Context Aware Servers and Ubiquitous Application Development

Junaid Ahsenali Chaudhry, Seungkyu Park
Ajou University, Department of Information and Communication Engineering,
Phone: +82-31-219-2532, Mobile phone: +82-10-3143-4751, Email: junaid@ajou.ac.kr

Abstract:
In this paper we discuss the development of ubiquitous applications in a context aware environment. Considering the requirements of ubiquitous smart space, we discuss execution of automatic cooking and seamless interaction of machines with humans as an exemplary scenario for understanding. The purpose is to discuss the service integration and use of context servers in formation of ubiquitous applications. The context awareness of a task helps us divide it into various tiny time slots and allocate each device its role. The device performs its role and reports back to the context aware server and waits for new instruction in next slot. When a new device enters in the area of influence, it notifies about his resources and the system allocates tasks for it in later planning. We try bridging the gap between man and machines using the explicit knowledge available in the form of active context. The principle objectives include the creation of vivid picture for developers to create a smart space as a demo of our project and make a killer application in the future ubiquitous society.

I. INTRODUCTION

The vision of omnipresent ubiquitous technology is to provide service to every request initiated within smart space (intra) and requests paged from outside (inter) [1][2]. The"off-the-desktop" computing emphasizes on embedding the interface in whole physical world in such a way that the user feel it seamless [3]. Those users can be countless devices (mobile or static), speckles and human users. Building on top of the ubiquitous culture and infrastructure i.e. OSGi [10], the Service Oriented Platform (SOP) is considered to be strong candidate for future ubiquitous networks [11]. All those services, residing at the platform, combine to make applications that directly interact with the users [4]. As the ubiquitous heterogeneous networks provides services to many type of devices like cellular phones, PDAs, laptop computers, and many others. The number of devices attached to the network increases drastically. On top of that great amount of work load, the quick application production is very tough. Before that, the home automation and networking of the devices was there. But improvement among those automated atomized solution and service availability was confined to that local place. Expanding the control will increase the usability and will bring ease in life that is the essence of ubiquitous computing [7]. We in this paper present a solution to this problem. We mention some services and on the bases of those services, end user applications are generated. An exemplary scenario is provided for better understanding. We have simulated some parts of the system (intra system) and mobile user connected through Session Initiation Protocol (SIP) through internet [5].The target area for the application, we are providing, is for smart homes and that is our objective for the development of a prototype in our project.

II. OVERVIEW AND ARCHITECTURE

In this section, we present an overview and architecture of the system. The components are chosen considering the delicacy of the system and maintaining the original layout of present kitchen environment. The domestic gateway manager is the managing service present on the gateway and its job is to manage the services that are being used through a gateway. The framework constitutes the entity that provides the operational platform to the services to negotiate with the devices and among themselves. The active profile manager runs throughout the process and it monitors the whole system and reports to the manager about the changes happening in the dynamic system. The recipe manager, interpreter, actuator unit, port manager, grocery manager, and application manager are the core components of the system. The details about them are given in the later sections. The cooking application runs on top of all and it is provided by the service provider. The tasks related to it are system update containing recipe update, protocol update and others.

1. RECIPE MANAGER

The recipe manager is a part of the service that contains the whole record of the recipe pool in XML format. The vendor is responsible for recipe formation. Some XML parser parses the whole recipe file and inference engine allots the jobs to individual device by keeping the available slots of that device in context and the nature of job. The benefit of keeping the recipe into the XML format is to get the ease in transportation of the data and when it is needed to use, it can be directly fed to the machine instruction translator unit or recipe dress up manager. So it will save time and provide efficiency to the system. When the remote user requests the dish, the interface is shown in figure 6. The dish name can be in the form of pre-filled items list that the cooking server can provide you. Recipe lookup button checks the availability of the dish and its contents. If the user wants to specify the recipe or the contents destination through the insurance of some third party, he can do that at this interface. The jobs are divided into time slots and during those time lots the system activates many threads and monitors the performance of each device. The manager can download updates and new recipe which can also be located on user request recipe. After all the inputs are prepared, the system gets busy for the specified time to make dish as requested. The remote user is connected with the server using Session Initiation Protocol (SIP)[5].

