EkSarva PIA

Kaladhar Rajoori

Problem Report Submitted to the

College of Engineering and Mineral Resources

at West Virginia University

in partial fulfillment of the requirements

for the degree of

Master of Science in

Computer Science

Dr. Y.V. Reddy, Ph.D., Chair

Dr. James D. Mooney, Ph.D.

Dr. Sumitra Reddy, Ph.D.

Department of Computer Science

Morgantown, West Virginia

2006

ABSTRACT

EkSarva PIA

Kaladhar Rajoori

Traditional collaboration systems bundled with a set of pre-defined tools with pre-specified static workflow may not fulfill new requirements imposed by today's heterogeneous and ever-changing collaboration contexts environments. These systems involve dealing with variety of information that users need to be aware of, complexity of usage and moreover are not adaptable automatically to the environment. Contextual information of what and where the user task is, what the user knows, and what the system capabilities are, can greatly simplify the user scenarios. So this leads us to the designing of more generic sophisticated collaborative systems. **EkSarva PIA** is a sophisticated Personal Information Agent in a generic collaborative environment which has context awareness, workflow and rules. Our main goal is to develop the EkSarva PIA, EkSarva test model and build a proto-type upon which future work can be done. We describe the architecture of our approach, key design and implementation issues, illustrate its feasibility and evaluate its effectiveness. A realistic collaborative scenario which involves users working in different organization projects by managing all the projects with **PIA** (Personal Information Agent) embedded in server and a client GUI is all that is being developed.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Dr. Yenumula V. Reddy for his constant support and guidance for my work and on guidance prior to this problem report. I would like to extend my thanks to Dr. James D. Mooney and Dr. Sumitra Reddy and for serving on my committee and also for their encouragement and support.

I would also like to thank Dr. Vijayanand Bharadwaj for his continuous, unconditional, priceless full-time help and support during this project.

I am extremely thankful to my project mates for their cooperation in this project and would like to wish them good luck for their future endeavors.

Finally, it will not be out of place to mention thanks and regards to my beloved parents and my friends for their encouragement and enthusiastic support rendered to me at various stages for the development of our professional career.

Table of Contents

Disclaimer: Some of the work shown here is reproduced from Dr.Reddy's research work with his consent. The first three chapters (theoretical background) are the same for all three of us in the group as we all deal with the same theory.

Chapter 1

Introduction:

The construction of computer systems which are collaborative problem-solving partners, intelligent is an important goal for the science applications. With the proliferation of networked computing, organizations are becoming increasingly geographically widespread leading to the distribution of employee skills and business intelligence. From the research perspective, the development of theories and mechanisms to enable building collaborative systems presents research challenges across various fields like computer science. From the applications perspective, the capability to collaborate with users and other systems is essential if large-scale information systems are to assist users in finding the information they need and solving the problems they have. In this address, it is argued that collaboration must be designed into systems from the start; it cannot be patched on.

In this networked world of information, effective collaboration among knowledge workers is imperative. Collaboration in its most fundamental form entails information generation called artifacts, and sharing it with the group information. The sharing of the artifacts often results in generating of new artifacts. With the boom of digital computers, researchers are resorting to digital computing to boost the effectiveness and efficiency of collaboration. These efforts have led to many research areas including Computer-Supported Cooperative Work (CSCW) is one of the most well-known areas which discusses how computers might be used more effectively to support collaborative work among people. Since then, CSCW, as an emerging interdisciplinary field and attracted a great amount of attention from researchers with a wide range of interests.

Over the last decade, the issue of collaboration has received increasing attention from practitioners in academia, education and industry. This applies to researchers from various fields and in particular from the Computer Supported Cooperative Work (CSCW), and the World Wide Web (WWW) communities. Collaboration has begun to receive considerable attention in CSCW and groupware research. Awareness is nothing but the information that the users exchange with their environment in order to accomplish the various tasks effectively. Exchange of awareness information by computer based mechanisms is of primary importance in today's technology world. Awareness in true sense also refers to the ability to maintain some knowledge about the various situations, activities and maintaining this type of knowledge should not require much effort and awareness has to be happen natural.

