Author(s): Gani, Abdullah., Mouratidis, Haralambos., Abouzakhar, Nasser S., Manson, Gordon A.
Article title: Developing an Intelligent User Manager System for controlling Smart School Network Resources
Year of publication: 2002
Citation: Gani, A., Mouratidis, H., Abouzakhar, N.S., Manson, G.A. (2002) 'Developing an Intelligent User Manager System for controlling Smart School Network Resources', Malaysian Journal of Computer Science, 15 (2) 56-69
Link to published version: http://mjcs.fsktm.um.edu.my/detail.asp?AID=132
DOI: (not stated)
ISSN: 0127-9084
Publisher statement: (not stated)

Malaysian Journal of Computer Science, Vol. 15 No. 2, December 2002, pp. 56-69

DEVELOPING AN INTELLIGENT USER MANAGER SYSTEM FOR CONTROLLING SMART SCHOOL NETWORK RESOURCES

Abdullah Gani, Haralambos Mouratidis, Nasser S. Abouzakhar and Gordon A. Manson
The Centre of Mobile Communication Research
The Department of Computer Science
The Faculty of Engineering
The University of Sheffield
211 Portobello St, Sheffield S1 4DP, UK
Tel: 0114 222 5888
Fax: 0114 222 1810
email: a.gani@dcs.shef.ac.uk
h.mouratidis@dcs.shef.ac.uk
n.abouzakhar@dcs.shef.ac.uk
g.manson@dcs.shef.ac.uk

ABSTRACT

This paper presents an Intelligent User Manager System (UMAS) for controlling access to network resources in a Smart School network. Network resources, especially in a Smart School, are in short supply and relatively expensive to acquire, therefore a control mechanism should be in place so that available resources can be allocated for legitimate usages only. An intelligent mechanism using Fuzzy Logic is deployed for the purpose of knowledge learning in order to process all the user requests accordingly. A decision of granting a network resource request needs to be based on several data sets that represent the current network state, transmission state and users. The system is analysed and designed using the Tropos Methodology. Tropos was chosen because it covers four stages of development. The proposed system was modelled using Fuzzy Logic algorithms for simulation purposes in order to find the relationship between two fuzzy sets with the computed allocated time.

Keywords: ***Network Management, Artificial Intelligent, Agent Technology, Software Development Methodology***

1.0 INTRODUCTION

Misusing network resources has exponential consequences to the performance and availability of network services. Network resources like bandwidth are costly and critical to most of the Smart School processors. We mean processors as a unit that is responsible for handling a specific task. Plainly, this resource has to be available to the users as it helps in achieving the objectives. Nevertheless, all the resources are prone to misuse by insiders that are users of a Smart School network. For example, there is a tendency to misuse the email for personal reasons at the workplace and it has grown substantially. This might be used to broadcast messages with the intention of disseminating racist, sexually explicit, or other offensive material. In addition, confidential information of the Smart School may be sneaked out without trace and this certainly would create immediate repercussions. When the email messages clog the network, the main functions of the Smart School will be affected [1].

David [2] reports an alarming figure in misuse of e-mail in offices for back-stabbing, and for sending racist and sexual material. It seems clear that employees of the organisation have tampered with network resources for personal interest. This certainly can clog the network with the transmission of email packets between the User Agent and Mail Transfer Agent for unproductive purposes. All these have led to the requirement of having control mechanisms that are not only capable of preventing unauthorised access to the network resources but also capable of providing intelligent reasoning in responding to user request for accessing network resources.

This paper presents a development of a User Manager Agent System (UMAS) that is capable of controlling user's request of accessing network resources and controlling the network resources availability intelligently. The UMAS deploys a Fuzzy Logic algorithm in order to reach a dynamic control decision. A general overview of the Smart School concept is presented in the next section.

2.0 SMART SCHOOL CONCEPT

The Smart School concept has never had a common definition amongst educationists. In the UK, Smart School was called the 'School of the Future' and others might call it as an e-School. Despite divergence in name, in essence, a Smart School refers to an educational process reengineering undertaking to transform all the school operations which were using conventional means for content delivery into the Information Communication Technology (ICT) dependency outset. Traditional ways of teaching and learning are replaced with the ICT-based techniques such as multimedia-based lessons, Authoring-on-the-fly and networking-based learning. The ICT is not only introduced in a class as a subject but as a tool for teaching and learning purposes. Students can access lessons at their own pace and it is available 24-7. Besides pedagogy that is conducted in the ICT fashion, the management of teaching and learning are also changed by the deployment of this relatively new technology. The most distinctive feature of the Smart School is the deployment of the information infrastructure that is a network for executing the school operations. The network in the Smart School plays vital roles in helping the achievement of Smart Schools' aims and objectives as more than half of the processes in the Smart School are involved with the use of electronic data exchanges [3].

As most of the Smart School operations are highly dependent on the availability of network services and resource, it is important to have mechanisms that are capable of controlling network resources availability to the right user(s). Without control mechanisms in place, user's request for applications can contribute to slow performance of network response, which could be detrimental to the efficiency of the network. Distributed applications performance are subject to communication channels efficiency for transmission, which means applications only can be accessed or transmitted if there is available and sufficient bandwidth. Therefore, it is clear there is a need for network resources to be managed intelligently in response to different types of applications and a great number of users who are using the same network for different purposes.

3.0 USER MANAGER AGENT SYSTEM (UMAS)

UMAS is an intelligent multi-agents system that is capable of coordinating user's request and network resource availability in a network. The scope of network resources has been identified and only covers network applications. When a user logs into a network, profile and policy files are read in order to ensure the right for accessing the network resources. A profile file is a repository of information about the user. A policy, in this context, refers to an abstraction of rules that govern the usage of network. These profile and policy files are created by network administrator(s) in compliance with the organisational policies. Because these files are manually created, changes would be difficult especially when the number of users is relatively large and having frequent policy changes. The UMAS will be able to eliminate these problems effectively after knowledge base has been developed as a result of user's interactions with the network. In other words, the UMAS learns about the user(s) itself by building the knowledge about the user(s) and the network for future use.

The UMAS is also capable of intelligently processing the requests from the user(s) by doing a supervised learning on historical data as well as current data about the user(s) and the network state. In doing so, the UMAS requires data acquisition from the Knowledge Base Repository, the Profile Repository, and the Policy Repository, which fundamentally are logical databases. Logical databases are places for storing the collected information of several parameters extracted from the profile, policy file and network state. These repositories lead to the requirement of having reasoning mechanisms for a supervised learning to take place. In this regard, Fuzzy Logic is used for reasoning about the stored data in the repositories.

