An Architectural View of the Entities Required for Execution of Task in Pervasive Space

K.Kalapriya,S.K.Nandy,V.Satish,R.Uma Maheshwari,Deepti Srinivas
CADL,Indian Institute of Science
India
kalapriya,nandy,satish,uma,deepti@cadl.iisc.ernet.in

Abstract

Aimed to provide computation ubiquitously, pervasive computing *is perceived as a means to provide an user the transparency of anywhere, anyplace, anytime computing. Pervasive computing is characterized by execution of task in heterogeneous environments that use invisible and ubiquitously distributed computational devices. It relies on service composition that creates customized services from existing services by process of dynamic discovery, integration and execution of those services. In such an environment, seamlessly providing resource for the execution of the tasks with limited networked capabilities is further complicated by continuously changing context due to mobility of the user. To the best of our knowledge no prior work to provide such a pervasive space has been reported in the literature.In this paper we propose a architectural prespective for pervasive computing by defining entities required for execution of tasks in pervasive space. In particular we address the following issues, viz. entities required for execution of the task, Architecture for providing seamless access to resources in the face of changing context in wireless and wireline infrastructure, dynamic aggregation of resources under heterogeneous environment. We also evaluate the architectural requirements of a pervasive space through a case study*

1. Introduction

Traditional computing has identified computing systems and human system as mutually inseparable. Increase in user mobility, network connectivity, miniaturization, availability of large amount of software services has lead to the new form of computing called "Pervasive Computing", where the intelligence is embedded into the system around the user also known as "Pervasive Space". This coupled with the context aware applications, has lead to separation of human system and computing system into two functionally separate entities, thus allowing the user to view the system in terms of high level task and hiding the low level configurations from the user. Also context-aware applications provide tighter ties between physical and virtual world of computers.

Context can be defined as the environment in which the task executes. Context can be decomposed into three parts viz (a) information about the computing environment, (b) information about the physical environment, and (c) information about the user. It is critical to mobile computing devices at the low level to balance the user needs with respect to what is possible and is cost effective. It depends and changes with an individual's changing intensions and locations. This is further nurtured by the distinct characteristics of mobile computing devices such as unfamiliar environments, collaborations, unpredictability and time-pressures. Due to this,the resource requirements in pervasive space are met by dynamically probing the neighborhood for availability of service thus making the context important.

Pervasive computing encompasses distributed computing, mobile computing along with smart spaces, transparency, invisible computations and localized scalability into one projected space. Though the basic technologies of mobile computing and distributed systems that form pervasive space is available today, pervasive computing seems to be more hypothetical rather than reality. Firstly, the development of robust applications even on divers and constantly changing set of devices and context pose a challenge for defining uniform set of operations that can be captured in a unified model for specifying requirement and services. Secondly, the requirements in pervasive computing span more than one domain of technology. This leads to difficulties in identification and placement of components in the respective spaces, leading to unclear demarcation of functionalities. Thirdly, the mobile devices are small enough to be taken everywhere. This leads to the fact that user may have near constant access to network-based computing services; in turn networked-based services will have constant access to the user. Fourth, the availability and execution of the task is largely dependent on the context and the resources available at the location of the user.

And finally there is a need for a new infrastructure, that shifts the weight of context aware computing onto network accessible middleware. To the best of our knowledge this is the first indepth study on execution of a task in pervasive space making the mobile clients as thin as possible.It facilitates as much of functionality as possible into the network to increase utilization, reliability, sharing and minimize maintenance. It is therefore conceivable that small network-based services be composed to support a larger application. In this paper, we provide system architecture for the pervasive space and carry out detailed analysis of the requirements of pervasive computing. We also propose architecture for resource aggregation. We identify the components necessary to meet the requirements of the pervasive space and also provide uniform abstractions and method for reliable services for common operations. The rest of the paper is organized as follows section 2 gives the importance of context and location in pervasive space. Section 3 outlines the system architecture and design principles. In section 4 we analyze a scenario in pervasive space to validate our architecture. Section 5 proposes a layered architecture for aggregating resources using a peer to peer approach. In section 6 we give the related work and conclude in secti

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

2. Importance of Context and Location in Pervasive Space

In view of improving the Human Computer Interaction(HCI), pervasive space should aim at capturing the human intensions and behavior and appropriately model them. Humans use any situational information to appropriately gauge behavior. This situational information enriches the communication between them. Traditional computing systems do not take advantage of the context of HCI. Context can be defined to be a representation of state of the most important situational parameters of human system. Context can be anything like a personal identification, time, location, task at hand, nearby objects, nearby people, etc. Alternately, context can be a set of premises that can be gathered intentionally or unintentionally and helps reach a set of inferences or some meaningful inferences.