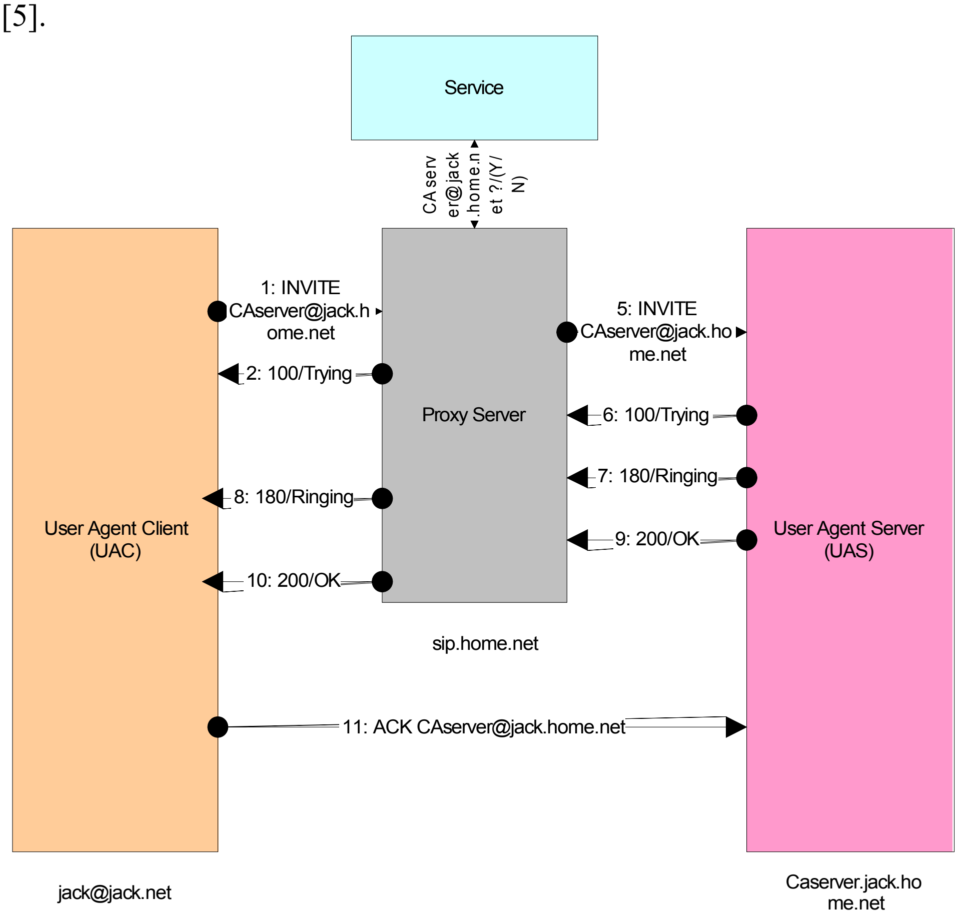
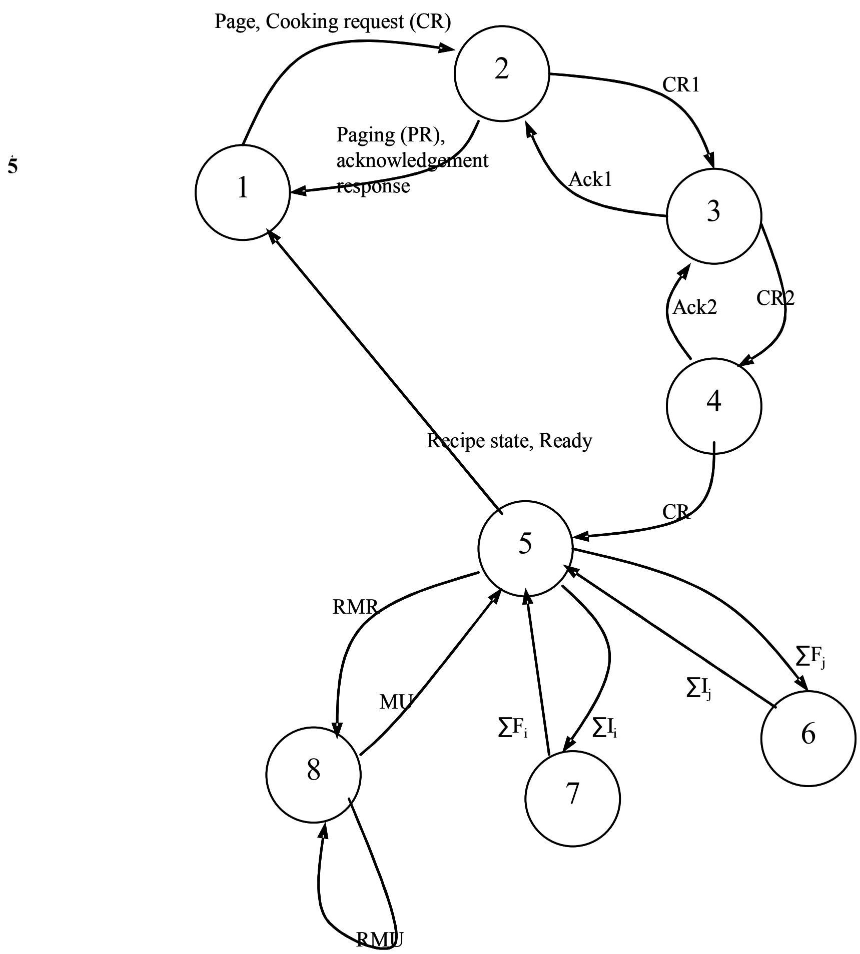


