Our architecture uses the notion of hyprid systems as it maintains a centralized resource location whereas resources themselves are decentralized. And finally, data layer contains back end databases that stores user profile, log file as well as user resources that are typical user files annotated using ontology.

3. SEWOS DESCRIPTION

SEWOS consists mainly of application layer which includes user interface as well as application manager, service layer which consists of transaction manager, application manager and resource locator, and the final layer is data layer which contains resource manager.

Whenever user Logs into system using portal, application layer transparently passes request to transaction manager which first authenticates and authorizes user, then work together with personalization manager to regenerate user's personalized desktop. User's desktop will include his layout, resources, bookmarks, applications as well as work copies which system maintains in case of application or connection failure.

Whenever user requests a resource, application layer passes request to transaction manager which initializes the corresponding application as well as requesting the associated resource from resource locator which works together with resource manager to locate and retrieve resource. On each request, personalization manager together with transaction manager maintains user log, profile and user session in order to provide personalization services.

The whole processes that are performed by SEWOS are shown in Fig.2 that shows system process diagram.

3.1 APPLICATION LAYER

This layer contains user interface (portal) as well as a set of applications namely: file manager, word processing, spread sheets web search and to-do list.

This layer provides an interface to user's files through accessible file system. This underlying file system employs web semantics.

As a result to our realization that semantic web's success depends on the availability of web pages annotated with metadata. These metadata are relevant keywords associated with or assigned to a piece of information describing the item.

In order for the underlying file system to use web semantics a set of rules has to be considered:

1- Every file is a resource: items (email, documents, images, etc) are converted into resources.
2- All resources are identified using URI.
3- Each resource is annotated based on the underlying ontology. This annotation makes better indexing and retrieval of resources.
SEWOS underling semantic file system uses file system ontology for providing annotation besides application ontology for application specific data. A sample of the proposed ontology is shown in Fig.3.