4.0 DESIGN OF THE SYSTEM

For modelling the UMAS Tropos [4], an agent-oriented methodology is used. This decision is made since Tropos covers the full spectrum of the software development stages from the early analysis to the actual implementation [5]. Tropos adopts the i* modelling framework [6] which uses the concepts of actors, goals and social dependencies. The methodology is divided into the following four stages:

- Early Requirement Analysis in which environment of the proposed system is modelled in terms of stakeholders and their objectives. The stakeholders of the system are represented as actors while their objectives as goals.

- Late Requirement Analysis in which actors are analysed and the proposed system is introduced as another actor. Furthermore, the proposed system's functional and non-functional requirements are modelled.

- Architectural Design in which the proposed system's global architecture is defined in terms of subsystems, interconnected through data and control flows. Within the environment, subsystems are represented as actors and data exchanges are represented as actor dependencies.

- Detailed Design in which AUML is used to complement the features of i* to model each architectural component of the system in more detail.

4.1 Applying the Tropos Methodology

4.1.1 Early Requirement Analysis (ERA)

In the ERA, stakeholders and their intentions are identified which are represented with actors and goals respectively. Tropos introduces actor diagrams in which a node represents an actor, and the link between different actors represents dependency of actor in accomplishing some goals. This case study has identified the following stakeholders:

- User is an authorised user of the Smart School network who wishes to access appropriately the network resources.
- Smart School Network (SSN) is a network that helps the achievement of the Smart School aims and objectives by controlling network resources.
- State Education Department is a government agency that wishes to have easy communication.
- Ministry of Education (MoE) is a government agency that provides network resources.
- Parent wishes to be well informed.

Fig. 1 illustrates actors (stakeholders) and goals (intentions) of the proposed system. The User actor has a main goal of *Accessing Appropriate Network Resources* and on the other hand, the *Smart School* has a main goal of *controlling the network resources*. The *User* depends on the Smart School to accomplish the goal and the soft goal of *network resources well utilised*. Another actor is the *Parent* that has a soft goal of well informed; State Education Department as an actor has a goal of *easy communication*; and Ministry of Education as actor has a goal of *Providing Network Resources*.

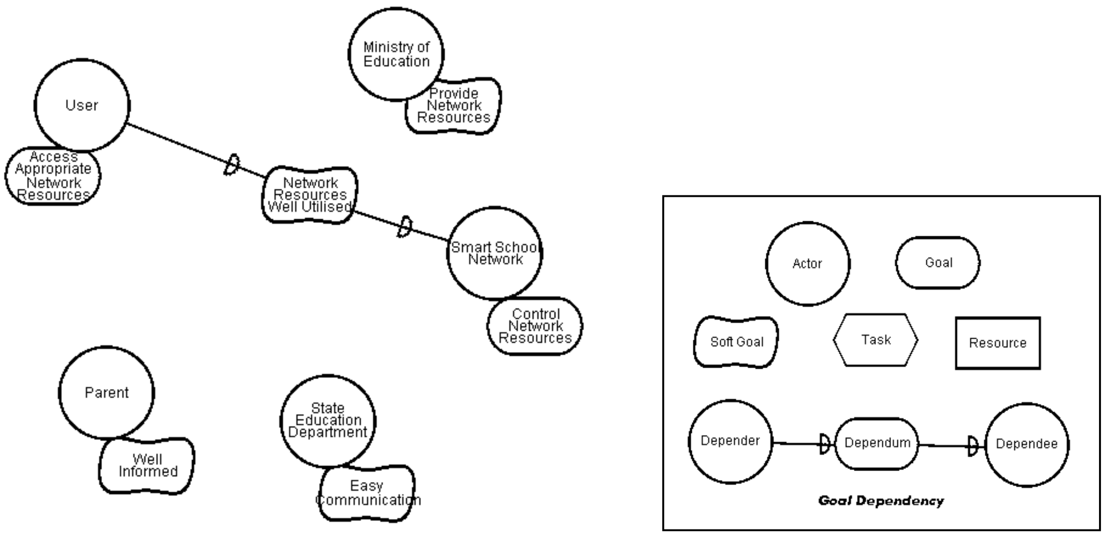

Fig. 1: The stakeholder of the proposed system

Despite several actors has been identified as depicted in Fig. 1, this research only focuses on two actors – *the User* and *the Smart School*. After identifying the goals and the dependencies between these two actors, the next step is to analyse in more details each goal relative to the stakeholder to find out responsibilities of its fulfilment. In doing so, a method called means-end analysis is used. A means-end analysis produces a diagram called Rationale Diagram in which goals are analysed to discover ways of fulfilling them.

Fig. 2 shows a goal of access appropriate network resources from a user perspective and its dependency is analysed. The user has one main goal that is *an access to appropriate network resources*. In order to fulfill the latter goal, it has to be decomposed into two sub-goals – request permission for accessing the network resources and use available resources. Subsequently the latter sub-goal is further decomposed into another two sub-goals – request Local Based Application (LBA) and Internet Based Application (IBA). Both sub-goals need the task of processing the request before permission to use application can be granted. Before granting the requests, three sub-tasks have to be completed – get current policy, get user policy and retrieve knowledge of network state.

