http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 4 (3): 251-255, 2005
ISSN 1812-5638

Towards a Communication System as Hyper-ware

[1]N. Ghoualmi-Zine and [2]A. Araar
[1]Department of Computer Sciences, Badji Mokhtar University, Annaba, 23000, Algeria
[2]Faculty of Computer Sciences, Ajman University, UAE

Abstract: We target the conception of a hyper-ware, which could be the futur of middelware. The presented model is inspired from hypermedia model. The objective of the hyper-ware is the integration of all information systems. It not depends on the organization. Hyper-ware disturbs neither existing application nor their local behaviour. It presents no limits of flexibility and adaptability for specific needs. The tasks are executed through communication networks composed of networked modules represented by a pieces of code belonging to information systems. An architecture separating the tasks, the communication networks and the communicant entities, assures a degree of a very high adaptability also coherence and integrity. This model is inspired from hypermedia model and conception of hypermedia could be the future of midleware.

Keys words: Hyperisation, granularity, integration, middleware, services

INTRODUCTION

The enterprise is passed during twenty years from an architecture strongly centralized to another distributed. Thus, traditionally the users of these applications belong to an enterprise collaborating by grafting external solutions like file transfer, mails and even re-entry of information. The necessity of integrating the information systems/applications in a global information system is crucial. This must be coherent, reliable and follows the evolution of the enterprise[1]. Also, today all is shared and virtual enterprise is in expansion. These decrease the cost of work and reduce execution time of the tasks. It also optimizes the useful of competence and resources (hardware and software). Software for such constraints is in evolution. It has passed from components to middelware to development of services on the web.

For the reasons cited above the proposed work is at the intersection of several domains and necessitates a cooperation of several disciplines.

The focus is the conception of hyper-ware that allows a communication among IS's enterprise among intranet, internet and extranet. This model is inspired from hypermedia model. It consists to apply the hyperisation process applied to a linear document to a set of information systems. The result is a hyper-ware allowing integration. This model will give flexibility missed in existant systems and a degree of genericity to include any module. The model aims also to optimize code and programming. Also, it would help reuse and leverage existing technologies and foster interoperability among different computing models implementations. It would direct the development of the Web services paradigm in a unified manner, targeting an overall solution for integration.

The present study aimed to merge hypermedia model that provides required structures for creating hyper-ware and the Web technologies that provide the open standards needed for achieving interoperability.

EVOLUTION OF SOFTWARE

The evolution of the software over the web context[2] can be summarised in four generations: The first is stand alone generation. The second is client/server generation. The third is Internet generation and the last, which is in perpetual evolution, is web services generation. This evolution has been produced between 1990 and 2000 and progress continues.

Middleware Services reside between applications on the one hand and the operating system together with the networking software on the other (Fig. 1)[3]. Middleware services replace the non-distributed functions of the operating system with distributed functions that use the network.

Distributed systems utilize middleware services for communication, coordination, control, transactional support, presentation management, information management, computation and system management. Additionally, middleware ensures reliability, scalability and performance to enterprise system.

Corresponding Author: Dr. N. Ghoualmi-Zine, Department of Computer Sciences,
Badji Mokhtar University, Annaba, 23000, Algeria
Tel: 21 33 8867180

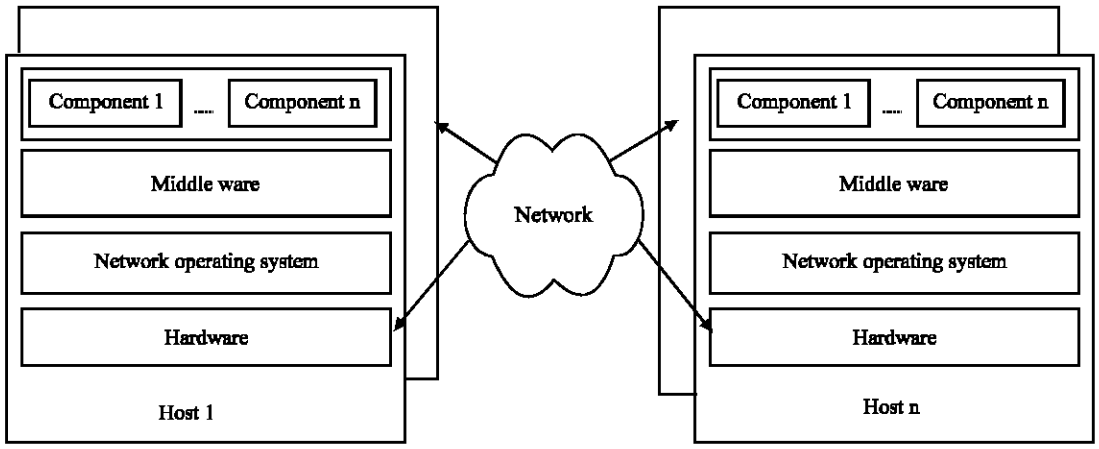

Fig. 1: Middleware in distributed system construction

Web services are components (pieces of application, interface) and middleware (services, communication, security, discovery, compositions) in one[4].