Context has been classified into computing context, user context, physical context and time context [3]. Context aware systems use the context to provide relevant information and/or service to the user depending on the user's task. User tasks largely benefit from this context by taking advantage of the increased relevancy of the information with its environment.

Enhanced mobility through wireless communication has enabled people to access their personal information and public resources "anytime, anywhere". Mobile aware applications should be able to provide the infrastructure in which the systems are able to explore and utilize the dynamically changing context rather than hiding the mobility. With increased mobility and need for seamless access it becomes imperative to device a mechanism to adapt the mobile applications appropriately to support interaction.

A mobile aware application also aims at providing services that are functions of mobile user's physical location. This enables these applications to interact with the environment and to utilize the resources available in the neighborhood. The need for the mobile devices to be thin would mean that their computing resources have to be compromised[1]. But the requirements of computation and data manipulation may far exceed the lightweight mobile entity. To reconcile these contradictory requirements computation and other resources of a mobile device is augmented with resources available in the wired infrastructure.

3. System Architecture and Design Principle

Pervasive computing aims at separating the human system from the computing system. This helps the human system to interact with the computing systems in terms of high-level task without the knowledge of low-level configurations. This has been developed with the intention of making the computing system intelligent enough to capture the requirement of the activities/tasks specified by the user and execute the task by aggregating resources from the nearby locations in a manner that is completely transparent to the user.

Fig. 1 gives an architecture using middleware components for execution of the high-level tasks in pervasive computing. The user agent is embedded in the mobile entity and is responsible for identification of a task. It identifies the resources required for the execution of the task and initiates the resource discovery system to satisfy the requirements.

The computing agent relies on the location

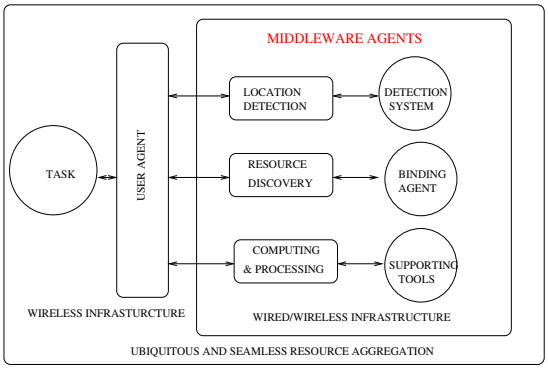

Figure 1. Middle ware architecture for execution of a task in pervasive computing space

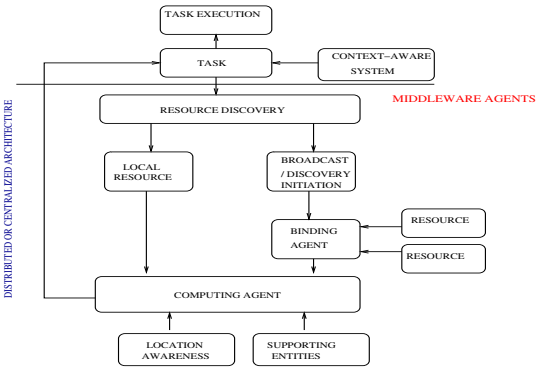

Figure 2. Execution of the task.

detection system for the task entities location co-ordinates and binding agent for resource to location mapping.

The computing agent computes the optimal resource availability and redirects the mobile entity/request to the nearest location that will satisfy the resource request. Fig. 2 describes the sequence of operations for completion of a task. Execution of a task begins with the identification of the resources required for the task. In the absence of the local resources to satisfy the requirements, external resources in the nearby location are used with the aid of binding agent. The computing agent enables the task to completion by coordinating the various requirements of the task to the entities that actually provide the resources.From the foregoing discussion we identify three important components that comprise the pervasive space *viz.* User agent, Middle ware components (which can be either centralized or distributed), and the WLAN and the wired infrastructure. The functionalities of the components are discussed in detail in the next section.