Also maintaining the awareness has proven difficult in real-time and distributed systems where information resources are poor and interaction mechanisms are alien. As a result, working together with a groupware system often seems inefficient and clumsy compared with co-ordinate work. It is becoming more and more apparent that being able to stay aware of others work plays an important role in the fluidity and naturalness of collaboration, and supporting awareness of others is looked on as one way of reducing the characteristic awkwardness of remote collaboration. Awareness is a design concept that holds promise for significantly improving the usability of real-time distributed groupware systems.

1.1 Importance of Collaboration in CSCW

Collaboration can be termed as e-mail, WWW, Internet chat etc. Different definitions for collaboration arise because there are so many technologies and its definition today is broad. Collaboration is the integration of many different technologies into a single application or

environment to accommodate information sharing and information management. Collaboration can occur in two different ways, one is real time collaboration where one can work with others at the same moment, taking turns to communicate the ideas. The second approach is asynchronous collaboration, in which one need not be present to participate. E-mail, chat, public databases, the Internet, and intranets are all forms of this kind of communication.

A collaborative technology provides major and key benefits to business systems. Some of them are central repository for storage of database, extensive and secure communication, and ability to extend existing technologies with new functionality.

As a relatively new field, the views on CSCW are diverse. In general, these views can be classified into two divisions: technology-centric and work-centric. Technology-centric CSCW approach places an emphasis on devising technologies to support and improve collaboration. The collaborative tools devised under this view are often collectively called groupware. The work-centric direction places an emphasis on understanding work processes to design better computer systems to support collaboration. Although both directions aim to improve the productivity and effectiveness of collaborative work, there is still a considerable gap between them due to different views on the subject.

Network based collaboration among groups and individuals; in business, general and social contexts is a fundamental prerequisite for building a Digital Society. Despite many advances in Computer Supported Collaborative Work (CSCW), network based collaboration has not yet become widespread. The main reason for this situation is CSCW, in practice continues to be tool-centric rather than workflow centric.

1.1.1 Examples of Collaborative Activities

Collaboration is central to intelligent behavior is clear from the ways in which it pervades daily activity. They range from the pre-planned, practiced and well coordinated collaborations of sports teams, dancers to the spontaneous collaborations of people who discover they have a problem solved together. Scientific collaborations occur on large and small scales and across the various sciences. To illustrate some of the features of collaborative activity, let us examine in more detail a small-scale collaboration: health care. The health-care example involves three people working together toward common goals;

Example: Health Care System

Patient health related problems:

heart attack with congestive heart failure

pneumonia and emphysema

Team:

Cardiologist(Heart Specialist)

infectious disease specialist team and pulmonary specialist

Collaborations:

diuretics: cardiologist & pulmonary specialist

antibiotics: pulmonary specialist & infectious disease specialist

In the health care scenario explained in the above figure, a patient arrives at the hospital with three problems affecting his heart and lungs. For this three specialists are needed for curing the patient. But this is not a one doctor or one disease situation. Treating the patient will require

teamwork of collaboration. For example, the cardiologist and pulmonary specialist must agree on a plan of action for reducing the water in the patient's lungs. And the infectious disease and pulmonary specialists must plan together the amount of antibiotic to prescribe.

No single doctor is the manager, telling the others who does what; there's no master-servant relationship here. The doctors need to come to a consensus about what to do and who's going to do it. Each of them has only some of the information needed to devise a plan for action; they will have to plan jointly. In doing so, each doctor will make presumptions about the others' beliefs and capabilities and must decide when to check out these assumptions. Each doctor will be counting on the others to contribute what they know to the solution and to do their share.

1.2 Context Awareness in CSCW:

Context refers to the physical and social situation in which computational devices are embedded. One goal of context-aware computing is to acquire and utilize information about the context of a device to provide services that are appropriate to the particular people, place, time, events, etc. Also Context awareness used for devices that have information about the circumstances under which they operate and can react accordingly. Context aware devices may also try to make assumptions about the user's current situation. For example, a cell phone will always vibrate and never beep in a concert, if the system can know the location of the cell phone and the concert schedule.

We can view context in two aspects: business logic context or collaboration logic context; and computing environment context in collaboration. Collaboration logic context involves the business scenario context through the entire cooperative work. It is often driven by workflow

rules and moves from one phase to another; exceptions should also be solved through pre-defined rules or other intelligent mechanisms.