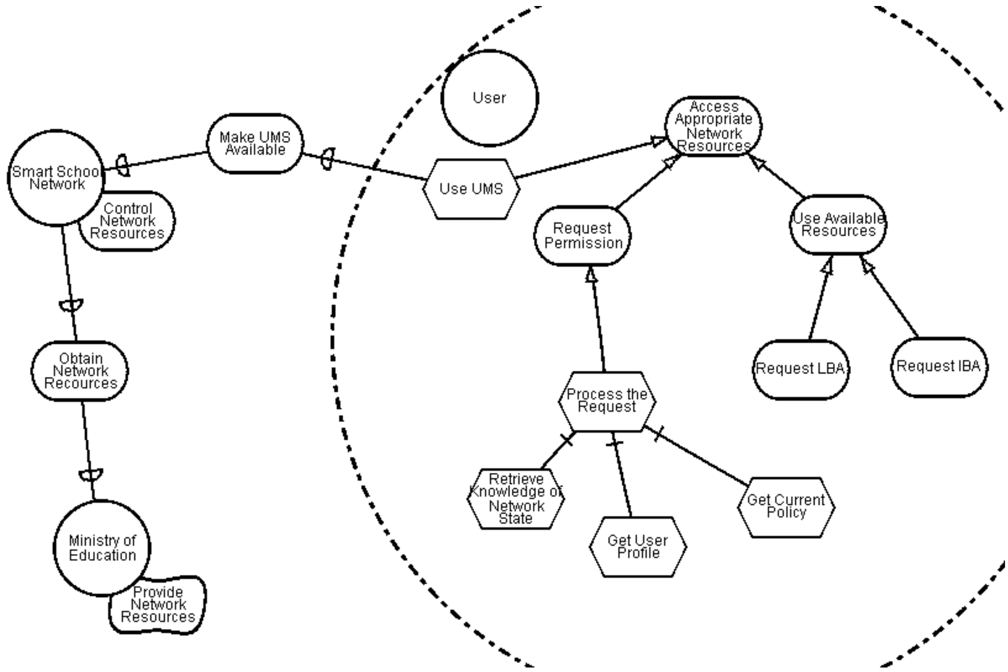

Fig. 2: Means- End Analysis of the User

Another important stakeholder is the Smart School. Fig. 3 illustrates the rationale diagram of the Smart School in which the goal is to control the network resources. The latter goal is achieved by a further three sub-goals decomposition - authenticate user's resource request, allocate appropriate resource request, and manage resource availability. The sub-goal of authenticate user's resource request will be fulfilled by a task of retrieving user permission which would validate the user's right to access the requested network resource. For a sub-goal of allocating appropriate resource requested, it needs two tasks for its fulfilment – find appropriate currently dynamic network resources and display appropriate currently dynamic network resources. The third sub-goal of the Smart School actor is to manage resources availability which is achieved by four dependent tasks – collect information current usage network resources, monitor threshold level, execute control actions and inform user/s network resources changes.

4.1.2 Late Requirement Analysis

The purpose of this stage is to describe the system-to-be within its operation environment along with relevant functions and qualities. The system is represented as two actors who have a number of dependencies. These dependencies define all functional and non-functional requirements for the system-to-be.

The research has identified the following requirements to be embedded in the proposed system.

- The system must be developed with the consideration of having the capability to provide the appropriate level of services to a right user or a group of users. This means that the network must be accurate in allocating the 'right' resources to the right user or a group of users.
- The proposed system must be intelligently capable of adapting to the dynamic of network resources in granting the user's requests.
- A request for a network resource has to be treated accordingly depending on all rules and conditions imposed.

- Priority is given to the request of network resources that has impacts on the Smart School objectives execution.

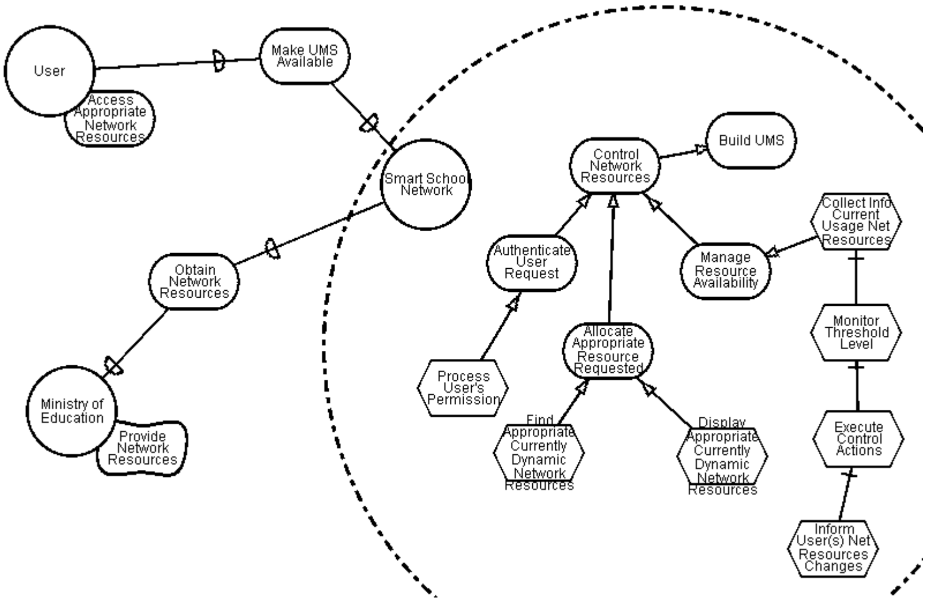

Fig. 3: Means-end Analysis of the Smart School

Fig. 4 illustrates the proposed system. It has another actor, called a User Management System (UMAS) that has a goal of managing user's requests upon the network resources utilisation. Both actors, the User and the Smart School depend on the UMAS to fulfil the goals. The UMAS aims to coordinate the goal fulfilment of both actors. In order to fulfil the latter, the UMAS has to process the requests of the user according to the specifications of the network resources control. This is needed not only to allow an authorised user to access the network resources but also to provide a service that would not put the network operations in danger due to exhausted network resources. The network must be able to provide a service that is merely needed to fulfil the aims and objectives of the Smart School. In other words, the network resources availability must dynamically be deployed to achieve the functions of the Smart School.

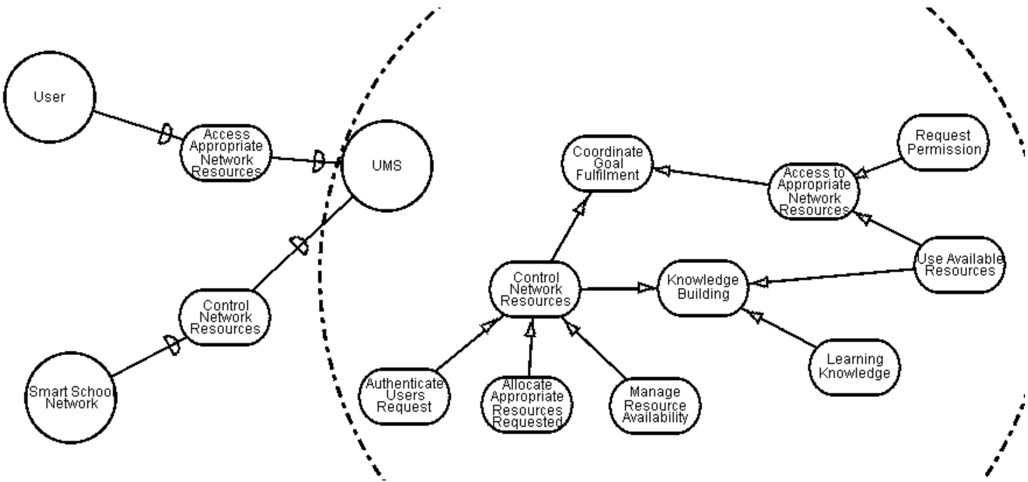

Fig. 4: Means-End Analysis of the UMAS

Apart from having a goal of coordinating the user request for accessing the network resources, the UMAS has another goal of coordinating the knowledge building. This goal can be achieved by two tasks of providing training data and storing into knowledge repository.

The soft goals of the UMAS are self-operated and self-regulated capability. Both soft goals have positive contributions to the UMAS goal achievement. Fig. 5 shows a partial decomposition of the actors and sub-actors of the UMAS along with their dependencies.

