

VALIDATION OF CONTEXT BASED SERVICE DISCOVERY PROTOCOL FOR UBIQUITOUS APPLICATIONS

Anandi Giridharan and Pallapa Venkataram
Protocol Engineering Technology Unit Electrical Communication Engineering
Indian Institute of Science, India.
{pallapa,anandi}@ece.iisc.ernet.in

ABSTRACT

Service Discovery Protocol (SDP) is important in ubiquitous applications, where a large number of devices and software components collaborate unobtrusively and provide numerous services without user intervention. Existing service discovery schemes use a service matching process in order to offer services of interest to the users. Potentially, the context information of the users and surrounding environment can be used to improve the quality of service matching. We propose a C-IOB (Context- Information, Observation and Belief) based service discovery model, which deals with the above challenges by processing the context information and by formulating the beliefs based on the basis of observations. With these formulated beliefs the required services will be provided to the users. In this work, we present an approach for automated validation of C-IOB based service discovery model in a typical ubiquitous museum environment, where the external behavior of the system can be predicted and compared to a model of expected behavior from the original requirements. Formal specification using SDL (Specification and Description Language) based system has been used to conduct verification and validation of the system. The purpose of this framework is to provide a formal basis for their performance evaluation and behavioral study of the SDP.

KEYWORDS

Ubiquitous Computing; SDL (Specification and Description Language); Ubiquitous application; Service discovery; context aware.

1. INTRODUCTION

In ubiquitous applications, users are surrounded by a variety of computing devices offering services of different types. As users need to discover and interoperate with such services, service discovery represents a crucial functionality. Ubiquitous computing applications are heterogeneous both in terms of networking infrastructures and interaction protocols. Ubiquitous applications, where not only PCs but also various other devices are connected to networks [1], are expected to offer context-aware services that match user situations and tastes. Because users' needs change dynamically according to the user context such as position or time, an idea to compose appropriate service elements in the network dynamically based on the user context is a promising approach [2][3], as an alternative approach to the conventional way of providing services, where service providers prepare services perfectly in advance. Context information includes device capabilities such as display resolution, processor speed and available memory, network bandwidth and user context such as user location, user preferences, age, and gender, goal, interest etc. Using these context information at hand, a context-aware system can provide mobile users with appropriate services according to the device/user context[4]. Failures of the system is detected by a separate Observer Unit, which observes the inputs and generated outputs and reports its failures in real-time[5]. It determines whether a failure has occurred by comparing

the observed and the specified behavior. Such automated detection of software failures offers a number of benefits. Early notification of occurrence of software failures allows the operators of software controlled systems to take corrective actions before the cumulative effects of failures result in major disruption of system operation. In this work, we propose a framework for the formal specification using SDL based system and conduct verification and validation of C-IOB model based service discovery model in a Ubiquitous museum environment. The purpose of this framework is to provide a formal basis for their performance evaluation and behavioral study. The rest of the paper is organized as follows. Section 2 discusses Context based Service Discovery Protocol. Section 3 illustrates the proposed C-IOB based Service Discovery model in Ubiquitous Museum environment. Section 4 describes role of context coping with failures and validating the errors. Section 5 draws the conclusion.

2. CONTEXT BASED SERVICE DISCOVERY PROTOCOL

When the geographical location of the users device and the geographical location of the services are known in the network, it is possible provides to the user the nearest service. In this situation new applications may appear which could offer the best nearest service that the user needs. The context information is the rules that service discovery uses to select the best service for the user. The location of the users device and the location of services could be the context information of the service discovery. Service discovery provides a mechanism which allows automatic detection of services offered by any node in ubiquitous environment. In other words, service discovery is the action of timely finding a service provider for a requested service. When the location of the demanded service is retrieved, the user may further access and use it. Each service discovery protocol consists of at least two basic participating elements: Server is an entity that offers the service and client uses the service provided[6]. The development of various ubiquitous computing service technologies with a new technological system which enables "the computing environment any time, any place and for anything" in which human centered interface application technologies are added on top of conventional web service technology is progressing.

2.1. Context -Information, Observation and Belief (C-IOB model)

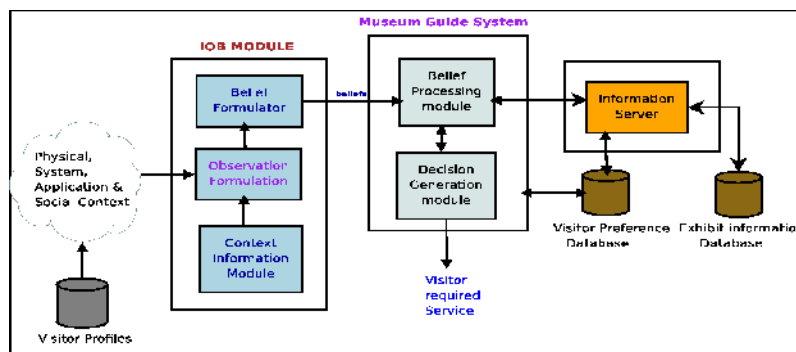


Figure 1. C-IOB Model

C-IOB model is characterized by three components: information , observations and belief module. The real world information can be modeled as a hierarchy of increasing abstract and using intuitive theory to model cause-effect relationship between context information, observation and also predicting the beliefs from these observations, which can be formalized as causal networks. For instance, a learner's belief about causal network structures in context aware domain is characterized by three class variables:context information, observations and beliefs, and there exists causal relations between theses variables[7]. Figure 1. illustrates the C-IOB model.