3.1. Middleware Component

Due to the limited computational capabilities of the thin mobile device that can be carried everywhere, pervasive computing aims at transferring computation from conventional mobile device to the computational entities in the nearby locations in a manner that is transparent to the user. This helps the mobile client to be as thin as possible and also to accomplish their tasks even when the entity is not stationary.

a) **Resource Discovery:** Resources requirements of a ta

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

to be identified and discovered dynamically in a hetergeneous and dynamically changing context. Resource can be anything varying from a device, workstation, peripherals etc. Resource discovery involves finding and retrieving those resources relevant to the execution of the task.

b) **Binding Agent:** Binding agent helps identify and retrieve the resources at a particular location.In centralized architecture it is a device that maintains a cache pertaining to the devices/services available in an environment and their respective locations. The resource to location mapping is updated frequently by the resources that advertise their availability. In the case of distributed architecture the binding agent is associated with each Access Point (AP) for local resource mapping (MAC layer resolution). On receiving requests from the client this agent is responsible for retrieving the information about the location of the resource.

c) **Location Detection System:** This system captures the coordinates of the mobile entity.

The co-ordinates helps in finding the nearest optimum location that can satisfy the resource requirements.

d) **Computing Agent:** A device that performs computations in order to determine the optimum resource availability. It takes as input the availability of the resource from the binding agent and location from the location detection system and computes the nearest location that can satisfy the resource requirement.

3.2. System Architecture

In this section we present the user agent, centralized and distributed middleware architecture from the perspective of executing a task for a mobile user. Mobility of the user poses problem for the execution of task in the new location due to reinstanciation of the task in the new location subject to availability of resources. State of the task has to be checkpointed for reinstantiation. Throughout the course of our discussion we concentrate on WLAN as wireless infrastructure, though it can be broadly extented to other wireless infrastructure.

3.3. User agent

The user agent acts as a personal assistant and helps the user in performing user tasks. This agent is responsible for execution of task, task-state management and task state retrieval. Fig. 3 gives the complete diagram of the user agent. Execution of the task begins by identifying the requirements of the task in terms of resources.

Execution of the task is largely dependent on the context, the importance of which is discussed below.

Contextaware system is responsible for retrieving the intention of the user as a task. Tasks can be represented using a personal calendar. An example of a task is sending an email. It takes input from the location detection tool that is part of the middleware architecture to determine the relevance of the context in the location. In the user agent its main The main functionality of the user agent lies in providing the user with information relevant to him/her and his/her task. Example of such information is reminding the user of meeting at the time specified taken as input from his personal calendar. It interacts with the task manager to accomplish the task and its functionality is discussed below.

The task manager is responsible for coordination and execution of a task. For a single task system, it determines the re-

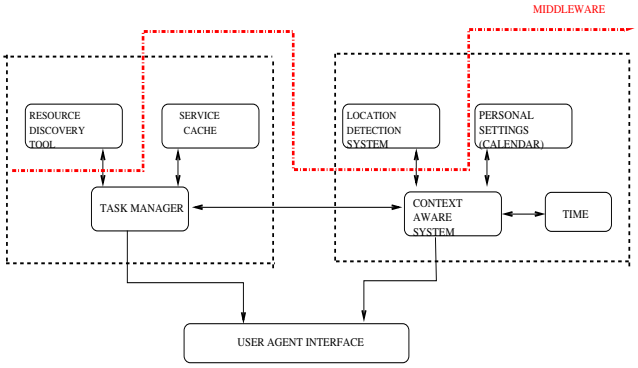

Figure 3. Components of User agent

source required and initiates the resource discovery process. It also maintains the state of the task to be retrieved on moving to a new location if any. On completion of the task, the task manager sends a feedback to the context aware system, which then retires the task from the calendar. To enhance retrieval of task for a mobile user a service cache is provided that maintains the current state of the task. For multiple task system this is responsible for prioritizing and scheduling the most important and relevant task.