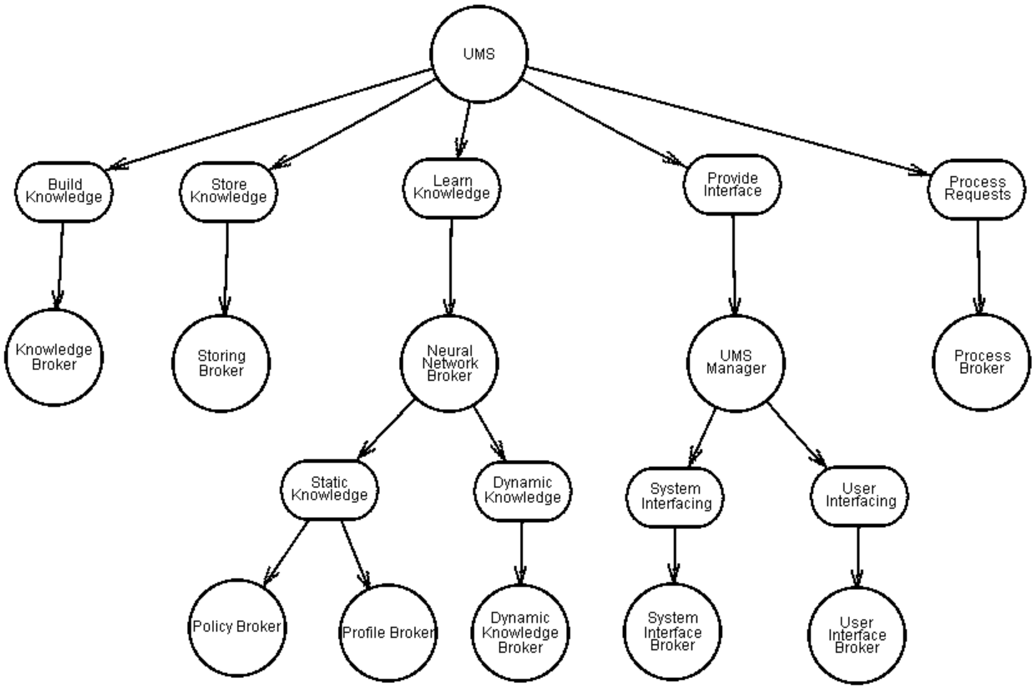

Fig. 5: Sub-Actors Decomposition of the UMAS

The UMAS depends on the Knowledge Broker to provide knowledge about the network and the users, the Storing Broker to store processed knowledge. The Learning Knowledge depends on Neural Network Broker to learn the knowledge. The Neural Network Broker depends on both Policy Broker and Profile Broker to gather training data for static knowledge, on Dynamic Knowledge Broker to build dynamic knowledge. The UMAS Manager provides interfaces to the UMAS. The System Interface Manager provides an interface with the components of the UMAS, and the User Interface Broker provides an interface with the users. The Process Broker provides a service to process the request for network resources.

4.1.3 Architectural Design

The architectural design includes the following four steps:

- Addition of new actors. This will allow the system to interact with the external actors as well as contribute positively to the achievement of some non-functional requirements. Tropos introduces the extended actor diagram in which the new actors and their dependencies with other actors are presented.
- Actor decomposition. At this stage, an actor is described in detail with respect to its goals and tasks.
- Capabilities identification. This stage identifies capabilities that are needed by an actor in order to fulfil its goals and tasks by analysing the extended actor diagram. Each dependency relationship can give place to one or more capabilities triggered by external events.
- Agent Assignment. This will identify agent types and assign its capability.

The UMAS is a software agent that will act on behalf of users. Each user will have an individual software agent, which will be customised to the individual needs. The software agent, the UMAS will have knowledge about the network state as well as the user's past behaviour to enable the UMAS to act intelligently.

The UMAS will consist of the following architectural choices:

- The system will consist of software agents as well as the users.
- End user will have an individual software agent.
- The network is able to adjust the level of services provided for a particular user depending on the legitimate needs.

- What software agent will be capable of learning and building knowledge?
- The software agents will be able to communicate between themselves and to the users.

Fig. 6 illustrates the extended actor diagram with respect to the Process Broker. The Process Broker is responsible for processing the request of the user to access the appropriate network resources. In doing so, the Process Broker interacts with Neural Network Broker to obtain knowledge of the network state, policy and user profile. The Neural Network Broker updates the knowledge acquisition dynamically and informs the user any current changes as well as past changes.