Context Information: The context information is gathered from physical environment, system environment, application environment and social environment of the visitor. The example of context information is: current location of the visitor, time of the day, room temperature at that particular instant, noise level of the room at particular instant. The behaviors are derived based on the context data or the data from previous history. For example, some of the commonly observed behaviors of user on some transaction: change in location and time, the surrounding noise and temperature, change of device (PDA, LAPTOP, etc.) and its characteristics (based on battery power, processing power, memory capacity), network connectivity (bandwidth, latency, etc), device operating system, application data (email, file transfer). Similarly the data from physical actions like creating, reading, writing, modifying or closing a document. The historical data from the average purchase frequency of the currency, vendor reliability based on past history.

Observation Formulation

The observation is a learning process that can improve the prediction capability of context information and it consists of receiving knowledge of the outside world (various environments) through the senses. An observation in the design of context aware system is obtained from the behaviors exhibited by the context environment. An observation is a summarization of the various observed context information parameters of a user in a particular ubiquitous environment. The observation formulation block formulates the observations depending on the context information of the user. For example, the following observations like visitor is on the move, visitor is going towards new exhibit, visitor device is not working, device network interface is not supported can be formulated for visitor in a museum environment.

Belief Formulator

Primarily, beliefs represent information about the world or an entity, the perceptions received from the external world, and execution of events update the beliefs. The observations made on various behaviors of the visitors will be deduced into beliefs[8]. Some examples of beliefs used in the proposed scheme are: spending more time near exhibit, spending less time and in a hurry, visitor is looking disturbed etc.

3. PROPOSED C-IOB MODEL SERVICE DISCOVERY PROTOCOL

Context Information: The context information is gathered from physical environment, system environment, application environment and social environment of the visitor. The example of context information is: current location of the visitor, time of the day, room temperature at that particular instant, noise level of the room at particular instant, etc. Figure 2. shows C-IOB Model Service Discovery protocol in ubiquitous application.

The context information can be divided into four categories:

- · Physical Environment context: which includes the context information parameters like location, time, temperature, noise level, pressure, position, orientation, etc. ,[9].
- · System context: such as device being used, operating system present in the device, supported network interfaces, output modes of the device, screen characteristics of the device, etc.
-

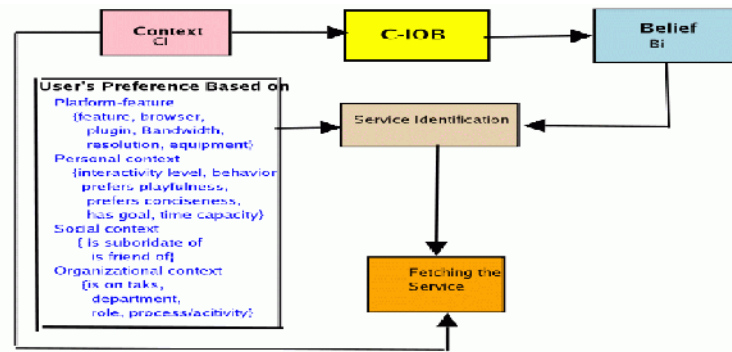


Figure 2. shows C-IOB Model of a Service Discovery Protocol used in a ubiquitous application.

- Application context: It includes the type of application, different data types in application, status of the application, resources required by the application, etc.
- Social Context: whose parameters include social behavior of user, preferences of the user, social identity, social trust on the user, etc.

Context information: can be represented as given below: $C_{(nk)} = (c_{11}, c_{12}, \dots, c_{1k})$, $n=1$ to 4 . where $C_{(1k)}$ represents the set of context information of physical environment context, $C_{(2k)}$ represents the set of context information of system context, $C_{(3k)}$ represents the set of context information of application context and $C_{(4k)}$ represents the set of context information of social context.

Observation Formulation: An observation is a summarization of the various observed context information parameters of a user in a particular ubiquitous environment. The observation formulation block formulates the observations depending on the context information of the user. For example, the following observations like visitor is on the move, visitor is going towards new exhibit, visitor device is not working, device network interface is not supported can be formulated for visitor in a museum environment[10].

Belief Formulator: Primarily, "beliefs represent information about the world or an entity, the perceptions received from the external world, and execution of events update the beliefs". The observations made on various behaviors of the visitors will be deduced into beliefs. There exists pre-established relationships between observations and beliefs, these relationships are of types "one- to-one, one-to-many and many-to-many". Some examples of beliefs used in the proposed scheme are: spending more time near exhibit, spending less time and in a hurry, visitor is looking disturbed. The belief formulator is a component of C-IOB which collects various temporal and symptomatic context information parameters from the environment. At particular time of the context information, corresponding observations are generated. The beliefs are deduced based on the new and available observations over a context. The belief formula which is used to represent individual belief is given by: $B = (b_1, b_2, \dots, b_n)$. Where b_i is a i^{th} beliefs in the set of beliefs, B , as literals and variables used to represent various observations on which the belief will be reasoned. The given belief representation is compatible with the working of belief formulator.