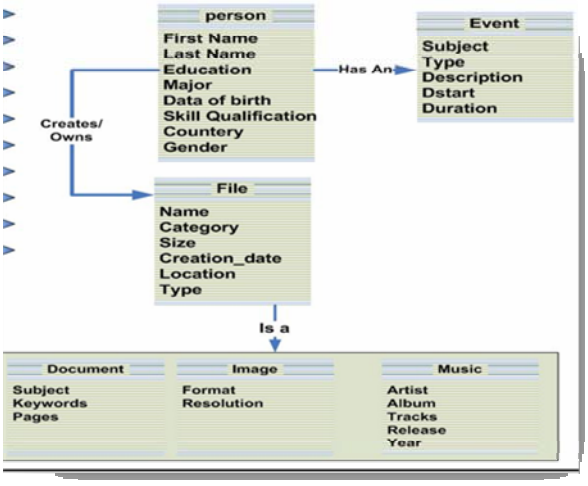

Fig.3. The used Ontology

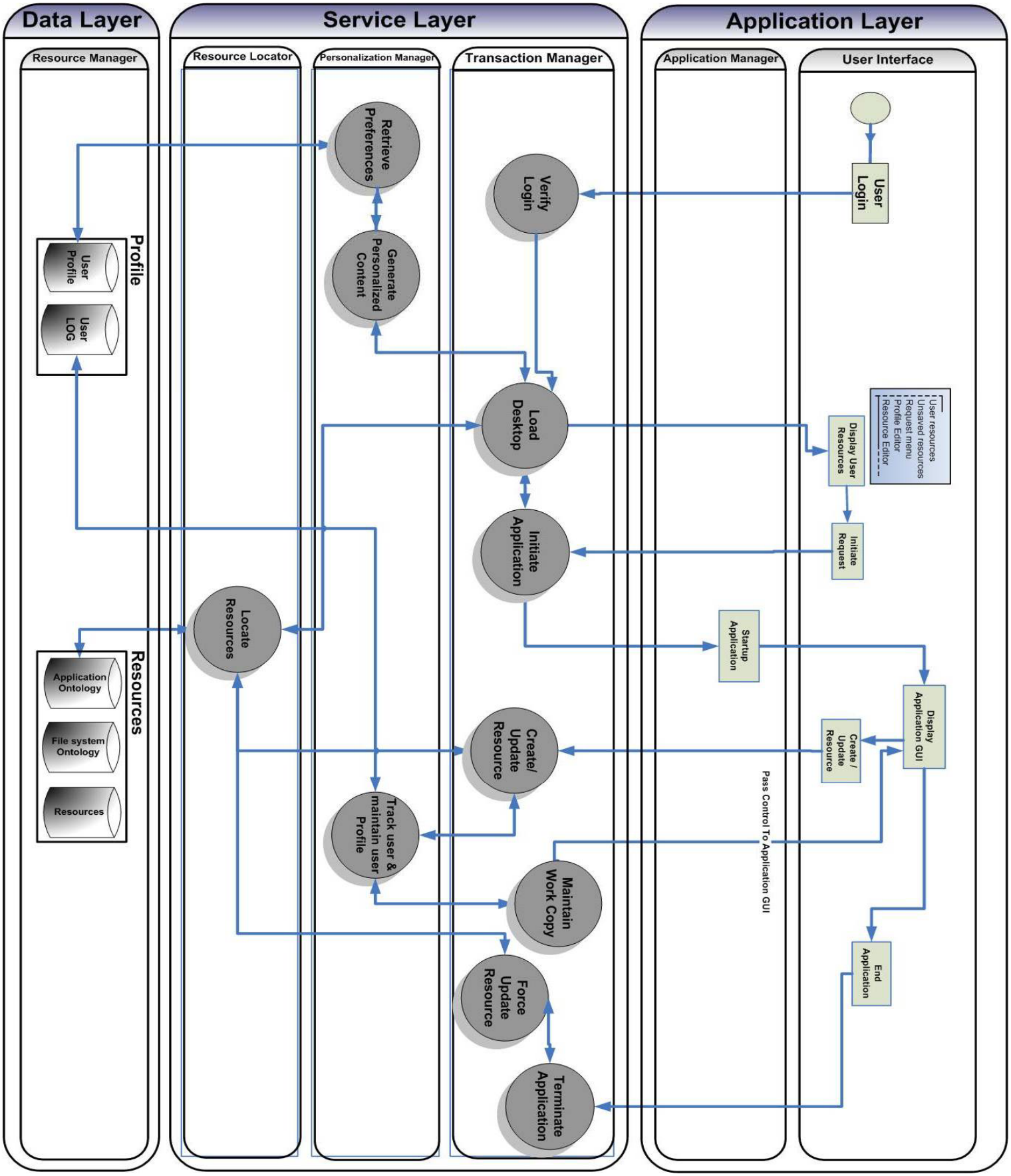

Fig.2 SEWOS Process diagram

3.2 SERVICE LAYER

The second layer service layer: includes transaction manager as well as personalization manager.

3.2.1 TRANSACTION MANAGER

Transaction manager manages user requests and works as a controller that coordinates the work between application and data layer. Transaction manager is responsible mainly for:

1- Managing user authenticating and authorizing.
2- Initiating personalization services for generating personalized desktop.
3- Initializing and terminating applications.
4- Working together with resource locator to create, update or load resources.
5- Transaction manager as well adds transactional capabilities that maintains user resources and keeps track of unsaved changes and forces update in case of application termination by user or even saves a work copy of resource in case of connection failure or application error. These transactional capabilities are new features that are added by our proposed system.

3.2.2 PERSONALIZATION MANAGER

While developing SEWOS framework, user satisfaction had to be always in mind. As previously stated, user will accept only a system that mimics what he has been used to. Thus SEWOS had to be a tailored operating system that we present to user virtually through web browser. This tailoring of web application based on user's needs and preferences is best served by employing personalization services. Personalization manager is responsible for maintaining user log, profile and preferences so that personalized content, layout as well as structure can be presented on user basis. Personalization process is discussed in details in section 4.