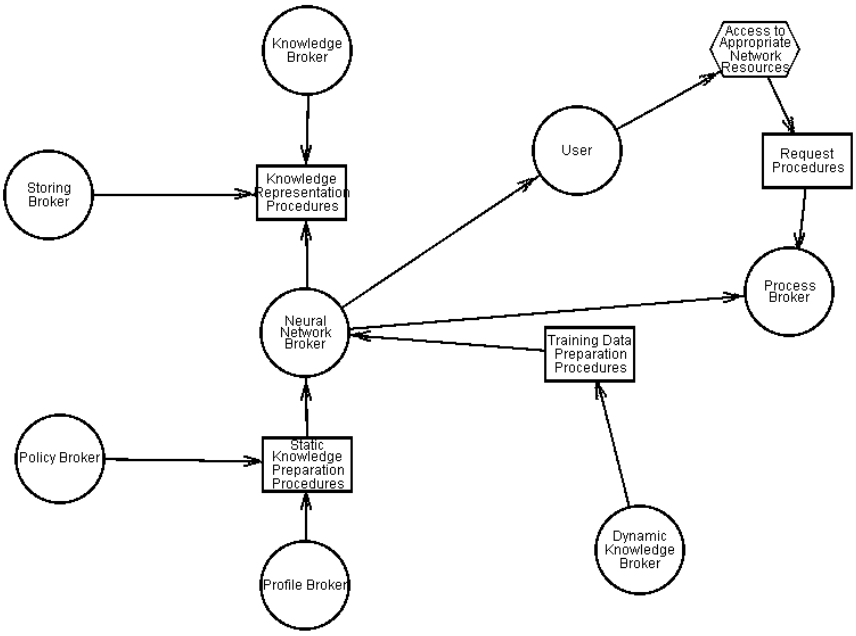

Fig. 6: An Extended Actor Diagram with respect to the Process Broker

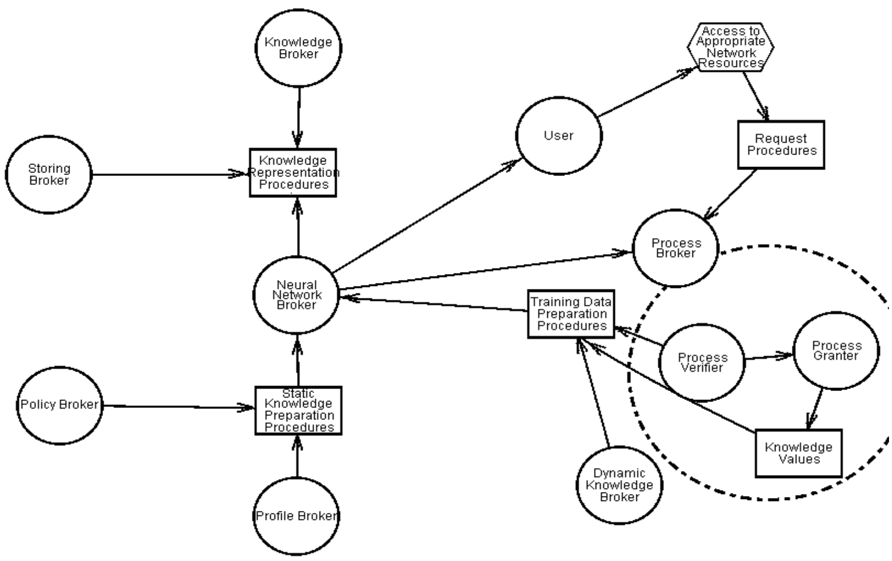

Fig. 7: Extended Diagram with respect to the Process Broker – Internal Decomposition

The next step in the architectural design is the decomposition of actors in sub-actors aiming to expand in detail each actor with respect to its goals and tasks. Fig. 7 shows the decomposition of the Process Broker actor with respect to the process request task. The Process Broker is decomposed into two sub-actors – Process Granter, Process Verifier. The former is responsible for retrieving knowledge and grant access to the services, while the latter is responsible for checking the knowledge quality. The Process Granter depends on the Process Verifier to approve requests from the user.

The next step is the capabilities identification in which the capabilities needed by each actor in order to fulfil its goals and tasks are modelled. The extended actor diagram is used to identify the capabilities since each dependency relationship can give place to one or more capabilities triggered by external events.

The last step of the architectural design is the assignment of the agents. In this step, each agent is assigned with one or more different capabilities that were identified in the previous step. The following table presents the assignment of capabilities on the agents with respect to the task of Process Request.

Table 1: Agent Type and Capabilities

Agent	**Capabilities**
Process Verifier	Get Training Data Get Policy Notation Get Profile Values
Process Granter	Get Training Calculate Data Error
Neural Network Broker	Get static knowledge Provide knowledge to User Provide knowledge to Process Broker
Policy Broker	Get Policy Notation Provide policy to Neural Network Broker
Profile Broker	Get Profile of particular user Provide profile values to Neural Network Broker
Dynamic Knowledge Broker	Get knowledge of current network state Get knowledge of current user Provide knowledge to Neural Network Broker
Storing Broker	Send knowledge values to repository Organise knowledge hierarchically
Knowledge Broker	Get data from network state Get data from Server Get data from Router Get data from Switches

4.1.4 Details Design

This stage has the aim of specifying agent capabilities and interactions. Therefore, during this stage internal and external events that trigger plans and the beliefs involved in agent reasoning are modelled. Tropos adapts a subset of the AUML diagram. These are:

- Capability Diagrams. We use AUML activity diagrams to model a capability or a set of capabilities for a specific actor. In the capability diagram, the starting state is represented by external events, activity nodes model plans, transition arcs model events, and beliefs are modelled as objects. An example of a capability diagram is the Process Granter of getting the user's training data. Process Granter will learn from the training data and then decide to either allow or deny access to the service.

- Plan Diagrams. This diagram is used to further specify each plan node of the capability diagram. For example, the User agent sends a login request to the Process Granter. The Process Granter learns from the training data by interacting with the Neural Processing Unit and if the Actual Output is equal to the Desired Output then the acknowledgement will be sent to the user and the Enforcer.

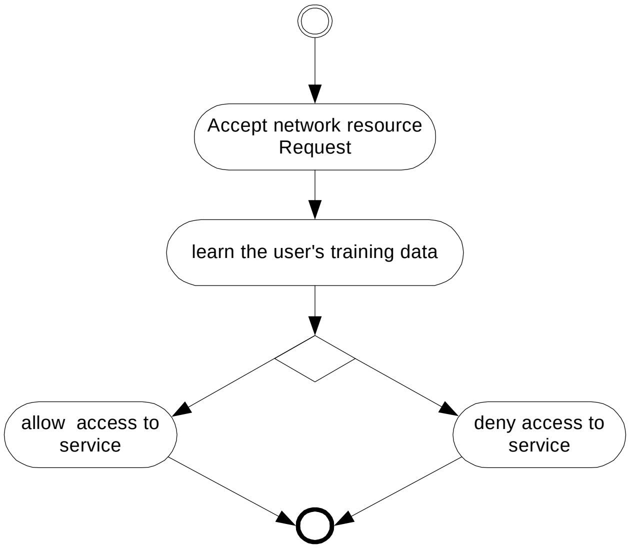

Fig. 8: Capability Diagram of the Allow or Deny Access Capability

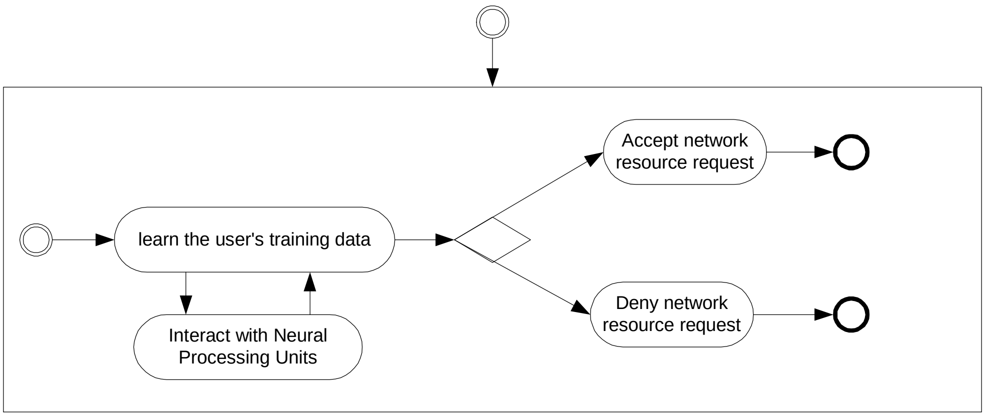

Fig. 9: Plan Diagram for the Access Request Authorisation from the user

- Agent Interaction Diagrams. This research applies the sequence diagram modelling agent interaction protocol. Fig. 10 illustrates an example of an Agent Interaction Diagram. The user sends an Access Request to the Process Granter, then the Process Granter forwards the details of the request to the Neural Processing Unit as part of the input and in return, the Neural Processing Unit will acknowledge the Process Verifier for final decision before forwarding it to the Enforcer.