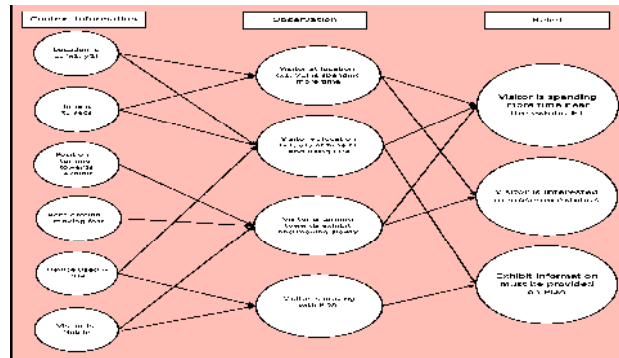


Figure 3: Causal graph of the model

Service Identification: Service identification is one of the challenging steps in the Service Oriented Development life cycle. Identification of the service mainly depends on the system's lookup status which is offering the service, components of such systems, identifying the suitable component in the system which provides the required service, type of interface mechanism provided in the service providing entity, types of inputs and outputs of the servicing entity. Also the orientation, etc.[11]. Identification of service depends on the present context at which the service has been requested. This identified service must be efficient in all possible ways when compared to other servicing entities in the look-up state.

Fetching the required service: Once the service required has been identified, it had to be fetched from the service providing entity. The service fetched will be based on the context, availability of the service in look-up table and current status of the service providing entity. The fetching of service also depends on the type of network interfaces used by both service requester and service provider which directly affects the QoS of the service provided. Figure 3. gives the causal graph of the model representing contextual information of environment and corresponding observation. Table 1 gives the sequence of exhibit information service discovery using the C-IQB module.

<i>Context Information</i>	<i>Context Information at particular instant</i>	<i>observation</i>	<i>Belief</i>	<i>Personalisation parameters</i>	<i>Service</i>
Location	at (x_1, y_1)	Visitor is at (x_1, y_1)	Visitor is spending more time near exhibit	History- Visitor is visiting museum for first time	Provide exhibit information to visitor
Time	t_1	at time t_t			
Position	turning towards exhibit	Visitor turning towards exhibit	Visitor is near exhibit E_1 and	Interest - He is interested in science related exhibits	
Acceleration	Moving slowly	and moving slowly	turning towards it		
Exhibit sensor	ON	Exhibit sensor is ON	Visitor movement	Personal profile - he is a well educated	
Visitor device	Laptop/PDA/Cell Phone	& visitor movement is slow	is slow & he is using PDA	person with good understanding level	
Visitor Preference	interests	Visitor preferences are identified & he is using PDA	Using visitor preference level		
Mobility of visitor	mobile/stationary	Visitor is mobile & he is moving in	of visitor has been decided	Preference - he likes to know more about the exhibits	
Direction	North/South/ East/West	north east direction			

Table 1: Sequence of exhibit information service discovery using the C-IOB module.

3.1 . Flow Diagram of Context aware Service Discovery Protocol

Context aware Service Discovery Protocol has two important entities:

1. Service Management Node (SMN): interacts with other SMN and maintains a repository of context data within a cluster of nodes.
2. Service Assistant Node (SAN): takes over the role as an SMN and it may also act as proxy between SMN and IP devices. SMN is again split into two parts. One part is the lower layers is called Service Management Module responsible of interacting with the external clients devices and services devices, by using legacy Service Discovery protocols such as UPnP. The other one is the Service Discovery Adaptation Layer , which is the interface between these protocols and the upper components. The upper layers

contain the main modules, which make it possible to filter the best service as in figure 4. and flow chart representing operation of Service Discovery Protocol is shown in figure 5.