3.2.2 RESOURCE LOCATOR

Due to the nature of web today, web resources may be scattered across several servers and locations. This decentralization of resources necessitated the existence of a resource locator. Resource locator is responsible for locating resources. However, our architecture tries to overcome the everlasting problem of delay and network traffic in distributed systems by using the notion of hybrid systems. Resource locator maintains a centralized record of resource locations whereas resources are decentralized.

Whenever Transaction manager sends a request for a resource to resource locator, it determines resource location using the centralized resource location index, then works together with resource manager which retrieves the required resource and sends it back to the transaction manager.

3.3 DATA LAYER

System data layer (Backend) contains user profile, user log file (Personal usage data), annotated resources And finally, file system ontology and application ontology:

1- **User resources:** as each user directory will consist of user's annotated resources.
2- **Ontology:** file system ontology as well as application ontology.
3- **User profile**: contains user's preferences and personalized settings.
4- **User log:** keeps track of user's usage data, sessions and data storage.
5- Data layer includes as well a set of work copies. As aforementioned, SEWOS presents transactional capabilities that help maintain users document and provide restoration capabilities' in case of application or connection failure.

4. PERSONALIZATION SERVICES IN SEWOS

In an earlier work of ours we clarified that the move to personalization is no longer an option, but a necessity [24,25]. Numerous challenges however confront personalization procedure to be successful, scalability, accuracy, evolving user's interests, data collection and preprocessing. As stated previously, users will accept nothing less than an experience that resembles their fully personalized traditional pc desktop. This fact required that our architecture includes a manager that is responsible for all personalization services. Personalization manager makes use of user log, preferences and profile in order to provide personalization. Personalization services that our system proposes are:

1- Memorization represented in user salutation, personalized access
2- User bookmarking
3- Guidance services through hyperlink recommendations
4- Customization services through customizing both layout and content.

SEWOS provides personalization services in different contexts:

1- Memorization and personalized access

After transaction manager authorizes and authenticates user, personalization manager maintains user state so that it displays salutation for user so that to increase user loyalty.

2- Customization:

System maintains user preferences, items, colors and resources so that user desktop is recreated on each access.

3- Bookmarking

This personalization function includes the display information from user's last sessions. In this context, SEWOS uses user log file to build ontology that specify the top ten resources used based on access frequency. The constructed ontology contains resource identification as well associated keywords and ranking sequence in the list to be recommended:

```
<TOPTEN >
<ID> resource ID </ID>
<SEQUENCE>ranking sequence </SEQUENCE>
<FREQUENCY> number accessed by user </FREQUENCY>
<KEYWORDS>extract keywords from file annotation</KEYWORDS>
</TOPTEN>
```

4- Guidance

In the work we pointed out [24,25], web mining techniques were employed in order to provide user a set of recommendations based on usage data analysis. The system built an undirected weighted graph of web site pages in which weights are used to formulate the correlation between each two pages. Then system clustered graph using a threshold value to get clusters of related pages that can be recommended to user based on the requested page identifying pages during user session which last as long as user navigates through web site. However, the earlier system didn't take into consideration temporal attributes besides each page can only belong to one and only one cluster. in this work we try to address these defects by extending the previous system, while the proposed system provides recommendations based on temporal analysis of usage data making use of web usage mining techniques in order to generate a list of recommendations based on the time at which user accesses his personalized desktop. Based on usage analysis we perform classification of user's resources based on access time. We divided access time into four classes (Morning, Afternoon, Evening and Night). However, a certain resource may be accessed several times during day, using typical classification techniques will assign each resource to one and only one class while using Fuzzy relation allows participation of a single resource in different temporal classes and to different degrees.

- In this system user session lasts only for 30 minutes during which log file is updated. On each new session, a new list is generated based on the current time of user. A weight is calculated to show the correlation between a certain resource and the access time based on the following equation:

  $W_{i,j} = N_{i,j} / N_i$

  Where:

  - ☒ $W_{i,j}$: weight to represent correlation between resource I and access time J.
  - ☒ $N_{i,j}$: the frequency of resource I accessed in time J.
  - ☒ N_i : the frequency of resource I access.