In the next section, the model of UMAS will be simulated using Fuzzy Logic for the purpose of seeking the right algorithm of controlling the allocated time for a particular session.

5.0 UMAS SIMULATION USING FUZZY LOGIC

Fuzzy logic and fuzzy-based systems are mathematically based systems that enable computers to deal with imprecise, ambiguous, or uncertain information and situations, i.e. the real world. Fuzzy logic originally extended Boolean logic to handle truth-values between the extremes of completely true and completely false [7]. In contrary to Fuzzy logic [8], rule-based systems assume that we live in a clear-cut world, where every hypothesis is true, false, or unknown. It also assumes that many systems make use of the closed-world assumption, where by any hypothesis that is unknown is assumed to be false, while this model of reality is useful in many applications, real reasoning process are rarely so clear-cut, such as network management systems.

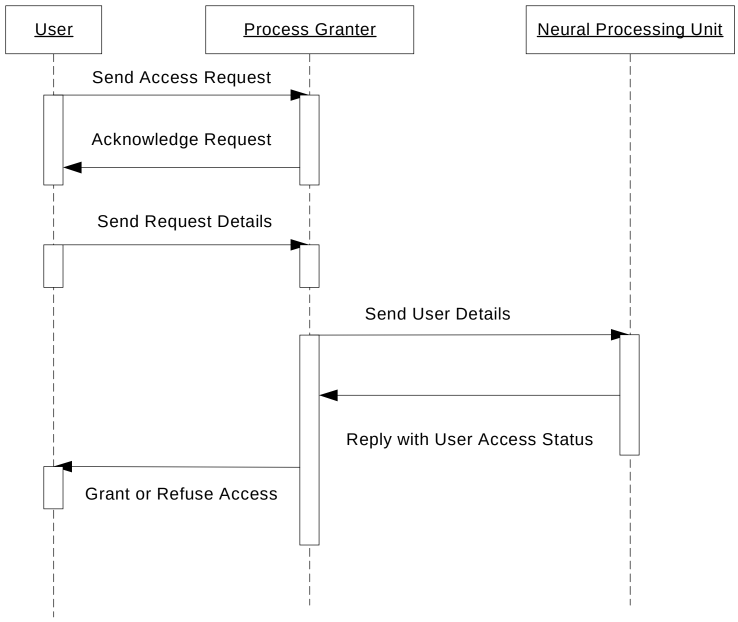

Fig. 10: Agent Interaction Diagram for Learning Procedure

Zadeh [9], the founder of fuzzy logic concept in 1965, proposed this concept to help computers reason with uncertain and ambiguous information. He states "Traditional methods of systems analysis are unsuited for dealing with systems in which relations between variables do not lend themselves to representation in terms of differential or difference equations. Such systems are the norm in biology, sociology, economics, and, more generally, in fields in which the systems are humanistic rather than mechanistic in nature. The methodology of fuzzy if-then rules serves to increase the machine intelligence quotient (MIQ) and lower the cost of products by making it possible to program their behaviour in the language of fuzzy rules, and thus lessen the need for precision in data gathering and data manipulation. In this context, the use of linguistic variables and fuzzy rules may be viewed as a form of data compression. It is certain to have a wide-ranging impact by showing how fuzzy set theory and fuzzy logic can be employed to exploit the tolerance for imprecision, and how they make it possible to address problems that lie beyond the reach for traditional methods".

Enhancing Agents with fuzzy reasoning capabilities can give them the ability to exist within domains where knowledge may not be quite as crisp as we are used to dealing with in traditional computing approaches. Since agents exist in the real world, therefore equipping them with fuzzy capabilities would improve their ability to cope with imprecise systems' events and users with uncertain behaviour. Fuzzy-logic provides suitable searching capabilities to retrieve certain patterns in uncertain environment. This feature would help in decision-making process for computing the time that can be allocated for a single session.

A decision of granting a user's request for a specific application depends on many variables of the Smart School network. The first consideration is the criticality (AppC) of the requested application. Criticality refers to the importance of application to Smart School functions. If the requested application is critical then the user is allowed to have a longer time to use the application. Nevertheless, this depends on the user priority. The decision of allowable time is important because of the nature of the network resources that always in short supply and the degree of availability varies. The hour of AllocatedTime (*t*) will be updated at the interval time of 100 microseconds, which is sufficient for manager agent to communicate with the router. As the number of application requests increases the AllocatedTime (*t*) will also be affected. The second consideration is bandwidth availability (BWA). If BWA is high then a longer time for accessing the network resources will be allocated. Both variables BWA and AppC have three membership functions (MF) – low, medium, and high. Similarly, the AllocatedTime variable has three MFs too. AppC is said to be low if the user uses the application illegally. For example, if a user uses an email application within a specific period of time that is prohibited, the criticality of such application is graded as low, and then will be given low priority. Similar application if the school administrators use it will be given higher priority and without any restriction. Determination of the criticality depends on many parameters that will be set by the school administrators such as roles of user, period that is prohibited, and application requirement itself. Memberships frequencies of bandwidth are determined from a set of parameters such as current load of network, latency, throughput rate and queue state. The following table summarises all the MF interval values that used in the modelling.

Table 2: Memberships Frequency Interval Values

	Low	Medium	High
AppC	0 –3.5	3.6 – 7.5	7.6 - 10
BWA(%)	0 -35	36 - 75	76 - 100
AllocatedTime (hours)	0 – 3.5	3.6 – 5.9	6 - 10