***Fig1**: Session Initiation Protocol for Remote user connectivity*

For connectivity, each device is SIP enabled. For session initiation one device becomes client and the other one becomes a SIP server. The scenario is shown in the figure 3. The prototype for the user connectivity is a service that has been designed to run on an iPAQ using the Jeode JVM, although it should work with any other PersonalJava compliant JVM. This service waits for an iPAQ-alert message to be posted to the event heap and then displays the *text* attribute of the message in a popup window. If further events are posted before the popup window has been closed, these will be displayed in sequence after the window has been closed.



User	RMR	1- $2\sum_{i=1}^{i=\infty} \sum_{j=1}^{j=\infty} F_{(i,j)}$
Remote Host	- Recipe Messaging	(Feedback)
Home Manager	Request	2- $2\sum_{i=1}^{i=\infty} \sum_{j=1}^{j=\infty} I(i,j)$
Home server	MR	(Instructions)
Cooking server	- Managed Recipe	3- $2\sum_{i=1}^{i=\infty} \sum_{j=1}^{j=\infty} U_{(i,j)}$
Actuator	RMu	(Uncertainty)
CA Manager	- Recipe Makeup	
Recipe Manager	F- Feedback, Instruction	

***Fig4:** Automata model for Integrated Application*

2. ACTUATOR UNIT

We have divided the implementation plan into two parts (1) inside house and (2) outside house. The first of the two includes the communication architecture of the application itself and its communication with the servers. The second one specifies the infrastructure outside the house considering the remote host and its request remotely to the cooking sever. In figure 5, we have shown model architecture of a kitchen containing a U2 Cook. The shelves on the right side contain the grocery chambers. On the left side the pots chambers are shown and in middle of that the stove, smoke collector and a place for robotic arm are shown. A conveyer belt is shown at the left bottom side, where the pot is placed and that is moved to the table along with the cooked food in it. The collector tray is placed between the chambers and the stove. That collector tray collets the grocery items and pots from their respective chambers and then injects them into the main process.

3. RECIPE DRESS-UP MANAGER

After having the recipe stored in the recipe knowledge base, there is a need to translate that into the executable instructions so that the actuator unit may start working on the preparation process of the food. We need to know about the resources we have. Knowledge on the resources are related with how many and what kinds of pans we have and what the status of the grocery is. Do we have all the hardware ready to start a new job? To answer such questions we have a context manager. Following is the predicate logic form in which the knowledge is represented describing the role allocated to each device in different time slots.