This weight is then fed to membership function to calculate membership value which is then used for ranking recommendations, the whole process is shown in Fig.4(a)

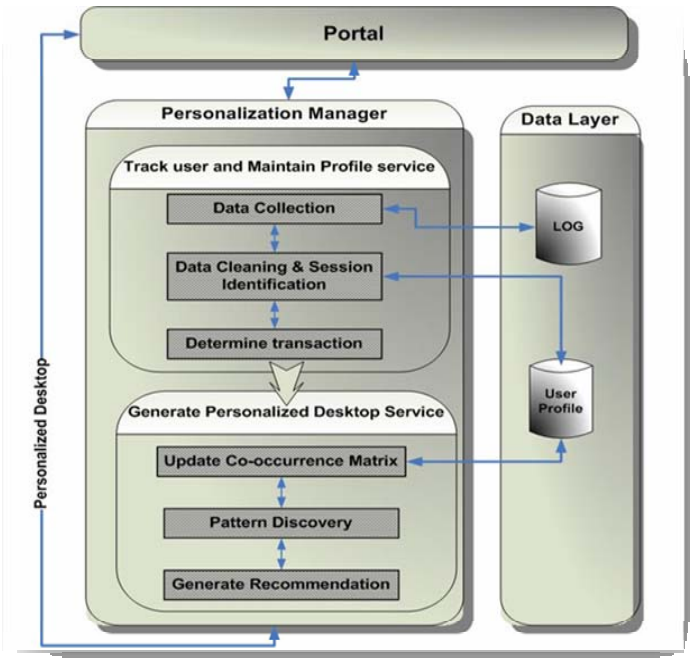

Fig.4(a) Personalization Process

- Besides temporal-based analysis of usage data, semantics-based recommendations are presented to user. While our previous work presented a set of ranked document based on similarity between content and keywords that are extracted implicitly through observing user through navigation. Those keywords would be hard to extract from the normal files. But when all resources are already described in detail (annotated) within Semantic Web ontologies this task carries no burden at all. Thus, a set of semantically related resources can be recommended based on the underlying ontology (i.e. resources with the same file type, author, keywords or even creation date). The whole process of personalization is shown in Fig. 4 (b).

After all, in this work we were motivated by the belief that employing both service oriented architecture as well as web semantics can enhance the functionality of web operating system as well as enrich user experience.

5. CONCLUSION AND FUTURE WORK

SEWOS will provide SOA-based virtual environment and functionality of traditional OS through a web browser where user can interact with applications and services in much the same way they do in their traditional desktop. SEWOS contains an underlying file system that employs web semantics through applying ontology to annotate user resources.

Thus our work is an effort towards building a fully personalized semantically enhanced web operating system that resides remotely and can be accessed through a web browser.