Consider an application like Cyber guide can take advantage of user mobility by adapting behavior based on knowledge of the user's current or present location. This location can refer to the position and orientation of a single person, many people, or even of the application itself. Location is a simple example of context i.e. information about the environment associated with an application. Context-aware computing involves application development that allows for collection of context and dynamic program behavior dictated by knowledge of this environment. In addition to dealing with raw context information such as position, a context-aware application is able to assign meaning to the events in the outside world and use that information effectively. Context-awareness is not unique to ubiquitous computing. For example, explicit user models used to predict the level of user expertise or mechanisms to provide context-sensitive help are good examples used in many desktop systems. Moreover, context-awareness is a critical feature for a ubiquitous computing system because important context changes are more frequent. In a ubiquitous computing environment it is likely that the physical interfaces will not be ``owned" by any one user. When a user owns the interface --as is usually the case with personal digital assistant or a laptop computer-- over time this interface can be personalized to the user. Context can be useful in these situations, as has been demonstrated by location-aware computing applications. Context-awareness will allow for this rapid personalization of computing services.

During recent years, ubiquitous computing and pervasive network connectivity have given rise to expectations of building a digital society, where remote participants with heterogeneous and dynamic connection and process capabilities can join seamlessly to accomplish a common task. This imposes a new requirement on today's collaboration framework: context-aware capabilities

of adapting to heterogeneous computing requirements. Computing environment context-awareness enables a collaboration process to adapt to users proactively without distracting users or by minimizing the distraction. The essence of the context-awareness is invisibility; the changes of physical locations, process speed or other technical barriers should be transparent or invisible to collaboration participants. The collaborators should focus on the workflows embodying the collaboration process and moving on to achieve the final target instead of spending a significant amount of time and efforts in reconfiguring tools and adapting to new computing environments.

1.3 Workflow Centric concept in CSCW:

In a society of knowledge employees/workers, collaboration consists of generating information and sharing it with others in the same or different community. This sharing of information will result in further generation of new information which will invoke actions upon them. The underlying paradigm of this collaboration can be simply stated as "information sharing, when acted upon, results in a state-change of the project which triggers further information exchange-until the project goals are achieved". This collaboration process is often referred to as "Workflow", which can drive a cooperative information system through different phases of collaboration. In the design of this EkSarva collaboration framework, we focus on the workflows driving the collaboration towards a common task.

1.4 Problem Statement:

Current existing technology-centric approach or groupware and its collaboration tools mainly focus on providing information sharing. They typically bundle the numerous functionalities found in applications such as email, chat, document editors, calendars, personal information

management tools and process management together and provide a single interface for their usage. This narrowly-scoped inefficient technology-centric approach only increases the usage complexity and is not adequate to capture the semantics of the collaboration operation. The work-centric approach has placed more emphasis on understanding the organizational structure, associated roles and responsibilities. The emphasis is largely on office automation and other mostly routine processes as may be supported with activities involving email, forms based data access, chat, personal information tools, documents and imaging, and sometimes a broader range of activities supported by information systems. The more fundamental issues that now make attention in today's collaboration include the ways to:

Support processes that are more flexible, adaptive, and hence dynamically modifiable, such as the computer supported organizational processes can support more of the intelligent and intuitive support that humans orchestrating and participating in an organizational process can provide flexible controls.

A high degree process reusability and using a repository for consistency of process ontology, resource ontology including user-centered ontologies such as organizational roles, authorizations, and other context required to create the appropriate views or models of information in a particular task, work process definitions, and detailed activity specifications;

Collaboration, providing a variety of tools (voice, video, audio, chat, white boarding) supporting human interaction, both at the individual activity level and the work process.

The main goal here is not only to take away from humans the more routine activities and automate or support workflow processes that require coordination, but to support them in processes and activities that involve coordination, collaboration, as well as decision making. In

other words, we propose that a more comprehensive approach to supporting organizational processes should include integrated support for management, coordination and collaboration of information.

To come up with a better solution that overcomes the failure of traditional collaboration techniques or tools in meeting the requirements of contemporary collaboration and that provides a generic workflow patterns rather than pre specified a generic collaboration framework, **EkSarva**, where the workflow is embedded in the enactment system and context-awareness is automatically maintained to support diverse and dynamic collaboration patterns. To enable adaptability, collaboration first must be aware of the changes of contexts: both collaboration contexts and computing environment contexts, and afterwards be able to react to these changes.