The task is executed with the help of middleware agents, which are responsible for providing the necessary supporting tools required for the execution of task. We propose both centralized and distributed architecture for execution of a task while allowing mobility of the user.

3.4. Centralized architecture

Fig. 4 gives a centralized architecture for execution of a task in a WLAN infrastructure. In case of Infrastructure WLAN,AP associates the station that transmits and receives data and connects the users within the network and also serves as points of interconnection between WLAN and a wireline network. A mobile device or a user holding a wireless enabled device is always in association with a wireless AP. We use the term mobile entity to represent either a mobile user (holding a wireless enabled device) or mobile device.

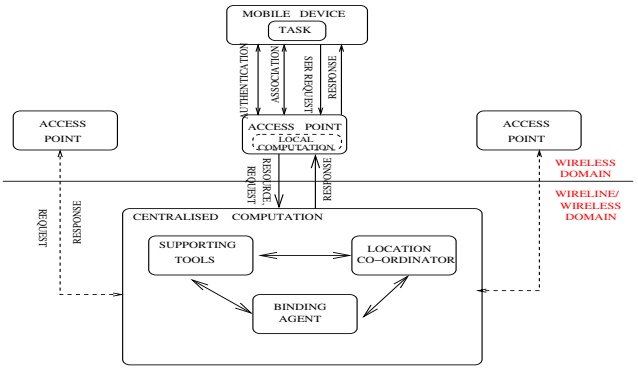

Figure 4. Centralized architecture for execution of a task using middle ware components

The association of a Mobile Entity (ME) with the access point starts with the authentication process where the client pro-

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

its identity to the access points. After completion of the authentication procedure the ME initiates association procedure. On sucessful association, user agent on the mobile entity identifies the task and detects the resources required and communicates it to the AP. The AP forwards the request to the binding agent to get the location of the resource provider. The supporting entities required for execution of the task are available centrally to all the AP entities. This will include a location detection system, bagent, and other supporting tools required for a specific scenario. These supporting entities are available on the wired backbone network. The mobile entity will be in constant access to the wired network through the access points.

Binding agent helps in mapping the resource to the location and computing agent looks for the best entity that can provide the resource after which the user agent (corresponds to the Service Providing Entity) and gets the required resource and proceeds to complete of task.

3.5. Distributed architecture

The distributed architecture differs significantly from the centralized architecture. In the distributed architecture resource discovery is dynamic and no central entity is required to map the resource to location. The location of the mobile user and the associated device is computed locally. In the absence of the local resource to execute the task the request is broadcast to other AP entities. Resources available over the wireline network are accessed directly through the wired backbone connecting the entities. On receipt of the request the AP forwards the request to computing agent associated with the AP, which is associated with every AP. This agent now checks the required resource within its own Basic Service Set (BSS). Availability of resource within its own BSS requires only the identity of the resource to be sent back to the requested AP. If the resources are inadequate within the BSS of the associated AP then the PBA broadcasts the resource request to its neighboring APs. In case of multiple responses, decision is based on the nearest location and is computed by the user agent of the AP that requested the resource. The client contacts the SPA and gets the required resource. Fig. **??** shows the decentralized architecture for execution of task in distributed architecture.

3.6. Brief comparison between Central and Distributed Architectures

Merits of Central Architecture:

1. Eliminates the AP-AP communication during resource detection phase, thus reducing the traffic and avoiding congestion in a limited bandwidth environment.

2. Provides less number of net message exchanges over wired/wireless communication links.

Demerits of Central Architecture:

This Central system is prone to some of the drawbacks inherently present in centralized systems that include

1. Resources have to be known in advance and require bookkeeping costs.

2. Single point of failure- Any malfunction in the infrastructure results in the failure of the entire system.

Merits of Distributed Architecture:

1. Requires relatively less book keeping information.

2. Resource discovery can be invoked even in diverse environment and no knowledge about availability of resource is known *a priori*

3. System is robust- this is because system is more dynamic and eliminates assumption about the availability of resources.

4. Preferred in dynamically changing context, since it reduces the bookkeeping information about the resources by eliminating the requirement of central binding agent.