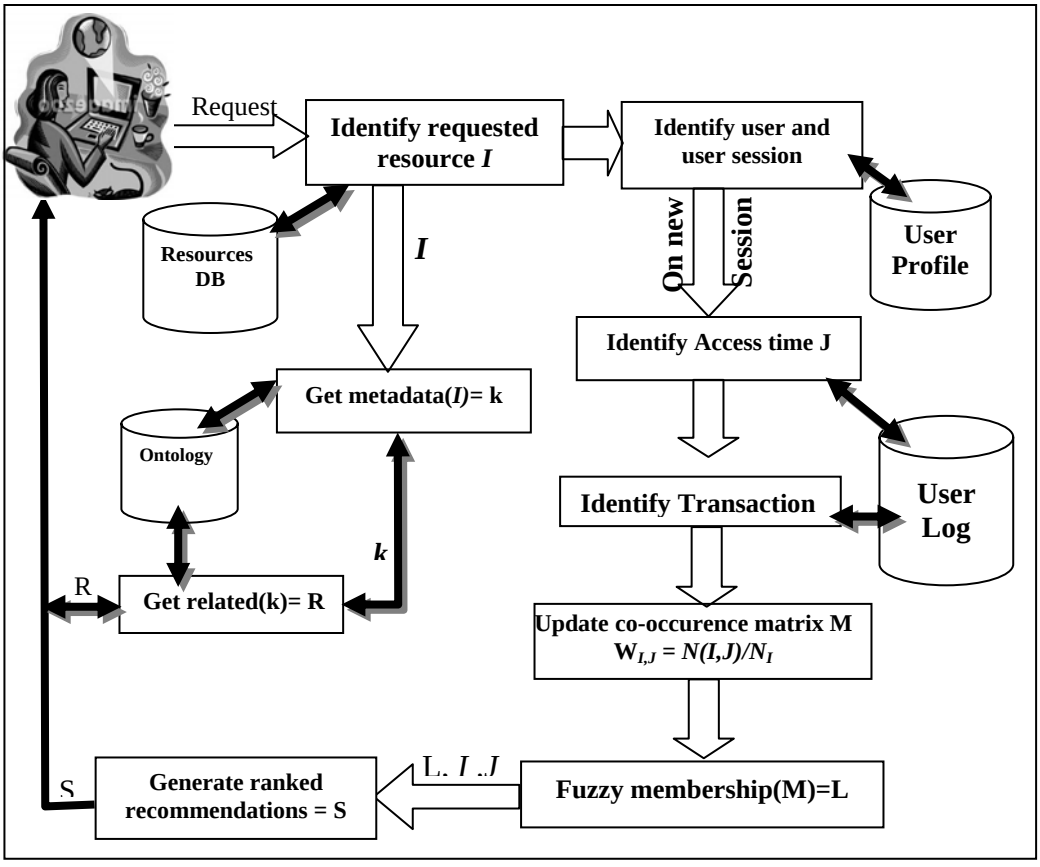

Fig.4(b) Block diagram of personalization Process

REFERENCES:

[1] G. Lawton, "Moving the OS to the Web," Computer, vol. 41, no. 3, pp. 16-19, 2008.

[2] P. G. Kropf, J. Plaice, H. Unger. "Towards a Web Operating System (WOS)". In Proceedings of WebNet'1997.

[3] N. A. Kofahi, A. Al-Taani, "Web Operating system and Computing On The Web", Information Technology Journal Vol. 4 Issue 4, P. 360-366, 2005.

[4] P. G. Korpf, "Overview of The WOS Project", Advanced Simulation Technologies Conference (ASTC1999). San Diego, California, USA, pp.~350--356, 1999.

[5] A. Mufti, K. Salah, "Web Operating System", https://eprints.kfupm.edu.sa/794/1/webos.pdf, 2001. (last visit 5-2009)

[6] A. Vahdat, T. Anderson, M. Dahlin, E. Belani, D. Culler, P. Eastham, C. Yoshikawa. "WebOS: Operating system services for Wide Area Applications", In Proceedings of the Seventh IEEE Symposium on High Performance Distributed Systems, 2002.

[7] A. Vahdat, T. Anderson, M. Dahlin, E. Belani, D. Culler, P. Eastham, C. Yoshikawa. " WebOS: Software Support for Scalable Web Services". Proc. of the 7th IEEE International Symposium on High Performance Distributed Computing, Aug. 1998, pp. 52-63.

[8] N. Abdennadher, G. Babin, P. Kropf. " A WOS™ - Based Solution For High Performance Computing", IEEE Computer Society Press, pp. 568-573. may 2001.

[9] S. B. Franceschi, L. Le, D. Velez. "Web-Based Technologies: Reaching Their Ultimate Potential On Restricted Budgets", Proceedings of the 32nd annual ACM SIGUCCS conference on User services, p.336-339, 2004.

[10] O. Krone, S. Schubiger. "WEBRES: Towards a Web Operating System", Kommunikation in Verteilten Systems. P.418-429, 1999.

[11] J. Iturrioz, O. Díaz, S. F. Anzuola, "Toward the Semantic Desktop: The seMouse Approach", IEEE Intelligent Systems archive. Vol. 23 , Issue 1. P.24-31. 2008 .

[12] T. Heath, E. Motta, M. Dzbor . "Context as Foundation for a Semantic Desktop", Workshop: 1st Workshop on The Semantic Desktop at 4th International Semantic Web Conference (ISWC2005), Galway, Ireland.