Role00125413e (001589GHJ, DGHJ003)

The context manager keeps track of very item in the kitchen. Whenever there is some need to start a new job, the context manager will be asked about the ingredients and their resources required. The manager will reserve the resources for that job and all the communication to the devices will be done via context manager. After requesting adequate resources from context manager, the dress-up manager prepares the job allocation tables. In those tables the jobs of all the system are assigned with reference to each time slot. Every time slot contains information to each and every device in contact. There can be two states of each device, e.g. 1- device in context and 2- device out of context. Every attached device is sent the instructions in each time slot. The sample instructions are shown in the table shown in table 1.

Device ID	Time Slot	Instruction
C001455D	*001589GHJ*	*DGHJ003*
C001456D	*001589GHJ*	*DGHJ013*
C001457D	*001589GHJ*	*DGHJ005*
C001458D	*001589GHJ*	*DGHJ001*
C001459D	*001589GHJ*	*DGHJ007*
C001455D	*001590GHJ*	*DGHJ004*
C001456D	*001590GHJ*	*DGHJ014*
C001457D	*001590GHJ*	*DGHJ006*

Table1: *Job Allocation to devices in time slots.*

During the job in process time there is a provisioning agent, monitoring the whole process. We shall discuss these issues in security section.

4. CONTEXT AWARE SERVER

Network resource availability varies with time not only due to dynamic demand from the users but also dynamic channel conditions. An RG needs to dynamically assess the availability of system resources and allocate them accordingly to meet QoS requirements of supported applications [6]. The context aware server is a *mini gateway* for kitchen. The message passing nature of the application and tight time slot marking will ensure the Quality of Service so

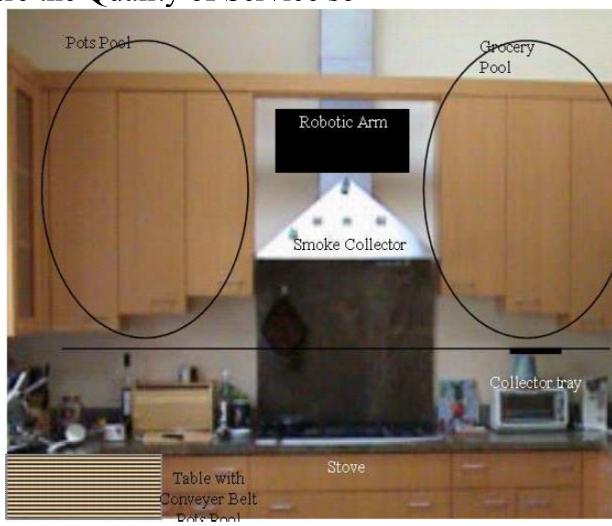


Fig. 5. Model Kitchen containing U2 Cook

Context Aware server is important in this scenario. Typically, devices/components will be provided by different vendors and could be installed in the kitchen at different points of time e.g. pots, grocery items etc. The probability of the devices leaving the ’effective area’ is high too. This requirement can be satisfied by”Plug and Run” environment. It is desirable that this should be a self configuring environment. For this the UPnP support is important and self configuration demands the ubiquitous driver access service that may enable a system to be detected and attached and that can make device work seamlessly. A device attached to the network is desirable to contain the following characteristics. Many of these depend upon the vendor. The more the probability of presence of these features into the devices is, the more efficient the network will be

1- Discovery of other devices on the network. The number of hand-over and take-over happening.
2- Discovery of the new interfaces and search oftheir support present for the network on the vendor site or at some other location.
3- Auto-assignment of Ubiquitous ID to new devices.
4- Profile management of each device coming in and handing over the profile to the new gateway if device leaving the network.
5- Self-provisioning and providing of data to other devices to set their state based on addition of new devices to the network.

Based on these requirements we explored various alternatives for discovery and initialization. While Jini is an attractive proposition, the UPnP suite of protocols appears to be more intuitive to use. It is obvious that suitable ”Device Control Protocols” (DCP’s) would have to be defined for the various components of U2 cook.