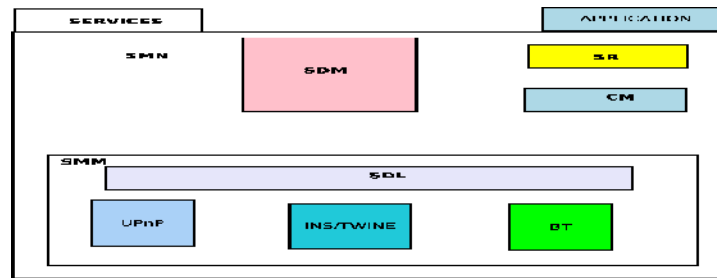


Figure 4: Service Discovery Components

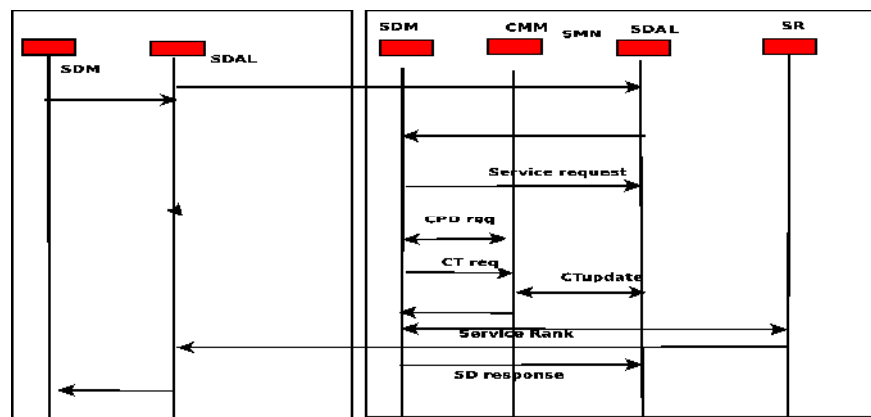


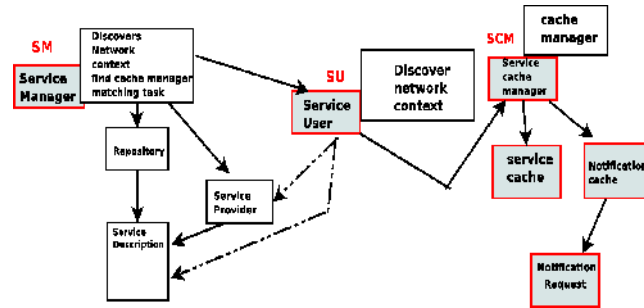
Figure 4: Flow chart representing operation of Service Discovery

3.2. Service composition framework of C-IOB model in U-museum environment:

Interaction between various components of context aware Service Discovery Protocol is shown in figure 6. The context is a kind of information that makes it possible to the Service Discovery filter all the connected services and choose the best. The first time visitors to museum must get registered with the ubiquitous guide for museum guide system. If the visitor is a well known by their profession or a celebrity, the system search their web-site and collects the visitor's profile, interest, preferences and available time for visiting the museum exhibits. Otherwise the system initiates the registration process to collect visitor's information online. This information is more useful when providing services to the visitor based on his/her interest, preference and time spend in the museum, etc. To accommodate user requests, the portal employs a composition framework to coordinate atomic, disparate services. There are different services available in museum see Figure 6. Interaction between various components of context aware Service Discovery Protocol.

- DirectionsFinder: Providing exhibit information, path to next exhibit.
- IdentifyingService: Identifying the nearest catering and reservation service
- MedicalService: Finding Medical service nearby, ambulance service etc.
- EmergencyService: Emergency fire exit path and calling a fire extinguisher service, ambulance service etc. This Identification of service is done by taking the current beliefs formulated by considering the present context of the visitor. Table below gives the belief driven service identification. After identification of the visitor required service, the system will also provide the

path or URL address to the service ,where it is available. The different services available in ubiquitous museum are stored in different servers which are fetched via the path or URL



addresses.

Figure 6: Interaction between various components of context aware Service Discovery Protocol

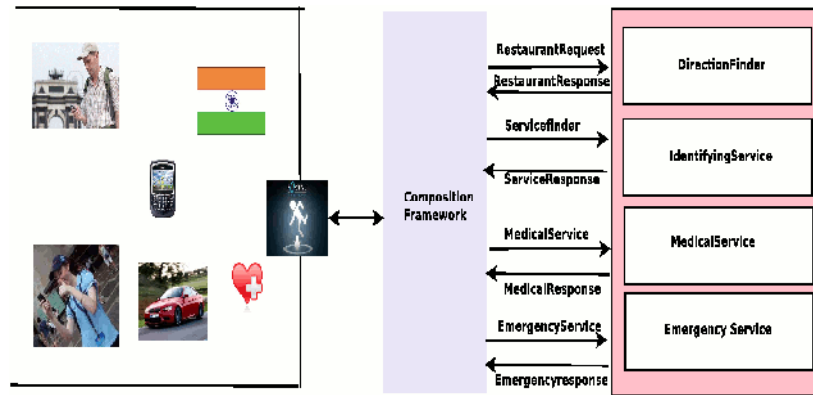


Figure 7: Context aware application finder

Case 1: Catering service location: Assume that visitor is in a museum, and time is mid-day and the belief formulated will suggests that "visitor needs the catering service", so system has to provide the catering service. To provide the catering service to the visitor, system must know the visitor preferred dishes, which is determined from the visitor profile. Once his/her preferred dishes are determined, then C-IOB system starts searching for the nearby restaurants , where these dishes are available. Once the search is complete, the system decides the best restaurant where visitor preferred dishes are available according to the visitor preference and also system reserves a table in the restaurant for the visitor to have lunch. After completing these procedures, the system informs the visitor about the restaurant where dishes of his/her choice will be available within the budget and provides the path to that particular restaurant.

Case 2: Fire emergency exit: If there is a fire in a museum or in any such public places, usually people are more distracted and no cool mind thinking due to the situation. In such a scenario guidance must be given to the visitors to the emergency exit so that they can come out without any problem. The occurrence of fire emergency can be determined by different context information such as room temperature, humidity, noise level in a room. The corresponding observations for this context informations will be sudden increase in room temperature, increase in humidity level and noise level which leads to the belief that there is some fire disaster. Once this belief has been formulated the system can decide that the visitor needs emergency fire exit

service and it provides the path to emergency exit as well as it informs the fire extinguisher service to control the fire. Figure 7. shows the context aware application finder in case of mobile and stationary tourist.