Demerits of Distributed system:

1. Involves AP-AP communication during resource detection phase thus making use of valuable wireless bandwidth.

2. The net number of message exchanges over wired or wireless communication links is relatively more due to broadcast of request and responses from multiple entities.

Hybrid architecture.

Hybrid architecture combines the advantage of both distributed and central system. Resources that are rendered by the moving device or dynamically changing can by dynamically discovered using distributed architecture protocol (broadcast and response), whereas resources known aprior can use the centralized architecture.

4. Case Analysis

4.1. The scenario

The project Aura [1] aims at improving computing efficiency, specifically in wireless environments, while minimizing "distractions such as poor performance and failures" by implementing a "personal information aura" that spans wearable, handheld, desktop and infrastructure computers. It aims at enhancing user efficiency related to task execution with greater system reliability and fewer distractions.

We study the case scenario given in [1] under the system Aura. The case scenario is given below.

We use this scenario to highlight the salient features of our proposed architecture for Pervasive Computing.

Jane is at Gate 23 in the Pittsburgh airport, waiting for her connecting flight. She has edited many large documents, and would like to user her wireless connection to e-mail them. Unfortunately, bandwidth is miserable because many passengers at Gate 22 and 23 are surfing the web.The system observes that at the current bandwidth Jane wont be able to finish sending her documents before her flight departs.Consulting the airport's network weather service and flight schedule service, the system discovers that wireless bandwidth is excellent at Gate 15, and that there is no departing or arriving flights at nearby gates for half an hour.A dialog box pops up on Jane's screen suggesting that she go to Gate 15, which is only hree minutes away. It also asks her to prioritize her email, so that the most critical messagesare transmitted first. Jane accepts the advice and walks to Gate 15. She watches CNN on the TV when she is informed that its

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

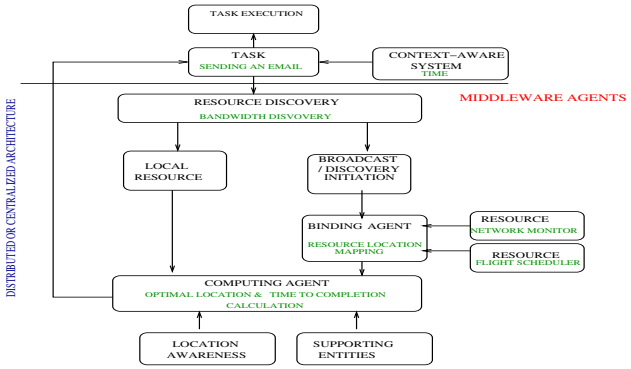

Figure 5. Interaction between entities for execution of task.

to being done with her messages,and that she can start walking back. The last message is transmitted during her walk, and she is back at Gate 23 in time for her boarding call.

4.2. Analysis of components and requirements for the execution of task

Fig. 5 details the different components and their interaction with user agent for completion of the task.

Fig. 6 gives the various entities in wired and wireless environment and interaction among the components that help in execution of the task of sending the e-mail. Step 1 indicates authentication and association of Jane's handheld device to the AP at Gate 23. Context aware system provides the information of the current time and Jane's schedule to the Task Manager, which identifies the task of sending an email, and initiates the resource discovery. User agent computes the location co-ordinates of the handheld device. The location co-ordinates help to compute the nearest optimum location to satisfy the requirements of task (network bandwidth resource) available. The service cache is constantly updated with the task status.

Every AP is provided with a computing agent to process the request of the entities associated with it. On identifying that the local bandwidth is inadequate for transferring email before Jane's departure, the computing agent queries the location of the network weather service and flight scheduler of the airport and send this information to the binding agent. The network weather service is queried for the network traffic at the different gates and the flight scheduler obtains the flight arrival and departure at different gates. The computing agent calculates that gate 15 is the nearest possible gate that would satisfy the bandwidth requirements for sending the email before the arrival of Jane's flight. Jane is suggested to move to gate 15, during which the service cache is used to checkpoint the task. With the help of location detection system (middleware component) the user agent identifies the arrival of Jane on gate 15 and initiates the authentication and association with the new AP. The computing system tracks the task to completion, which informs the user agent of the completion of task. The service cache is used to checkpoint the status of the task to be retrieved across associations and re-associations between various AP's as Jane moves from gate 15 to gate 23 (during Jane's return).