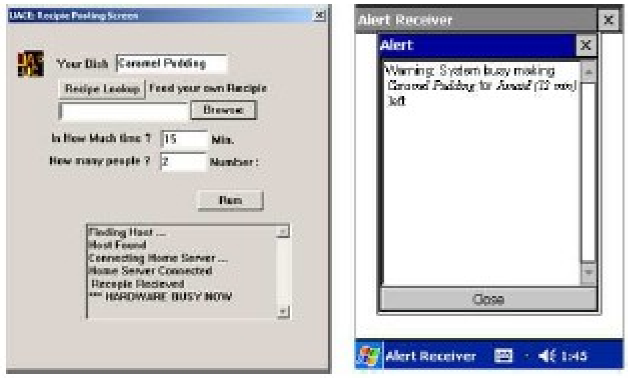


Fig. 6. Recipe Request and lookup Interface

This would allow various vendors to communicate with each other while still allowing for vendor differentiation and value addition. These features (addressability and discovery) fit to our scenario to assign Ubiquitous ID to the devices and self configuration features in the network. Once discovery has been performed, UPnP allows for the exchange of device description in an XML based format which includes URL's for device control. This mechanism can be used by the various devices in U2 cook to describe them and obtain a description of other devices on the network. This would allow the newly added device to initialize itself with network specific parameters. In addition to a suitable discovery and initialization protocol the controller needs to access interfaces on the devices to accomplish a task. This can be completed by passing an appropriate message to the device. A suitable abstraction for this is the device control interfaces as described by UPnP. This allows for the device interfaces being invoked by sending SOAP messages over the network. In the absence of a message passing and event notification mechanism the device management and synchronization of the whole process will be very tough or nearly impossible. So we need some message passing and even notification architecture so that all the devices that are involved in the active operation should be synchronized and in 'ready to perform' state. This will lower the delay in the process and will increase the efficiency.

5. GROCERY MANAGER

A lot of research has been going on at this issue of grocery management [8][9]. For managing the grocery items we have a grocery managing mechanism. The grocery is stored in the chambers shown on the right side of figure 5. A chamber has some area *A* and height *h*. The area of the aperture is *a* and the amount of item coming out is *u*. The structure of each chamber is shown in figure 8. Now according to Torricelli model we can predict the amount of grocery item coming out from the chamber. The Torricelli model for this problem is based on two physical concepts. First one is The Bernoulli relationship between pressure, *p* density, and speed, *u* for the matter along a stream line

$$p = \frac{1}{2}\sigma u^2 \quad ...(1)$$

The second one is the relationship between changes in pressure over the height column and gravity (acceleration *g*), density and the height *h* of the column

$$p = \text{pgh} \quad ... (2)$$

Equating 1 and 2 gives a relationship between height of the column above the hole and the speed at which it spews forth, *u* is equal to square root of *2gh*. The volume of grocery item lost from the bucket must equal the flux through the bucket's hole (with area *a*), and using the fact that (for a bucket with regular sides and constant cross section in height) the relationship becomes

$$ua = \frac{dV}{dt} = \frac{d[Ah + Vo]}{dt} = A\frac{dh}{dt} \quad ...(3)$$

Now from the relationships we have the following formulae helps up estimating and thus controlling the amount of grocery item, coming and then being fed for filling by the backup chamber. When some quantity of grocery item is drown out from the chamber, delta u is the change in quantity and *u* is the amount coming out. When some amount of grocery item is needed, instructions are sent to the aperture. Since time t in which the aperture was open has direct relationship with the amount of quantity drown out from the chamber, we can say that t is the function of change in amount of the grocery coming out. So according to the leaky bucket theorem, we get the following expressions and the graph shows the relation among the time and amount of quantity got out. The process is less stochastic so the accuracy is on higher side that will ensure the higher value of Quality of Service (QoS).

$$h = \left(\sqrt{h_o} - \frac{a\sqrt{2g}}{2A} t\right)^2 \quad ...(4) \qquad t_{empty} = \frac{2A\sqrt{h_o}}{a\sqrt{2g}} = \frac{A\sqrt{2h_o}}{a\sqrt{g}} \quad ...(5)$$