4 Role of Context coping with failures:

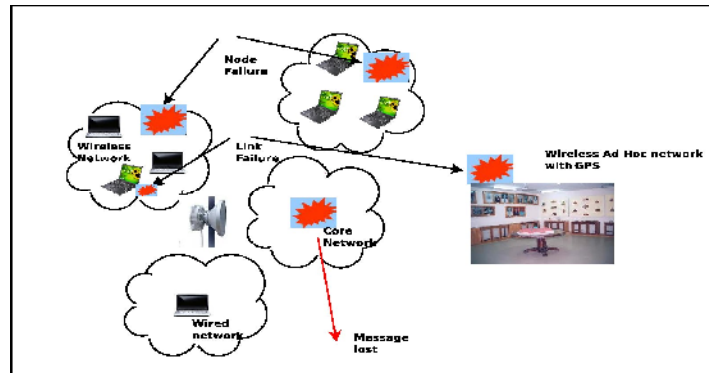


Figure 8: Failure of Nodes, links & message loss in a U-Museum environment

Context has a major role in the approach. A framework needs to acquire context from different context sources in a disciplined way. A mechanism is also needed to describe what action to take when the tourist enters a certain context. Another important challenge in providing a context aware service composition facility is dealing with various failures. For instance, failures may occur at composition time, as a result of context fails or changes with missing service descriptions. Failures may also arise at run-time, for example, because of the loss of network connectivity, node failure, link failure, etc. The design of the framework must ensure its ability to operate under increasing load, increasing complexity of requests and increasing size of resulting composite services. Figure 8. shows failure situations in Ubiquitous museum environment. In case of major power failure after the discovery phase has completed, services and notification requests are registered, most entities lose some internal state: all nodes lose discovered SCMs; SUs lose SDs for service previously discovered. A number of different failures may occur during the composition of service process:

- **Node failures**, link failures, network level disconnections, Sensor failure, service discovery failures, service execution failures.

System attempts to locate a different provider for the context type for which there is no availability of context information. If no providers are available for the context type in question the system acquires historical contextual data from the Context Service. It may also construct a composition request based on a previously executed task.

Discovery failures: Service discovery failures may occur for the following reasons:

- Inaccessible Service Registry, Network disconnection, System overload, unavailability of requested service in the registry, unavailability of suitable service instance, Mismatching of service operation. To avoid the Service Registry being a single point of failure, the system has been designed to operate with multiple service registries. This can be achieved both by deploying a number of replicas of the same registry, as well as by mediating among a number of different registries. Unexpected environment changes: Environment changes may be due to: Different

composite service from that designed & Recomposition Fails. Finally, if the recomposition process fails as well, control is passed back to the composition request management layer, in which a composition request is modified or a new one assembled.

Service unavailability: Services may become unavailable for the following reasons: · Network Disconnection Failure can be handled by service instance, composite service.

These are some of the techniques used to overcome failures:

- Number of cached copies are backed up to avoid the delays inherent in rediscovering a suitable service instance.
- New service is fetched from the Service Registry.
- The system may also opt to run redundant instances of composite services.

4.1. Consistency Maintenance in the C-IOB model during node, link failure.

In the proposed framework goal conditions are represented in Specification Description Language (SDL). This allows for easy import of goals into the software component that stores them. Each goal condition is a specification. Belief conditions: that form the composition request may result from the task intention of the user or from the Context of the user. Goal: Any goal condition that purely describes a users task intention, independent of the current context, is a core goal. Examples are (directionfound) goal conditions.

Minimal goal: The minimal goal condition that needs to be satisfied to achieve a viable solution for a given composition request is termed a minimal goal. Context goal tags are introduced for representing context goals, which are goal conditions that arise in a specific context. Each tag has three parts, a context type, context value and goal condition.

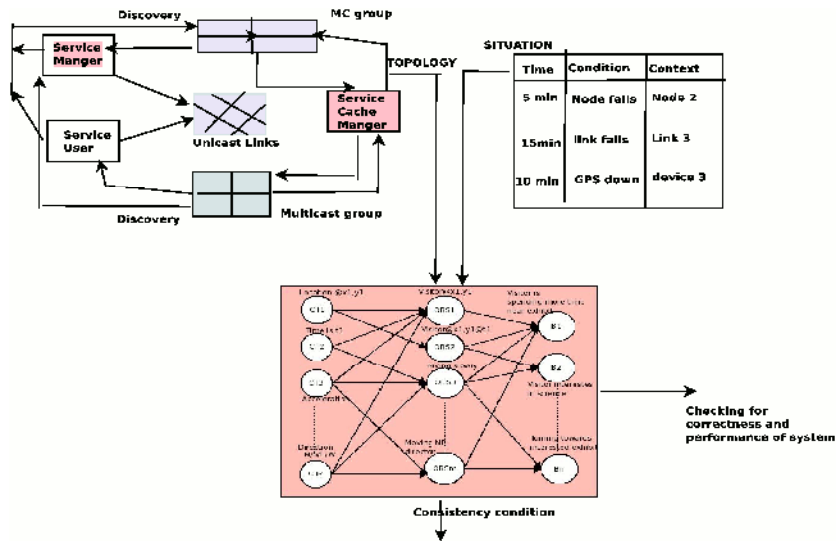


Figure 9. Validation for correctness

Following are the description of construction of a composition request:

User selects a desired task, Corresponding core goal conditions are fetched from the Goal Service, The Context Proxy fetches the current context data from the Context Service, · Construction of tags. The Context Service may fail to provide access to context data, for example,

because of a failure of respective context provider as a result of sensor unavailability or network disconnection. If a desired context value cannot be retrieved from the Context Service, during construction of tags, the Context Proxy attempts to obtain a past context value from the Context Service. The Context Service tracks past contexts in which the user has submitted composition requests. The software component that implements the Context Service has the facility to cache the historical values. context value of this type. Secondly, it supplies the most frequently occurring context value of this type. This historical context data can also be used to generate probabilistic predictions about current and future contexts. Context value orderings facilitate the substitution of related context values. By substituting context values new context goals are created. This is useful for transformation of unreachable context goals into ones that can be solved. Figure 9. shows validating error and correctness in the U-museum environment. Specification and Description Language was used to specify the functionality of each component and communication between them to facilitate fault-tolerant, context aware service composition[12].

Context Entity: Each composition request is a formal definition of the users task intention. The Context entity has been modeled to receive context signal c_{req} from Ubiquitous museum environment (based on user's location, device, user's movements, gestures, behavior etc.) Raw Context data received from U-museum environment undergoes acquisition c_{acq} and processing for appropriate data format c_{pro} .

Observation Entity: Context data signal c_{data} is later sent to Observation Entity as show in figure 10. Entity performs context analysis and reasoning C_{analy} and Observation formulation is carried out $O_{bsformu}$ to formulate Beliefs. This Belief data B_{data} is sent to Belief Entity.

Belief Entity: This Belief entity is the abstract service composition management module, that has abstract plans that executes specified service instance. Receiving Belief Data, Belief entity automatically identifies the the service S_{id} with help of Service Discovery scheme without user's intervention and fetches F_{ser} the required service from appropriate service provider entity and presents it to the Visitor as shown in figure 11.

Figure 12 shows the Message sequence charts that represents the transaction between various components generating belief under normal working conditions. By introducing activities, we can improve the system's response by returning different URL's according to the current user situation. Three handlers like Location of the user, Navigation type and route and Activity to be carried out by user is configured as shown in Figure 13.

5. Simulation and Results

For Consistency analysis of C-IOB model , simulation scenarios were done to check the behavior of C-IOB model in Ubiquitous Museum environment. A experiment room with many statues and photos dispersed at fixed locations. As the visitors visited the room, simulation were conducted considering various failures like node, link etc. and latency time of communication was noted at each instant.

All the Performance metrics were measured like total number of hits, when the tourist viewed certain statue in the room and number of misses, when the the information was not available immediately to the visitor due to node, link failures were also noted. Graphs for number of hits and number of misses are as shown in Figure 10. Figure 11 gives number of hits, when tourist were successful in viewing information without any delay.

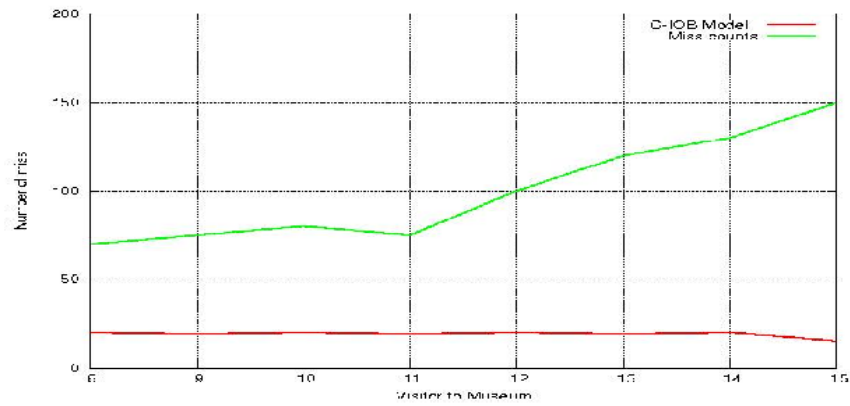
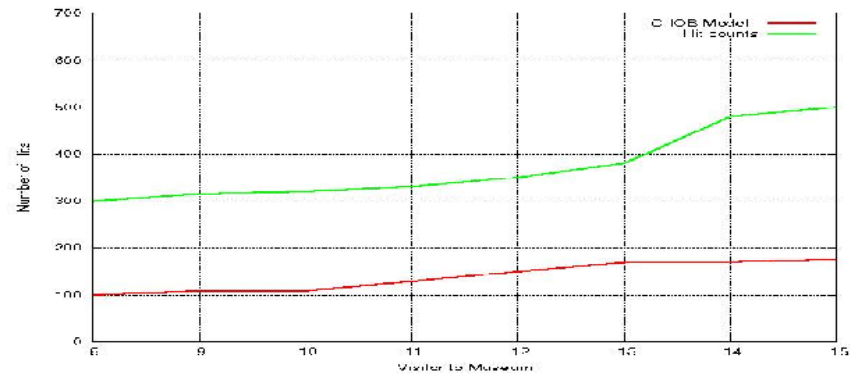


Figure 10. Plot of number of visitors to museum vs number of misses
 Figure 11. Plot of number of visitors to museum vs number of hits



Success ratio and delay were also observed. C-IOB model provides more relevant services for visitors with reduced latency as seen in the figure 12.