Finally we can view the task as a real time task with soft deadline determined by the arrival of flight. The execution of the task requires resources (network bandwidth) to be completed before the deadline. The inadequate local resources initiate the resource discovery. This requires that aggregation of resources under dynamically changing and heterogeneous environment be addressed. In the next section we provide layered architecture for aggregating resources in pervasive space.

5. Layered Architecture for Resource Aggregation: a peer-to -peer approach

We present a layered architecture for resource aggregation. In pervasive computing resources requirement is largely dependent on the scenario. We provide a generic architecture for aggregating resource in pervasive space using a peer-to-peer approach. Resource can be anything that helps us accomplish the task. Resources are requested by the task and provided by the end-hosts/service providers. Either the task might need more than one resource or the computation context may change which pose a need for aggregating resources. Computation context is anything like network connectivity, network bandwidth and nearby resources such as printers, peripherals, displays and workstation themselves. Furthermore, constant change of location due to the mobility of the user might lead to change in availability of resources in the new location. The entities that serve to provide resources are considered peers.

The resource requirements can be broadly categorized into computational resource requirements and non-computational resource requirements. Example of computational resources is CPU cycle requirement for tasks such as online encoding/decoding, execution of compute bound tasks like simulations. Examples of non-computational resource requirement are printers, peripherals, editing tools that are not available local to the device. Furthermore location of the resource also plays a critical role in satisfying the request. Fig. 7 gives the logical categorization of the resources based on the location and computational requirements.

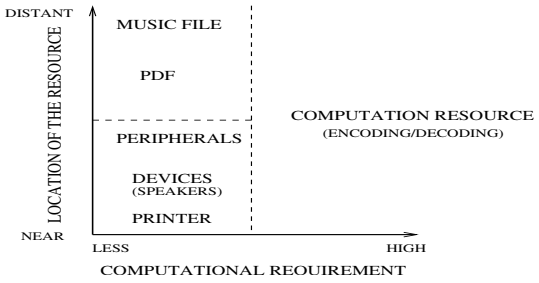

Figure 7. Resource categorization

Fig. 8 presents a layered architecture that groups logically the functionalities required for management and aggregation of resources in pervasive space using a peer to peer approach. The whole system can be grouped into three spaces user space that initiate the task, the pervasive space that coordinates the mapping of tasks to services and the end systems that provide the services. Task specification layer is responsible for representation of the task. Tools are used by the tasks while applications are originators and consumers of tasks. Components of the Resource Aggregation layer are mainly part of the middleware and are responsible for aggregating resources required for the execution of the task. This requires a scheduler to schedule the incoming r

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

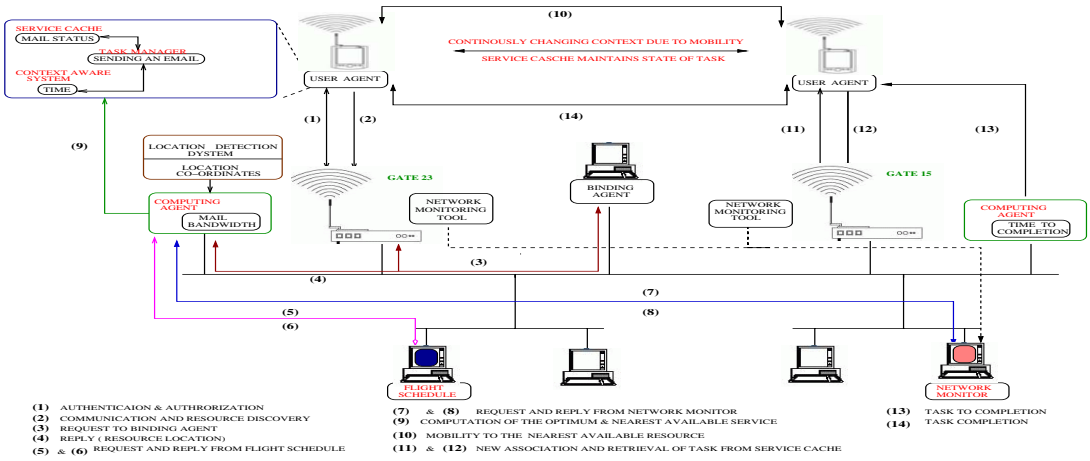

Figure 6. Hybrid architecture for execution of the task of sending the email

from various tasks. Meta Data information on the end-systems represents the footprint of the services available. Resource Aggregation is closely coupled with the discovery of the resource, transparent routing of the request and location of the resource. Communication between the entities is defined by the protocol specified in the communication layer. Fig. 9 details the placement of the functionalities in different entities of the pervasive space.