Web services are layered between the applications running on one platform and the applications running on another platform, even across enterprises Web services are capable of replacing the distributed middleware services functionalities that typically use a LAN network with functionalities that use the Internet.

After this summary, it can conclude:

- To make component based applications from different vendors interoperate is however a difficult and tedious task.
- A significant difference is however that Web services aim at preventing incompatibilities between middleware of different vendors.

HYPERMEDIA MODEL

The systematic approach to hypermedia modeling is a recent discipline. Departing from the initial proposals of RMM, HDM and OODHM, several evolutions have been made[5,6].The methodology presents the following set of phases:

Conceptual modeling: It reflects the information structure in terms of classes and relations.

Navigation modeling

Navigation space: Introduces the notion of navigational class, a UML stereotype. These represent the information accessible through a specific navigational perspective-relating to a particular use case or actor. In the database domain, it represents views.

Navigation structure: For the navigation structure diagram, we have four basic stereotypes: (a) Indexes represent computed lists of connections (anchors/links) from instances of one class to a selected set of instances of another class. (b) Guided tours are determined similarly, but the navigation process is stepwise. (c) Queries represent free selection criteria that result in an index. (d) Menus are predetermined groups of links.

The result is a set of models covering different perspectives (or use cases), each one corresponding to a couple of diagrams (space and structure).

Presentation models: It is the static presentation models, that include several stereotypes like frames, collections, forms, text, etc.

HYPER-WARE MODELLING

The conceptual model: Inspired by hyper document structure. Te conceptual model represents the hyper-ware strucure. The structure contains all inventoried information systems. So, hyper-ware has a description like name of enterprise, version number, date, authors, purpose… It is composed by a set of information systems or/and applications (IS1, IS2,… ISn). Each one or all are composed by a set of modules associated to icon in menu.

Thus, the set of applications is modeled as non-linear document and forms a hyper-ware. The links are referential and structural. The obtained structure represents the integration of all information system in a global application The level of granularity is fixed by the processess of hyperisation. It is fixed in this model to module associated to an icon of the menu. With this model, information systems are decomposed into sofware entities or modules. They represent communicant entities.

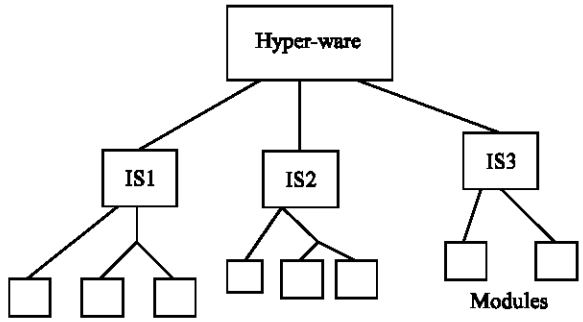

Fig. 2: Conceptual hyper-ware structure

	CE_1	CE_2	CE_3	CE_4	CE_5	CE_6	CE_7	CE_8	CE_9
CE_1	X	x	x		x		x	x	
CE_2	x	X	x	x		x	x		x
CE_3	x	x	X	x	x				
CE_4		x	x	X		x	x	x	
CE_5		x	x	x	X				
CE_6						X	x	x	x
CE_7	x			x	x	x	X		
CE_8	x	x		x	x			X	x
CE_9	x		x		x	x			X

Fig. 3: Interactions matrix

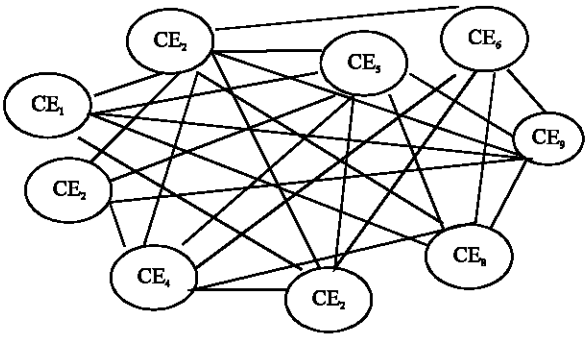

Fig. 4: Web associated to interaction matrix

So, re-definition of applications in term of modules (software entities) or communicant entities[7] is proposed.

Navigational model: A problem with hyper-ware model is that the space of the application in terms of the number of information systems and module can become very large. A large hyper-model becomes complex as the navigational paths of associative links, structural links and navigational design patterns are employed to encompass all possible user tasks and navigational requirements. Unfortunately, it is difficult to track all links of modules. Navigational paths may change, links become invalid and task becomes obsolete[8,9]. Thus, Navigational model is represented in hyper-ware by interaction matrix.