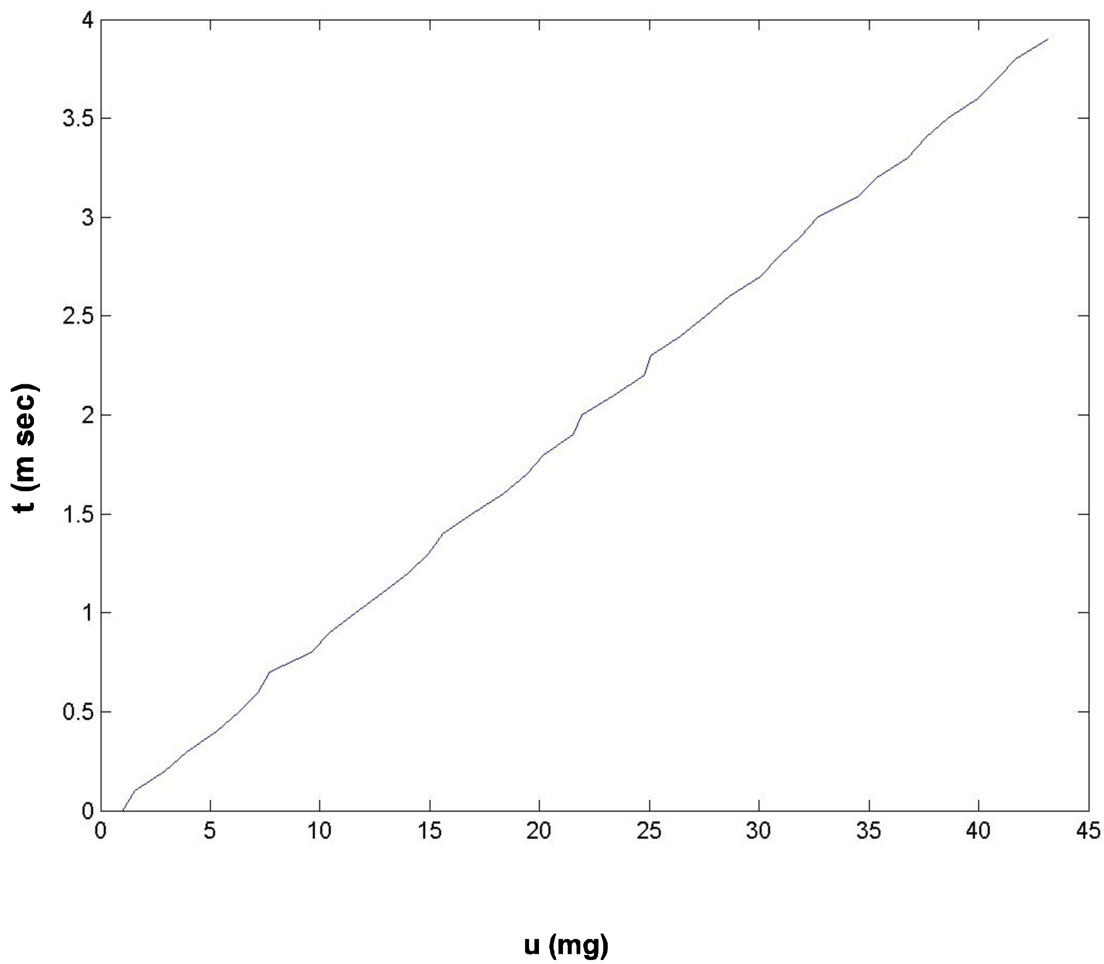


Fig7: The simulation results of grocery item verses time slots. The performance curve shows linearity which means that there will be uniform flow out from chamber after continuous refills.

$$u = \sqrt{2g}\left(\sqrt{h_o} - \frac{a\sqrt{2g}}{2A} t`\right) \quad ... (6)$$