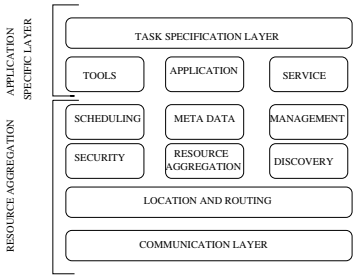

Figure 8. Layering of the functionalities for aggregation of resources.

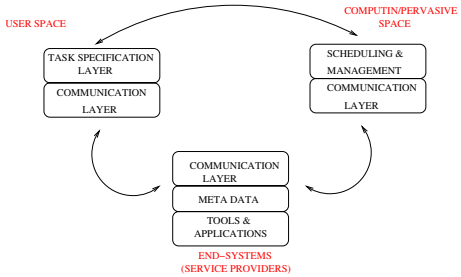

Figure 9. Interaction between the layers of entities and their placement in computing space

We provide an insight into some of the issues that have to be addressed for aggregating resources for pervasive computing using a peer-to-peer approach.

1. Response time guarantees have to be provided for the tasks. Response time is deterministic. The system exhibits soft time guarantees. The response time is largely dependent on the service model.

2. Mobility of the user poses further problems. Resource discovery has to be initiated whenever the computing context changes due to mobility of the user or due to the transient nature of the peers. Resource Aggregation is also responsible for aggregating resources seamlessly in an ad-hoc environment.

3. Span defines the distance to which a resource discovery is propagated. Span depends on the resource and tolerance it requires

4. Placement and management of scheduling functionalities and Resource Aggregator to provide interoperability between wired and wireless domain and to provide seamless access to resources maintaining transparency to the user.

5. Reliability, Response time guarantees and Service guarantees are some of the issues that must be addressed for providing resources seamlessly to the tasks executing in the pervasive space.

6. Related Work

Satyanarayan in [1] proposes the challenges in the field of pervasive computing. He identifies the various components that comprise the pervasive space and points out why pervasive computing appears to be a fiction today rather than reality. He points out that pro-activity and self- tuning as some of the main key capabilities in today's traditional computing systems.

Context aware system has been extensively studied in [11], [12], [5]. Context has been defined and context being a constraint for execution of task is clearly brought out in [12]. Further context has been classified as computing context, user context, physical context and time context in [3]. It gives deeper insight into importance of each of these contexts.

Next phase of information technology has been viewed a convergence of control, communication and computation in [16]. It has been argued that software architecture will play a major role in proliferation of this convergence. They have given various architecture challenges involved in integration of large n

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

of sensors, actuators and computational units, all interconnected wirelessly or over wires, to interact with the physical environment.

A detailed comparison between proactive computing and automatic computing has been discussed in [8]. The paper identifies significant overlap between autonomic and proactive systems and both autonomic and proactive systems are necessary to provide us with tools to advance the design of computing systems in a wide range of new fields. It has been proposed that expectations of a mobile users requires fundamental changes be made in way people perceive the role of devices, applications and environment [9].

Task driven computing is explained in [14]. Tasks are identified as intentions of the user devoid of low-level configurations. A task execution model has been devised by composing tasks as application-independent primitives that are bound to actual services in the environment. System architecture for pervasive computing is given in [4]. A lightweight architecture that can be implemented across heterogeneous devices has been proposed. They attribute the lack of infrastructure in existing technologies to support pervasive computing and hence it requires additional services like task-state management and context-sensitivity. It has been shown in [2] that deployment of distributed system helps in increasing fault tolerance. We propose architecture and identify the components required for execution of high-level task in pervasive space.