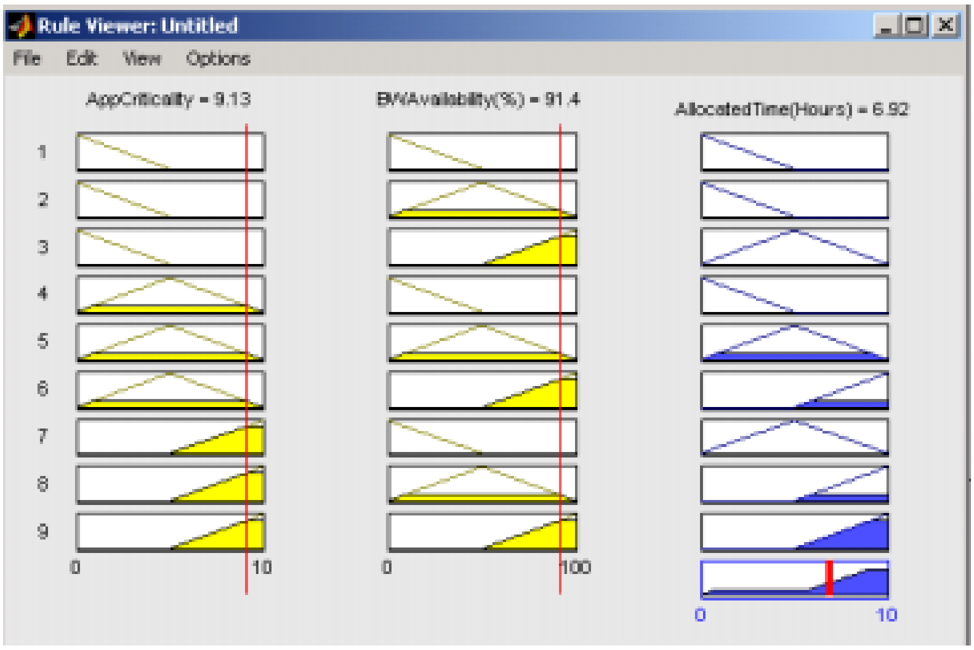

Fig. 11: Rules View of High BWA and High AppC

Fig. 11 illustrates that AllocatedTime (t) corresponds to the application criticality and bandwidth availability. The AllocatedTime (t) will increase for a user to use the requested application as the application is critical and the bandwidth is available for use. As the AppC decreases the AllocatedTime (t) is also will be affected for a shorter time. Fig. 12 shows scenario of having high percentage of bandwidth availability but the application requested virtually has minimal impact to the Smart School objective achievement when the AllocatedTime (t) declines. If the user request for another application the current AllocatedTime (t) will be discarded and a new AllocatedTime (t) will be computed. As long as there is no new request for application, the user can continuously uses the application within the AllocatedTime (t).

As the BWA drops substantially to a low because of many new requests from the users to access the applications but the application that is currently running is critical, the AllocatedTime(t) will also be reduced. Fig. 13 illustrates a scenario as AppC escalates to a high interval while the BWA drops to a low interval, the AllocatedTime (t) remains at the medium interval.

For a worst-case scenario where both variables are at low MF, plainly the AllocatedTime (t) will be affected and could lead to denial of access. Fig. 14 shows the AllocatedTime (t) drops to a lesser hour as a result of both variables dropping.

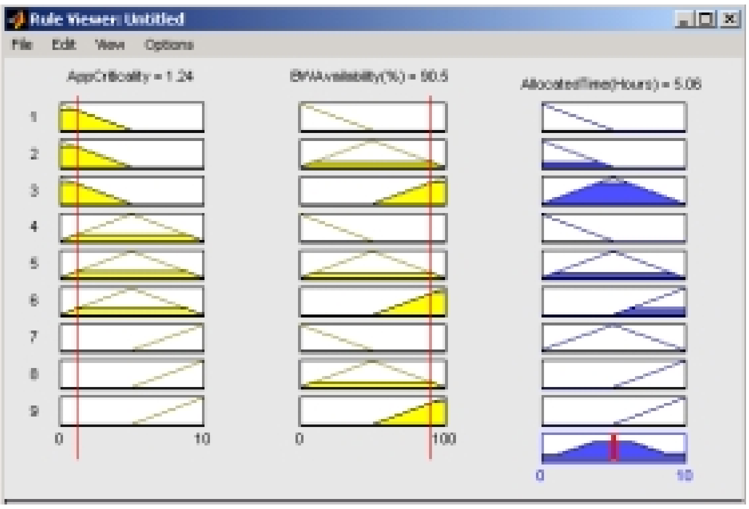

Fig. 12: Rules Views of High BWA and Low AppC

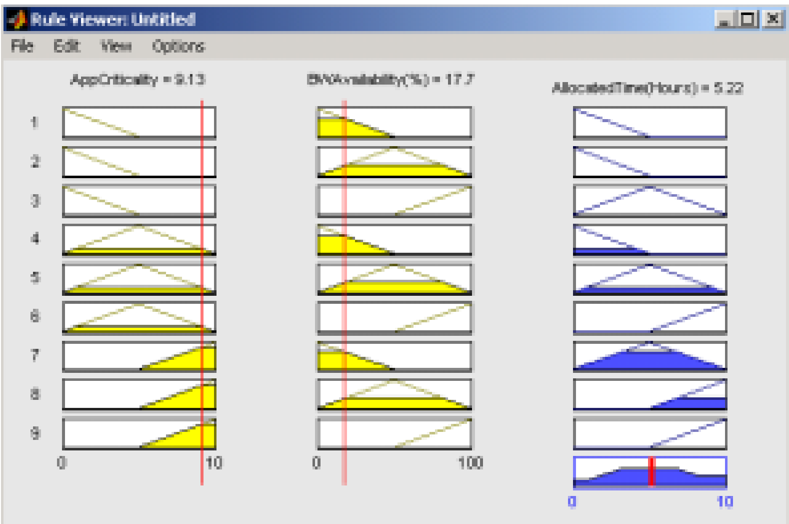

Fig. 13: Rules View of High AppC and Low BWA

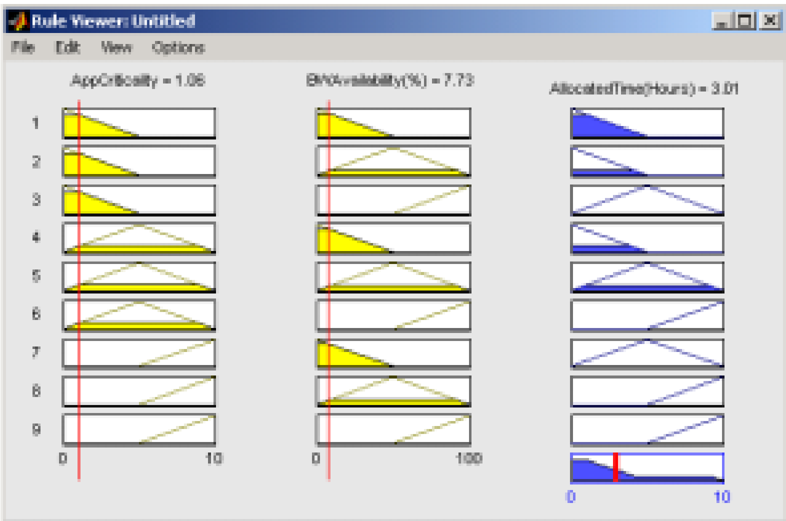

Fig. 14: Rules View of Low AppC and Low BWA

Fig. 15 is a Surface Diagram that shows the relationship between AppC and BWA for AllocatedTime of a user request. It is clear that relationships between both variables are a non-linear type. Although, the AppC increases proportionally with the AllocatedTime (t) but it is still subject to BWA. So, BWA is an important factor that determines the AllocatedTime.