[13] L. Sauermann, G. A. Grimnes, M. Kiesel, C. Fluit, H. Maus, D. Heim, D. Nadeem, B. Horak, A. Dengel. "Semantic Desktop 2.0: The Gnowsis Experience", Proc. 5th Int'l Semantic Web Conf. (ISWC 06), LNCS 4273, Springer, 2006, pp. 887–900.

[14] D. Schwabe, D. Brauner, D. A. Nunes, and G. Mamede. "HyperSD: a semantic desktop as a semantic web application". In Proceedings of the ISWC Workshop on the Semantic Desktop. 2005.

[15] G. Tummarello, C. Morbidoni, P. Puliti, F. Piazza. " The DBin Semantic Web platform: an overview", In. Workshop on The Semantic Computing Initiative at. WWW2005 , 2005.

[16] G. Tummarello1, C. Morbidoni1, F. Piazza1, P. Puliti, "The DBin platform: toward a personal tool to experience the Semantic Web". SWAP 2005 (Semantic Web Applications and Perspectives, Proceedings of the 2nd Italian Semantic Web Workshop), 2005.

[17] J. Bao, L. Ding, D. McGuinness, J. Hendler. "Towards Social Webtops Using Semantic Wiki", In Proceedings of International Semantic Web Conference (Posters Demos)'2008.

[18] D. Karastoyanova, F. Leymann. "Service Oriented Architecture –Overview of

Technologies and Standards", Information Technology , Vol. 50 , Nr. 2, p. 83-85. 2008.

[19] D. Anicic, M. Brodie, J.de Bruijn, D. Fensel, T. Haselwanter, M. Hepp, S. Heymans, J.org Hoffmann, M. Kerrigan, J. Kopecky, R. Krummenacher, H. Lausen, A. Mocan, J. Scicluna, I.Toma, M. Zaremba, "A Semantically Enabled Service oriented Architecture", WICI International Workshop on Web Intelligence (WI) meets Brain Informatics (BI) (WImBI), December, 2006.

[20] M. P. Papazoglou, "Service -Oriented Computing: Concepts, Characteristics and Directions", Proceedings of the Fourth International Conference on Web Information Systems Engineering, p.3, December 10-12, 2003

[21] A. Erradi , S. Anand , N. Kulkarni, "SOAF: An Architectural Framework for Service Definition and Realization", Proceedings of the IEEE International Conference on Services Computing, p.151-158, 2006 .

[22] T. B.Lee, J. Hendler, O. Lassila, “The Semantic Web,” Scientific American, p: 35–43, 2001.

[23] M. C. D. Leo J. O. Kevin T. Smith,"The Semantic Web: A Guide to the Future of XML, Web Services, and knowledge Management", Wiley Publishing, Inc., 2003

[24] A. M. Riad, H. K. Elminir,S. F. Sabbeh, “Recommender System For E-Research”, Mansoura Engineering Journal, Faculty of Engineering, Mansoura University, Vol. 33, No 1, 2008.

[25] A. M. Riad, H. K. Elminir , S. F. Sabbeh, “An Intelligent System for Supporting Personalized Scientific Search”, Port Said Engineering Research Journal (PSERJ), Faculty of Engineering, Suez Canal University, Vol. 12, No 2, September 2008.

A.M. Riad - Head of Information Systems department, Faculty of Computers and Information Systems, Mansoura University. Graduated in Mansoura University from electrcal engineering department in 1982. Obtained Master degree in 1988, and Doctoiral degree in 1992. Main research points currently are intelligent information systems and e-Learning.

Hamdy K. Elminir was born in EI-Mahala, Egypt in 1968. He received the B.Sc. in Engineering from Monofia University, in 1991 and completed his master degree in automatic control system in 1996. He obtained his PhD degree from the Czech Technical University in Prague in 2001. Currently he is an associate professor and the head of communication department – masr academy for engineering, Mansoura, Egypt.

Sahar F. Sabbeh was born in Damietta, Egypt in 1982. She received the B.Sc. in Information systems from Mansoura University, Egypt in 2003 and completed her master degree in Information systems 2008.Currently she is an assistant lecturer in Alzarka Higher Institute For Computer And Adminstration Sciences, Damietta, Egypt.