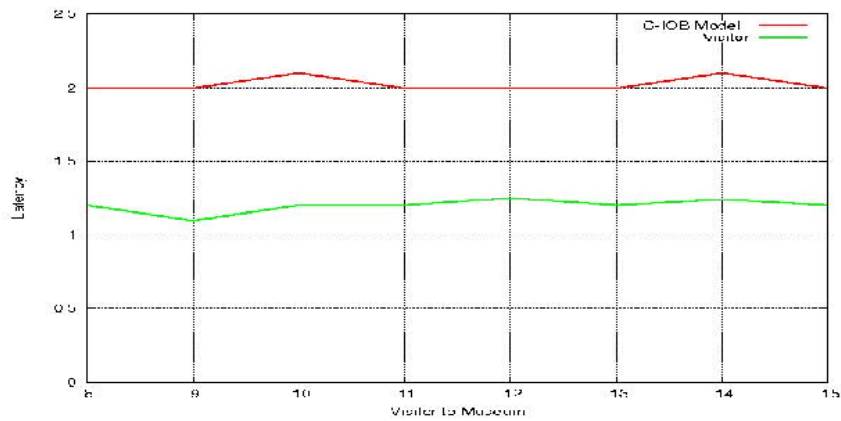


Figure 12. Plot of number of visitors to museum vs Latency

4.2 Validating Errors using Specification and Description language:

Specification of System context , system process (protocol behaviors) and procedures using Specification and description language is shown in figure 13 and 14.

Message sequence chart , a graphical specification language that graphically displays interaction behavior of the C-IOB system as shown in figure 15.

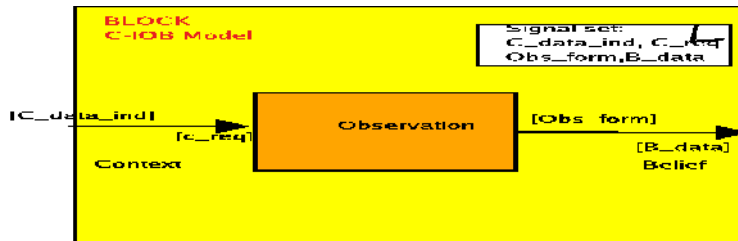


Figure 13. System context

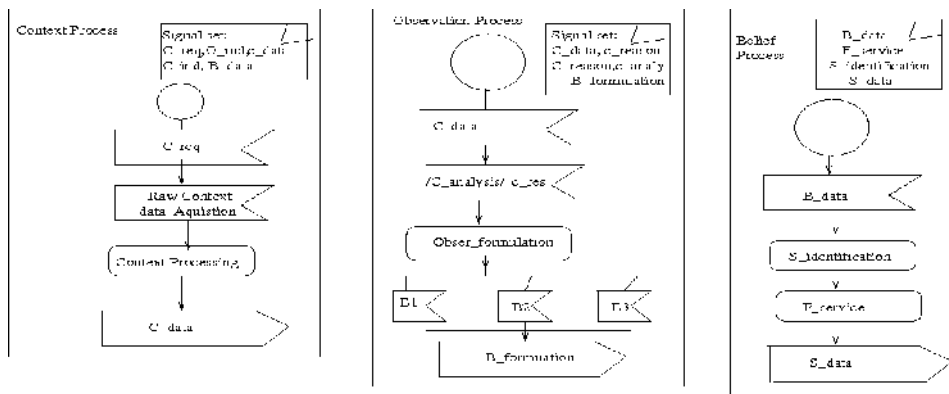


Figure 14. System Process (behavior of protocol)

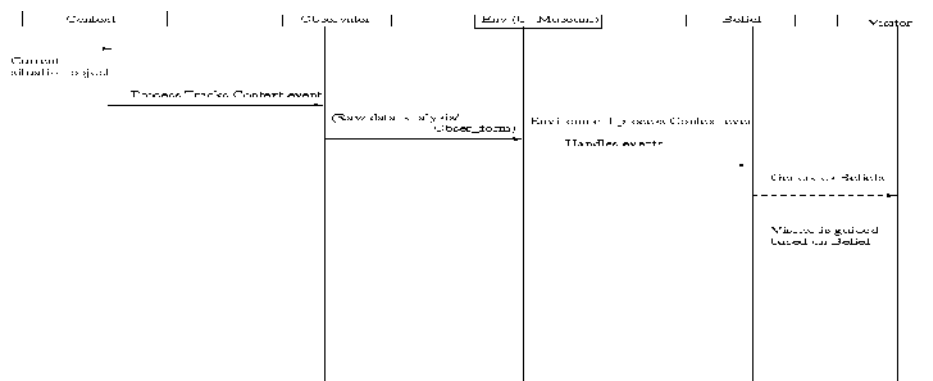


Figure 15. Message sequence chart showing normal operation of system

System tracks the context features as they change due to certain communication failures. C-IOB model keeps track of history of the visitors (e.g. the visited nodes) existing features (e.g. the weather conditions tagged as good or bad), navigation map etc. Thus, C-IOB model supports

using a detailed Message sequence chart of all process state transitions and signals exchanged between processes and between processes and the environment. The Validator can be used both to explore the system in an unguided mode looking for errors, and to perform a guided exploration in order to check whether a given system-level MSC is satisfiable.