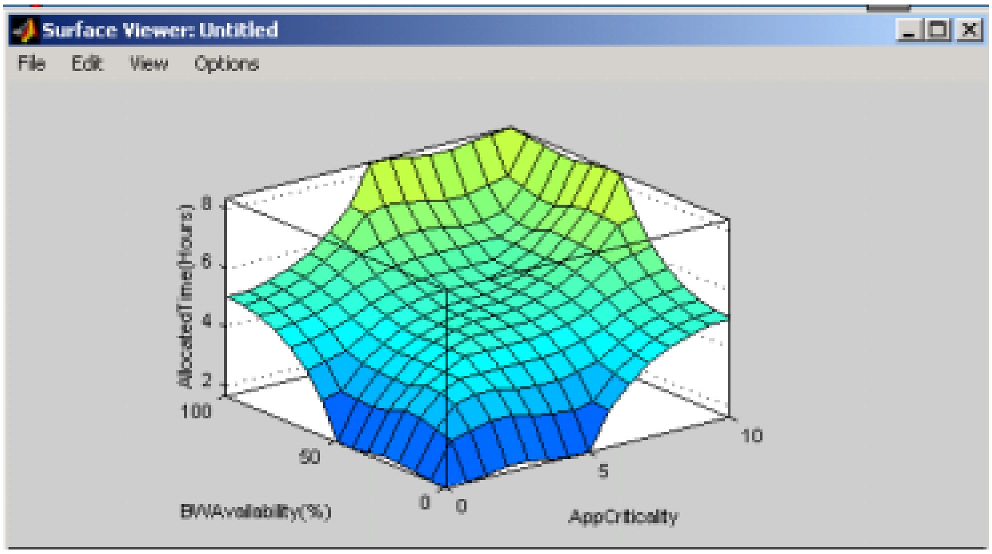

Fig. 15: Relationship between AppC and BWA for AllocatedTime

Based on all depicted scenarios, it is clear that bandwidth availability is significantly important to the computation of time allocation for having a session. However, bandwidth availability should be made adaptive in providing an appropriate level of time interval and never let be in idle. Although application critically contributes less impact to the decision of time allocation, it is worthy to be considered in the decision process. This is to ensure that critical based applications are given higher priority than there non-critical counterparts. Usually, critical applications are meant for either higher priority users, who require QoS, or used during specific times, in which resources need to be effectively allocated.

6.0 CONCLUSION AND FUTURE WORK

This work shows another alternatives to control access mechanism apart from Role Based Access control and Policy Based Control [10]. The idea of this work is simply to create another middleware that can be used for interfacing between network resources and user's requests. In other words, the work is trying to establish mechanisms that are capable of proactively deciding the best option for both the users and the networks, especially in Malaysian Smart School networks. Nevertheless, the idea of UMAS can be applied to another area of applications such as e-Business, e-Health Management System, and other areas of network management.

This work needs further study as more variables should be identified and be included in the decision making process. Secondly, the number of requests need to be included too as it directly affect the bandwidth availability. Thirdly, establishing data acquisition for UMAS to fetch all variables of data for frequent updating the AllocatedTime (t) at specific interval time. By doing this, the UMAS will be able to intelligently provide dynamic decisions for users' request and this certainly will contribute to the achievement of a Smart School objectives.

REFERENCES

[1] J. P. L. Kenneth C Laudon, *Management Information Systems*. Seventh Ed. New Jersey: Prentice Hall, 2002.

[2] D. Fickling, "E-Moral Way to Get to the Top", in *Metro*. Sheffield, 2002, p. 11.

[3] NITC, "National Information Technology Agenda". *National Information Technology Council*, Malaysia, 1999. http://www.nitc.org.

[4] J. Mylopoulos and J. Castro, "Tropos: A Framework for Requirements-Driven Software Development", in *J. Brinkkemper and A. Solvberg (eds.), Information Systems Engineering: State of the Art and Research Themes, Lecture Notes in Computer Science*, Springer-Verlag, June 2000, pp. 261-273.

[5] Perini, P. Bresciani, F. Giunchiglia, P. Giorgini, J. Mylopoulos. "A Knowledge Level Software Engineering Methodology for Agent Oriented Programming", in *Proc. of the Fifth International Conference on Autonomous Agents*, Montreal, Canada, 28 May - 1 June 2001.

[6] H. Mouratidis, P. Giorgini, G. Manson, I. Philp, "Using Tropos Methodology to Model an Integrated Health Assessment System", (to appear) in the *Proceedings of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002),* Toronto-Ontario, May 2002.

[7] Michael Knapik and Jay Johnson, *Developing Intelligent Agents for Distributed Systems, Exploring Architecture, Technologies, and Applications*, McGraw-Hill, 1998.

[8] Eral Cox, *The Fuzzy Systems Handbook, A Practitioner's Guide to Building, Using, and Manipulating Fuzzy Systems*. Second Edition, AP Professional, 1999.

[9] Adrian A. Hopgood, *Intelligent Systems for Engineers and Scientists*. Second Edition, CRC Press LLC, 2001.

[10] Carlo Bellettini, Elisa Bertino, Elena Ferrari, "Role Based Access Control Models", in *Information Security Technical Report*, Vol. 6, No. 2, 2001.

BIOGRAPHY

Abdullah Gani obtained his Master of Science from Hull University in 1990. Currently, he is studying for a PhD in Computer Science at the University of Sheffield in the area of network resources control. He is a member of IEEE.

Haralambos Mouratidis obtained a B.Eng (Hons) in Electronics with Computing Science from the University of Wales, Swansea, UK (1999) and an M.Sc in Data Communications from the University of Sheffield, UK (2000). He then joined the Computer Science Department at the University of Sheffield to read for a PhD in Computer Science. His main research interests are: agent technology, agent-oriented software engineering, health care information systems, distributed systems, and mobile information systems. He is an associate member of IEE.

Nasser S. Abouzakhar is pursuing his PhD research with the Centre for Mobile Communications Research (C4MCR), University of Sheffield, Sheffield, UK. Currently, his research area is mainly focused on applying Intelligent and Learning Agents approach for securing Telecommunications and Information Networks. He received his MSc (Eng) in Data Communications from the University of Sheffield in 2000

Gordon A. Manson received his B.Sc. degree in 1971 and his PhD in 1977 from Strathclyde University and a M.Sc. in 1983 from the University of Manchester. He is currently a Senior Lecturer in the Department of Computer Science at the University of Sheffield. He was the assistant director of the National Transputer Support Centre 1987-96. For the last 13 years he has worked with universities and colleges in Malaysia. His current research interests are in mobile agents, dynamic services and the use of Java technology in wireless and wired networks.