Entity	CE_1	CE_2	CE_3	CE_4
CE_1	X	X		
CE_2		X	X	
CE_3			X	X
CE_4				X

Fig. 5: Interactions matrix for one task

The methodology: Two phases are proposed. The first, consists to dress the communication pattern: List all tasks for each pattern and eliminate tasks that not need computers. The second, consists to follow steps: 1- Draw the conceptual model. 2- Classify modules with their function informations, which produce and/or consume. 3- Make out the interactions matrix for all modules. 4- Draw the web associated to the matrix. 5- Apply the predicate execution task to determine all communication networks. 6- Establish a management of rules in order to assure that the communication network does not have incompatible modules and does not exist communication network incompatible to user tasks.

Therefore, navigational struture is a set of communication networks. We assume that navigational spaces is provided by hyper-thechnolgies: ports, RMI, RPC, protocols,...

Example: Let an enterprise with three applications constituted by nine communicant entities. Figure 2 presents the conceptual structure of the hyper-ware after hyperisation process. Figure 3 shows the interaction matrix is dressed. An x is put for each relation related to two communicant entities. Based on the interaction matrix web is drawn. Figure 4 shows the web associated to the matrix. With just nine communicant entities, it is difficult to be oriented on the graph. Therefore, it is harder when the number of communicant entities increases during the life cycle of the applications and enterprise.

In the second step with the predicate execution task and after dressing all tasks for each pattern, we have to determine the communication network for each one. Figure 5 gives a choosen task T in present virtual example. T is associated to the communication network composed by CE_1, CE_2, CE_3, CE_4.

The example above is the simple case: The communicant entities are sequentially executed like in Fig. 6a. A communication network could contains parallel sub-tasks like in Fig. 6b.

One task could be associated to many communication networks. One communicant entity could be involved in many communication networks (tasks).

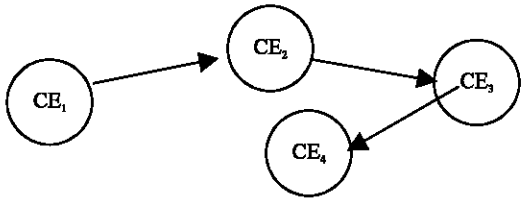

Fig. 6a: Communication network associated to the interaction matrix

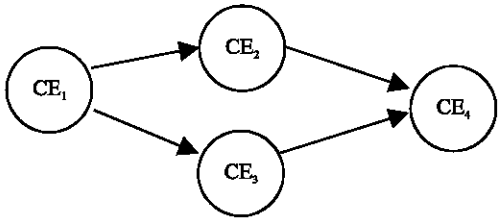

Fig. 6b: Communication network with parallel sub-tasks

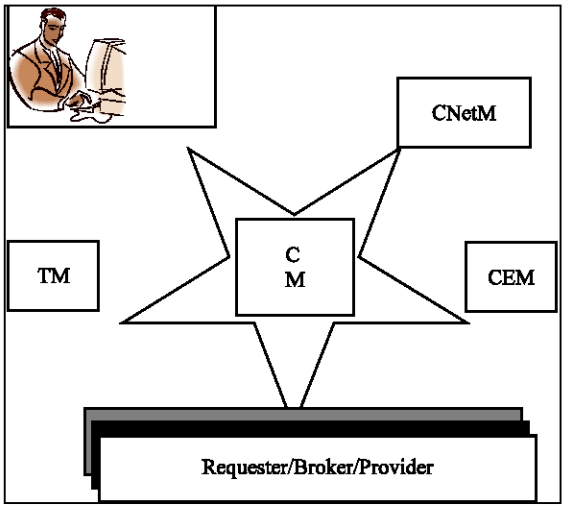

Fig. 7: The architecture of hyper-ware

HYPER-WARE ARCHITECTURE

To avoid the problem of desorientation and manage the navigation' toward the execution of running tasks on the web, an architecture is proposed. This architecture is composed by three bases, each one has its manager and communication manager which links the three bases and execute selected task. Thus, we have a base of tasks, a base of communication networks and a base of communicant entities (Fig. 7).

Communicant entities base: Its role is to make the update the base of software entities /modules with their attributes like name, application and localization. We have to determine input and output of each entity.

The existence of the CE manager is justified because communicant entities have not been designed to work cooperatively with others, but only to exchange information with their immediate environment.

Communication network base: Its role is to update of the base of networks communication. It contains user's tasks. All interactions that the communicant entity wants to save them are in the base. One task can have many communication networks and then we can apply optimization processes (many parameters are defined like time, location, busy resource…). The manager has the reconstitution of tasks. It should know the process at any moment. It has a learning process because emergent knowledge is very useful to improve the tasks. It also knows the constitution of the communication network to send information to local but also to distant communicant entity when the task is outsourced in term of module. Tasks are web services on Internet, Intranet and/or Extranet of the enterprises.