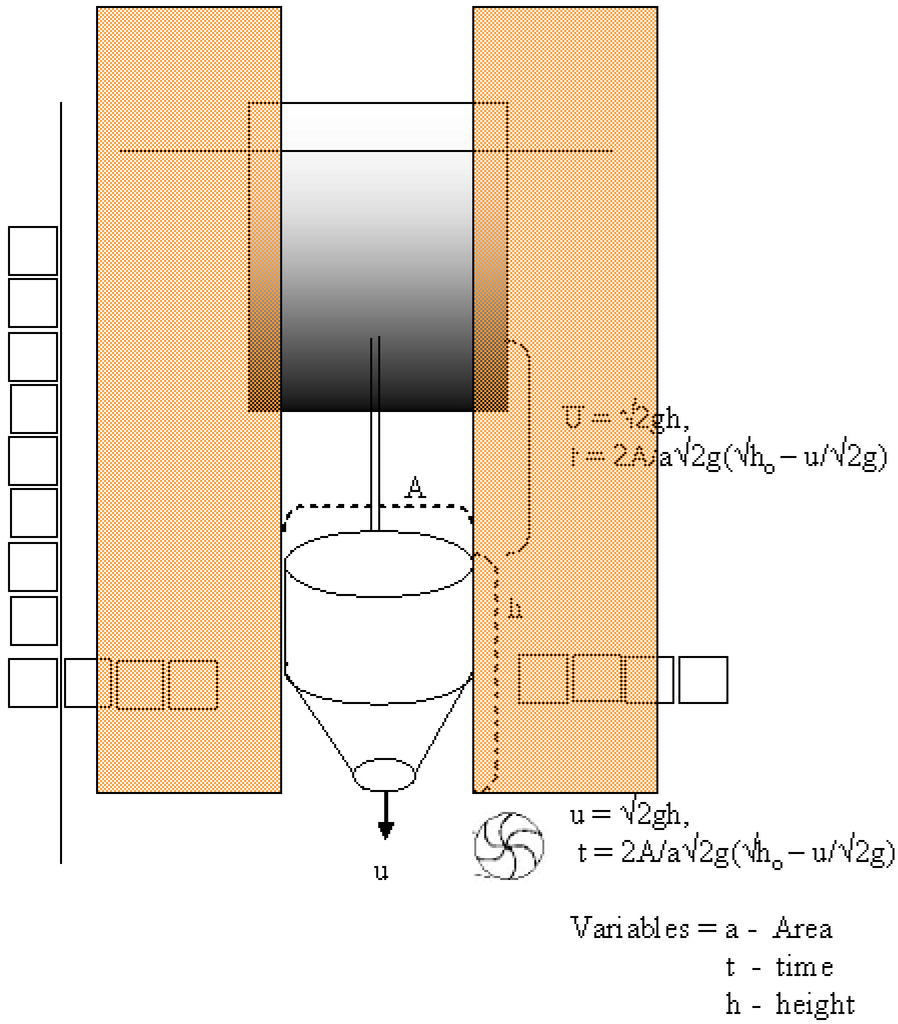


Fig8: Internal Architecture of Grocery chamber

IV. IMPLEMENTATION

We plan to implement it in two languages, Java and C++. The reason is the powerfulness of C++ at system level and platform independence of Java. The designing of modules is done keeping highly mobile environment but since it's a prototype so we keep the quality of software low and leave it for future revisions. We decide that at context aware server the device is registered and server is provided with some option like registering the slot for a task.

The other part of context aware server is the planning and translation of recipe into machine level instructions. We need to follow the sequence shown in fig 9.