EkSarva TestBed is a generic collaborative system with all the concepts of EkSarva. It is a test bed for EkSarva. The main goal of **EkSarva TestBed** is to put workflow at the center and hide the details and complexities of tools through intuitive interfaces and familiar paradigms. By combining advances from the areas of intelligent agents, workflow engines, context, coordination theory, markup languages, and other related technologies with everyday tools like e-mail, chat, personal information tools and word processors. This framework will be easily deployable in a wide variety of business and other processes.

Our Report is organized as follows. We first describe the key design considerations of EkSarva TestBed. Then related work is reviewed. Next, we describe in detail the EkSarva PIA (Personal Information Agent) and EkSarva TestBed framework design and the formalisms it provides to model collaboration. Following this, we use an example to illustrate the usage of this framework and discuss our ongoing implementation of a prototype system.

Chapter 2

Background and Related work

2.1 Definition of EkSarva

EkSarva is a context based workflow centric collaborative system which supports PPP/SST model.

Ubiquitous computing and universal network have given rise to expectations of building a more sophisticated Digital Society, where many aspects of business and social activities may be accomplished by remote participants. Context-awareness, workflow-centric and Intelligence collaboration is an essential prerequisite to fulfill that expectation. EkSarva is a framework, which enables realization of such collaborations in diverse domains. This is accomplished by identifying the key concepts and their inter-relationships (dubbed the PPP/SST paradigm), which form the foundation of this framework. The key concepts include: (a) Project, (b) Person, (c) Place, (d) Signal, (e) Situation, and (f) Transcript. This paper describes how these concepts collectively can give the design of EkSarva framework place emphasizes on the following two considerations: workflow centric, context awareness and adaptability.

2.2 Conceptualized Collaboration Framework (EkSarva)

This framework recognizes any collaborative organization or enterprise may be represented in terms of the following basic concepts:

Project (P) that represents the overall project and goals of the projects.

Two or more persons (P) who participate in some collaborative activity of the same and different project

One or more places (P) where *an* activity takes place

Number of situations (S) that result from the above activities

Number of different signals (S) that trigger different situations

Transcript (T) the total results from collaborative activities at every situation.

This framework PPP/SST (referred to as EkSarva - a combination of two Sanskrit words: Ekatra - one place, and Samatra – all places) has the necessary generality to define compactly most business processes, which can be implemented in a networked environment, By using a markup languages such as XML for the Transcript as well as for the behavioral profile, role and other knowledge representation. As the transcript could be analyzed by a machine, it could be used to represent process intelligence. This Smart Transcript can generate the necessary signals to advance the process from one situation to another. EkSarva TestBed will also provide an interface to create a representation of a business in the above framework, much in the spirit of Expert System Shells that enabled introduction of intelligent systems into a variety of domains.

There are several existing commercial collaboration products such as Microsoft Exchange, Macromedia Breeze, NetMeeting, Haystack and IBM Lotus that employ a collaboration framework in certain forms as well. But these are simple tools that provide only a tool centric view. Although sharing many common goals such as promoting productivity, EkSarva framework is significantly different from the above ones, in that it brings context-awareness and workflow-centric to the forefront. The EkSarva framework has well specified project goals, workflow behavior description and a resource model. By using intelligence into the framework, EkSarva framework facilitates the dynamic alteration of project workflow and collaborators' behaviors.

2.3. Collaboration Network

The framework conceptualizes the following entities

2.3.1 Person (P)

Person is an individual or a group member involved in a collaboration process. Capturing the behavior of a person under certain collaborative circumstances or instances enables us to model the entity's roles, privileges and designated activities. By analyzing the Persons involved in collaboration we could conveniently incorporate a software agent as a person into our collaboration.

2.3.2 Project (P)

A project is a specification of the collaboration goal and the terminal point of the collaboration. The goal is specified as a series of tasks where each task is a Situation. A task is complete when a situation or a composition of situations is reached. Hence we could view the collaboration project progressing, by means of signaling, through various situations until the final goal is achieved.