5. Conclusion

Observer observes the inputs and outputs of the target software system and automatically detects its failures. To determine whether the observed behavior is correct, the Observer internally executes (interprets) a Reference model derived from the specification of external behavior of the target system. In our, C-IOB model in case of node failure or link failure or if web server is down, Adaptation environment has a set of handlers that are in-charge of reacting to these context changes and generates its specific context- aware behavior accordingly based on the user's navigational behavior and history. C-IOB model provides more relevant services for visitors with reduced latency. The paper considered the case, when the specification is expressed in a formalism based on communicating extended finite state machines, specifically the ITU-T SDL.

REFERENCES

- [1] M. Weiser, Ubiquitous Computing, Communications Of ACM 36(1), 75-84 (1993).
- [2] S. Gribble, et al., The Ninja Architecture for Robust Internet-Scale Systems and Services, Special Issue of Computer Networks on Pervasive Computing, 2000.
- [3] M. Takemoto, et al., A Service-Composition and Service-Emergence Framework for Ubiquitous-Computing Environments, SAINT2004, pp. 313-318, Jan. 2004.
- [4] Sejong Gu, Hongzhou Shi, Jian Ye, Zhenmin Zhu, "A Service Discovery Framework for Ubiquitous Computing", 8th Intl. IEEE 2007
- [5] Tianyin Xu, Baoliu Ye, "A Gnutella Inspired Ubiquitous Service Discovery Framework for Pervasive Computing Environment", Tokyo 163-0914, JAPAN, 2008 IEEE.
- [6] "Service Advertisement and Discovery in Mobile Ad hoc Networks", Bethlehem, PA 18015
- [7] E. Guttman, C. Perkins, J. Veizades, and M. Day. "Service Location Protocol", Version 2, Request for Comments (RFC) 2608, June 1999.
- [8] B.Miller and R.Pascoe, "Salutation Service Discovery in Pervasive Computing Environments", Technical Report, IBM White Paper, February2000.
- [9] Paramesh C. Upadhyay, and Sudarshan Tiwari, Sant Longowal , "Fault Tolerant Distributed and Fixed Hierarchical Mobile IP", Institute of Engineering and Technology, India and Motilal Nehru National Institute of Technology, India, IJU 2010, Volume 1.
- [10] Anis Ismail, Abd El salam Al Hajjar, ziad ismail, Liban and Telcom paristech, "A New system architecture for pervasive computing", Liban and Telcom paristech, France, IJU 2011 volume 2.
- [11] Marzieh ilka, Mahdi Niamanesh and Ahmad Farrah, Paam Noor univerty, "A Group-based Method for Context-Aware Service Discovery in Pervasive Computing Environment", Tehran, Iran, IJU 2012.
- [12] Alireza Souri, Fault Tolerant Location-Based Service Discovery Protocol (LBSDP): A New Approach in Vehicular Networks, American Journal of Scientific Research, ISSN 1450-223X Issue 57 (2012), pp. 84-89. © EuroJournals Publishing, Inc. 2012

Authors

Venkataram Pallapa received his Ph.D. Degree in Information Sciences from the University of Sheffield, England, in 1986. He is currently the chairman for center for continuing education, and also a Professor in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. Dr. Pallapa's research interests are in the areas of Wireless Ubiquitous Networks, Communication Protocols, Computation Intelligence applications in Communication Networks and Multimedia Systems. Dr. Pallapa is the holder of a Distinguished Visitor Diploma from the Orrego University, Trujillo, PERU. He has published over 150 papers in International/national Journals/conferences. Written three books: *Mobile and wireless application security* Tata McGraw-Hill, *Communication Protocol Engineering*, Prentice-Hall of India (PHI), New Delhi, 2004 (Co-author: Sunil Manvi), and *Multimedia: Concepts & Communication*, Darling Kinderley(India) Pvt. Ltd., licensee of Pearson Education in South Asia, 2006. Written chapters for two different books, and a guest editor to the IISc Journal for a special issue on Multimedia Wireless Networks. He has received best paper awards at GLOBECOM'93 and INM'95 and also CDIL (Communication Devices India Ltd) for a paper published in IETE Journal. He is a Fellow of IEE (England), Fellow of IETE(India) and a Senior member of IEEE Computer Society.

Anandi Giridharan, received MSc(Engg) from Indian Institute of Science. She is currently working as Senior Scientific Officer in ECE Department, Indian Institute of Science, Bangalore. Her Research Interest are in area of Ubiquitous learning, Communication Protocols and Multimedia systems.