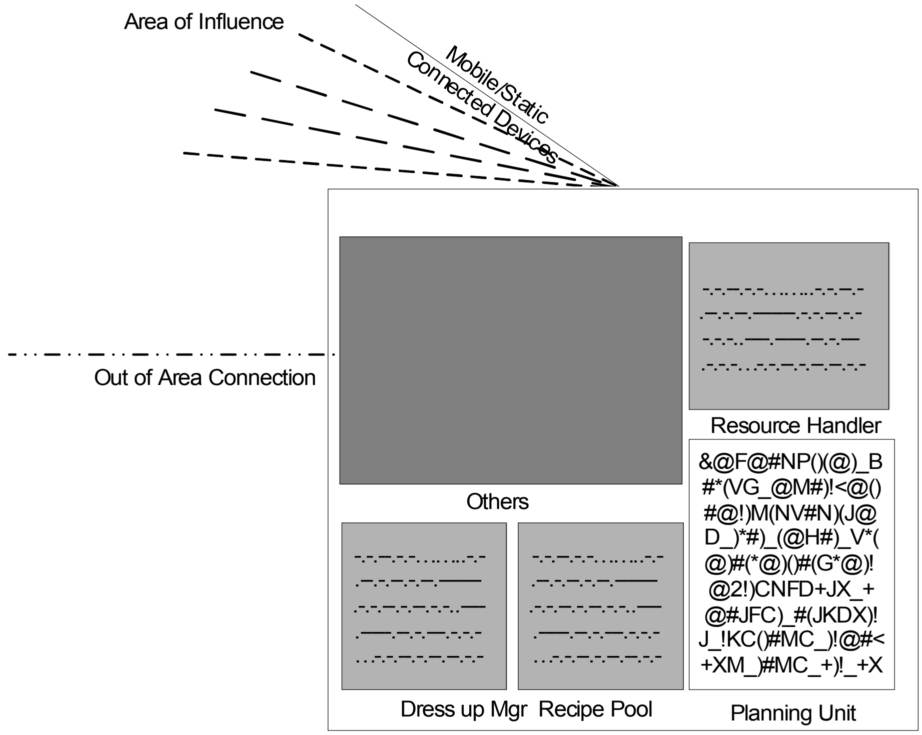


Fig9: Context-aware server Architecture

V. FUTURE WORK

In this paper we have proposed the idea of a ubiquitous cook, its architecture and operation of individual part. The ubiquitous technology emphasizes on the availability of the services everywhere, so the infrastructure for these services should be present everywhere also. The AUto Cook should have some common platform like Open Service Gateway initiative (OSGi). In future we will implement it using OSGi. The next challenge in development of such system is the translation of a recipe keeping the resources in mind, or the working of recipe translator. One more area that, we think, needs future research is the area about the increased level of coordination among the various parts of the system and the devices present around.

REFERENCES

[1] Junaid Ahsen Ali Chowdary, Won-Sik Yoon, Jai-Hoon Kim, and We-Duke Cho, ”U- Kitchen: Application Scenario.” The 2nd IEEE Workshop on Software Technologies for Embedded and Ubiquitous Computing System(WSTFEUS), vol. 1, no. 1, pp. 169-171., May,2004.

[2] McDonald, D.W.; “Ubiquitous recommendation systems,” Computer, Volume: 36 , Issue: 10 , Oct. 2003

[3] Hedberg, S.R., “After desktop computing: a progress report on smart environments research; Intelligent Systems, IEEE [see also IEEE Expert] , Volume: 15 , Issue: 5 , Sept.-Oct. 2000

[4] Yu, M.; Taleb-Bendiab, A.; Reilly, D.; Omar, W., “Ubiquitous service interoperation through polyarchical middleware,” Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on , 13-17 Oct. 2003

[5] Stan Moyer, Dave Marples, and Simon Tsang,, “A Protocol for Wide-Area Secure Network Appliance Communication,” IEEE communication Magazine 2002

[6] Bansal, D.; Bao, J.Q.; Lee, W.C., “QoS-enabled residential gateway architecture.”

Communications Magazine, IEEE , Volume: 41 , Issue: 4 , April 2003 Pages:83 - 89
[7] Ricci, A.; Omicini, A., “Supporting coordination in open computational systems with TuCSon,” Enabling Technologies: Infrastructure for Collaborative Enterprises,2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on , 9-11 June 2003 Pages:365 - 370
[8] Kourouthanassis , P., Koukara, L., Lazaris , C., Thiveos, K.: ”Last-Mile Supply Chain Management: MyGROCER Innovative Business and Technology Framework” in the Proceedings of the 17th International Logistics Congress on Logistics (2001) 264-273
[9] Asthana, A., Cravatts, M., Krzyzanowski, P.: An indoor wireless system for personalized shopping assistance. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, IEEE Computer Society Press (December 1994) 69-74
[10] Oen service Gateway Initiative, www.osgi.org
[11] Takemoto, M.; Sunaga, H.; Tanaka, K.; Matsumura, H.; Shinohara, E., “The ubiquitous service-oriented network (USON)- an approach for a ubiquitous world based on P2P technology,” Peer-to-Peer Computing, 2002. (P2P 2002). Proceedings. Second International Conference on , 5-7 Sept. 2002

Article received: 2005-04-21