In [6] a resource discovery mechanism has been proposed for dynamic and mobile networks where the request for resources may consist of a description of the specific attributes required in the device or from the service required. Applications use language to describe what they are looking for and not where they are looking for. This Naming System uses a simple language based on values for its names and attributes. We propose a layered architecture for aggregating resources in highly dynamic and heterogeneous environment.

In our work we present a detail analysis of the execution of task in pervasive space. We identify the system components

required for completion of the task in highly dynamic and hetergeneous environment. The system components has been designed and located in the wireline network with a view to keep the mobile devices thin. We have also concentrated on making the communication in the wireless domain as light as possible by restricting the number of messages exchanged between the AP and mobile client to a minimum. Using the layered architecture resources from the pervasive space are seamlessly provided to the applications.

7. Conclusions

We propose a middleware architecture that shifts the weight of context-aware system to network access for execution of a task in pervasive space. We considered both centralized and distributed architecture and compare these architectures. Key principles in design of such systems involve maintaining transparency, seamless access to resources in changing context. We strengthen our principles through a case analysis and identify three entities viz., user agent, middleware components (location detection, computing and binding agents) and wireline/wireless infrastructure as key components for execution task in the pervasive space. In order to allow the mobile clients to be thin we keep the communication and computational overhead as light as possible on the mobile entities by locating a majority of the middleware agents on closely accessible wireline network. We also propose layering of functionalities for aggregating resources for execution of tasks in highly dynamic environments. Finally we categorize the resources based on their computational requirement and elucidate the issues that must be addressed.

References

[1] M. Satyanarayanan, "Pervasive computing: Vision and challenges," IEEE Personal Communications, vol. 8, pp.

10–17, Aug. 2001.

[2] D. Chakraborty, F. Perich, A. Joshi, T. Finin, and Y. Yesha. A Reactive Service Composition Architecture for Pervasive Computing Environments. In Proceedings of the 7th Personal Wireless Communications Conference (PCW'2002), Singapore, 2002.

[3] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381,

Computer Science Department, Dartmouth College, Hanover, New Hampshire, November 2000

[4] R. Grimm, T. Anderson, B. Bershad, and D. Wetherall. A system architecture for pervasive computing. In Proceedings of the 9th ACM SIGOPS European Workshop, pages 177–182, 2000

[5] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modelling Context Information in Pervasive computing systems. Pervasive 2002, LNCS 2414, pp. 167-180,2002.

[6] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. The design and implementation of an intentional naming system. In Proc. 17th SOSP, pages 186–201, December 1999.

[7] Narayanan, D. and Satyanarayan, M. Predictive resource management for wearable computing. Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, May 2003.

[8] Want, Pering and Tennenhouse,Comparing Automatic and Proactive Computing, IBM Systems Journal ,vol 42,

No.1 , 2003

[9] Guruduth Banavar, James Beck, Eugene Gluzberg, Jonathan Munson, Jeremy Sussman, Deborra Zukowski,"Challenges: An Application Model for Pervasive Computing",in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, 2000, pp.266-274.

[10] Christopher Lueg, "Representations in Pervasive Computing",in Proceedings of the INaugural Asia

Pacific Forum on Pervasive Computing, Nov 2002.

[11] Anind K. Dey and Gregory D. Abowd, "Towards a Better Understanding of Context and Context-Awareness",in the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000).

[12] Saul Greenberg, "Context as a Dynamic Construct", Human-Computer Interaction, Volume 16 (2-4) 2001, p257-268.

[13] David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste, "Project Aura: Toward Distraction-Free Pervasive Computing", in IEE Pervasive Computing, special issue on "Integrated Pervasive Computing Environments", Volume 21, Number 2, April-June, 2002, pp. 22-31.

[14] Wang, Z., and Garlan, D. Task-Driven Computing. Carnegie Mellon University School of Computer Science Technical Report CMU-CS-00-154, May, 2000.

[15] G. D. Abowd and E. D. Mynatt, "Charting Past, Present, and Future Research in Ubiquitous Computing," presented at ACM Transactions on Computer-Human Interaction, 2000.

[16] Scott Graham and P. R. Kumar, "The Convergence of Control, Communication, and Computation." To appear in Proceedings of PWC 2003: Personal Wireless Communication, September 23-25, 2003, Venice, Italy.