2.3.3 Place (P)

A place is a concept of a virtual meeting place like a meeting room, where groups of people come together to engage in a task. In the EkSarva framework the place supports both synchronous and asynchronous collaboration. The place is considered, to be persistent throughout the collaboration. The concept of place is realized as several instances where a person is part of multiple collaboration instantiations.

2.3.4 Signal (S)

In EkSarva framework signaling is a key mechanism. Completion of tasks and achievement of situations generate signals to indicate their status and the framework proceeds with newer tasks and situations.

2.3.5 Smart Transcript (T)

It is an object that links all the situations that are part of collaboration. One could imagine the transcript as the collaboration instance memory. Transcripts could be used by the persons for archival and references. Transcript plays a major role in enabling the persistency of the Place. In situations where members of a collaboration instance are unable to participate in group activities due to failures like network failure, system failure then the transcript could be used to recover.

2.4 General Problems in Collaboration and EkSarva

Even after all these there is no clear picture of Collaboration or EkSarva has yet emerged from the CSCW community. With a few exceptions, current collaboration systems support involves localized solutions to specific domain problems that are difficult to generalize to other situations. So groupware designers have little principled information available to them about how to support collaboration in other domains and also in new systems. In a co-located and co-related setting, a lot of information about the other people and their activities by asking simple questions or just by being present in the same place. We can take a glance at what they are doing, when they are doing, and interpret what they are saying in light of their body language and so on. In a distributed setting we lose much of this information. To make up for this, there is a need for computer systems to provide such collaboration.

Now consider the following problems involved in collaboration:

The percentage of ownership

How many types of resources are shared?

How workload is shared or divided?

The privileges and rights to use the work

These are the sample problems faced by groupware designers setting out to support collaboration: what type of information to collect and to distribute. Also how to present the information to the group, and when the information is useful.

2.5 Validation of the Collaboration (EkSarva TestBed) Model

Let us begin by examining what it means to evaluate EkSarva TestBed and follow this by specifying evaluation criteria. Next we describe an evaluation methodology is proposed with an experimental framework. The architectural details of the experimental framework are explained in the chapter3.

2.5.1 Evaluation of CSCW systems

Evaluating any awareness systems is critical in understanding whether or not an innovative new system provides awareness. Considering evaluation of CSCW systems is a challenge due to various factors. There are many different kinds of activity that one might consider performing under evaluation related to CSCW systems. Evaluation of the CSCW system is not complete by verifying and validating just the functional and non-functional requirements such as reliability, performance, usability, and security. As these systems are used in a variety of application domains evaluating the impact of the system on the outcome and overall quality of group work becomes essential. This evaluation is closely tied to the way the system is used by its users in

their work. The evaluation of collaboration too is no easier than any other aspect of CSCW evaluation.

2.5.2 Evaluation of Collaboration (EkSarva TestBed) in CSCW

Evaluation of Awareness has proceeded along the type of awareness support, the quality of awareness support.

a. Evaluating Type of Collaboration (EkSarva TestBed) Support

One could evaluate how well a groupware application or tool supports specific types of collaboration. The propagation of the relevant information to the users must be studied and analyzed.

b. Evaluating Quality of collaboration (EkSarva TestBed) Support

A groupware could evaluate the quality of the type or types of collaboration it supports. One could evaluate the system's mechanisms of informing the user, is it audio or only visual or a combination of both and quality factors such as relevancy, frequency and volume could be evaluated.

c. Evaluating Impact of collaboration (EkSarva TestBed) Support on Work

Let us considering that there is close relation between awareness and collaborative effort it is essential to evaluate in what manner collaboration support provided by a groupware or tool impacts work.

2.5.3 Evaluating the impact of collaboration Quality on Group Work

As discussed above that collaboration acts as an enhanced active directory of collaboration information sources and this directory aims to empower users to choose the quality of awareness to enhance the work. The directory not only shows what is available but the meta information it provides about each source indicates the quality of collaboration available.

2.6 Validation Methodology

The choice of validation methodologies for the collaboration model is as follows:

2.6.1. Implementation and Testing of Prototype

This involves creating a proof of concept implementation of the Collaboration (EkSarva TestBed) Model. The implementation would have to be deployed in a real-world group work environment and tested using the criteria described in the section above. Such testing would require that suitable mechanisms be employed to collect metrics of how users accessed and used the Collaboration (EkSarva TestBed) Model during the course of their work. A complete picture of the overall impact the EkSarva TestBed Model on group work would be obtained by collating the results of all users' interactions with the EkSarva TestBed Model and would give the complete picture of the impact of the EkSarva TestBed Model on the quality of work.