The task base: This base permits to create new user task and delete old and obsolete ones. It permits also to fixe who is allowed to invoke the task. It defines:

- Task type, task, sub-task, (sub tasks can be sequential or parallel or mixed.)
- Communicant entities that are involved in its execution

THE COMMUNICATION MANAGER

It is used as demon to assure the coherence and integrity of communications and to make execution of communication network correspondent to the task invoked with characteristic recorded in the base of communication network (URL, path, type of application,…), of course after all necessary tests. The communication manager constitute a navigator on the web of the enterprise. The execution may needs Requester, Broker or /and Provider. It has to find the location of networked modules and execute them following the associated communication network**.** It receives the requests of users and produces the error messages back to the users. It contains two kind of interfaces: one for admonistrator and one for users. The interface's administrator permit to folow the evolution of the enterprise in sense on new applications, new tasks, olds task, new communication network (to optimize time of execution, Qos …). It also determine profils and authorizations. The users's interface permit users to execute task after checking authentication control, list of

tasks attributed to each user. This architecture is a platform. Each site has to install its own instance.

THE HYPER-WARE PROPERTIES

The hyper-ware is layered between existing heterogenous information systems and Internet /Intranet and Extranet technologie and users. Hyper-ware has the properties below:

1. Hyper-ware disturbs neither evolution of existing information system nor their local behaviour.
2. Virtual Enterprise is allowed when task is outsourced
3. Several network communication could exist, so, when one is failed to execute a task, another is selected by the comunication manager. This represents a degree of autonomy.
4. Based on task, it is considered as web services on internet/intranet. Many standards exist to achieve execution.
5. Description of communicant entities like location permits to evaluate the communication cost in term of time, required equipments,… etc.

CONCLUSIONS

The process of implementing information systems spanning different enterprises comes across the heterogeneity and distribution problems. Software evolution is in progress to resolve interoperability related to heterogeneous information systems/applications. In this article, a model is proposed and architecture for hyper-ware based on hypermedia modelling. Conceptual structure and navigational structure is defined. The granularity is fixed to module, which is sensed has input and output and function. This model would be applied at any enterprise having applications during exploitation and could not support new applications during a development not expected before. Predicate execution task creates communication network. This is defined by interaction matrix. It allows the reutilization and versioning of existing modules. Thus, the cost on coding is reduced. It permits also an optimization of execution task. Hyper-ware disturbs neither existing application nor their local behaviour. Its propose to consolidate the obtained model by an architecture, which will avoid disorientation. It consists to have three bases: task base, software entities base and communication network base. The disadvantages of this model are strictly linked to these of hypermedia. Hyper ware is the futur of middleware.

REFERENCES

1. Gomond, G. and P. Monique, 2000. Expérimentation aux Voies Navigables de France mise en place d'un Intranet, maillon du système d'information. INFORSID, XVIIIe Congrès, 16-19 May.
2. Aoyama, M., 2002. Web services engineering big promises and big challenges too. IFIP TC8/WG8.1 Working Conf. Eng. Inform. Sys. Internet Context, Septempber 25-27.
3. Emerich, W., 2000. Software engineering and middleware: A roadmap. Proc. Conf. Future of Software Engineering, Limerick, Ireland, pp: 117-129.
4. Dimka, K., 2003. Components, Middleware and Web services. In: Proc. IADIS'03, Conf., www/internet 2003, ISBN 972-98947-1-X, 5-8th Nov., pp: 927.
5. Falkenberg, E., 2003. A framework of information system concepts. The FRISCO Report, IFIP WG 8.1 Task Group FRISCO, [online], ftp://ftp.leidenuniv.nl/pub/rul/fri-full.zip accessed March, 2003
6. Fraternali, P. and P. Paolini, 2000. Model-driven development of web applications: The autoweb system. ACM Trans. Inform. Sys., 28: 112-123.
7. Roose, P., 2001. ELKAR: A Component Based Re-engineering Methodology to Provide Cooperation. 25th Ann. Intl. Computer Software and Application Conf., (COMPSAC 2001)-IEEE Computer Soc., Press, PR01372, ISBN 0-7695-1372-7, Chicago, USA., pp: 65-70.
8. Kerr, S. and M.G. Daniel, 2003. Partitioning the navigational model: A component driven approach. WWW/INTERNET, IADIS'03, ISBN 972-98947-1-X, 5-8th NOV.
9. Koch, N. and A. Kraus, 2002. The Expressive Power of UML-based Web Engineering. In 2nd Intl. Workshop on Web-oriented Software Technology. Schwabe, D., O. Pastor, G. Rossi and L. Olsina, (Eds.)