2.6.2. Simulation

Verifying and validating a system's dynamics through simulations is a popular approach and an alternative to testing an actual implementation. A model of the system is created with the EkSarva TestBed Model as essential components. User's behavior towards accomplishing the group's work and consequently their interaction with the EkSarva TestBed Model is simulated to ascertain the system's dynamics. Similar to a full-fledged implementation simulation data is

collected along the above criteria. The data collected must indicate the impact of using the EkSarva TestBed Model on the user's behavior. After the careful study of the comparison of the Full- implementation and simulation, a hybrid approach is chosen as the method of validation.

2.6.3. Hybrid Approach to Evaluation

Our approach is a hybrid of both the above traditional approaches. The EkSarva TestBed Model is implemented fully. However the elements of the EkSarva TestBed Framework such as the medium, and the applications (sources of information) are simulated. Human users are required to interact with this system, which is emulating an EkSarva TestBed Model and simulating the rest of the EkSarva TestBed Framework. Their interaction will proceed according to pre-defined scenarios designed to evaluate the EkSarva TestBed Model. However users are free to act in a natural manner with the applications. On the other hand the advantages of simulation such as reproducibility, automating some aspect of the human behavior by using agent programs, as well as saving in time and resources is obtained. In the next chapters we will describe about PIA (Personal Information Agent) client and server architecture, GUI for client PIA and its functionality by using EkSarva TestBed architecture

Chapter 3

EkSarva architecture and Scenario description

3.1 EkSarva Architecture

This architecture is defined by Dr.Reddy and his research team members. The architecture of EkSarva consists of three main components: collaboration manager, context manager, and transcript manager.

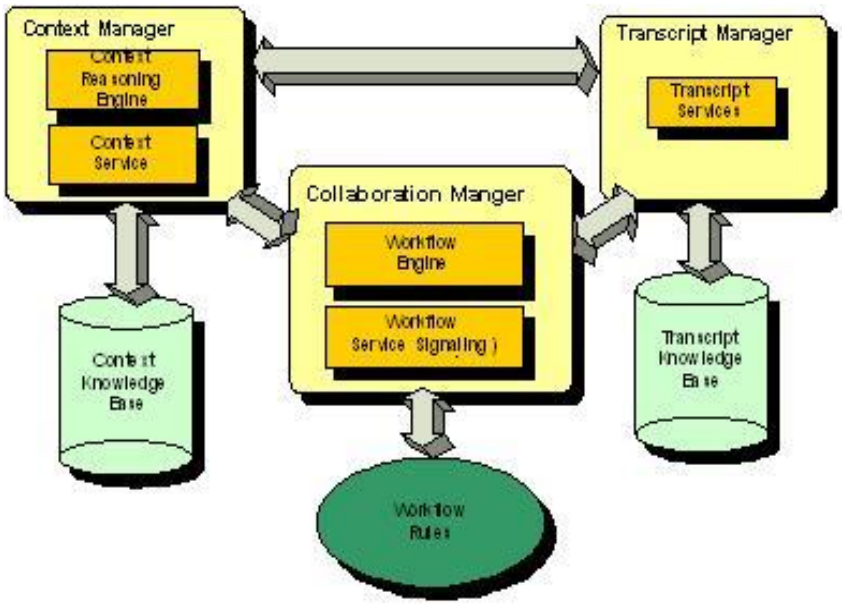

Figure **1.** EkSarva Architecture

3.1.1 Collaboration Manager

The collaboration manager is the main command center for any collaboration project. It consists of Workflow engine and Workflow service (signaling). The main function of the collaboration manager is to drive the process forward to next phase until the final goal is reached. It is responsible for all the collaboration embedded in the workflow rules by providing coordination,

synchronization and communication. The core of collaboration manager is a workflow decision engine that makes decisions to control the flow of the collaboration process based on different collaboration situations including changes of contexts. Workflow decision engine sends commands (signals) through a workflow service component. All workflow rules enforced in the collaboration process are stored in the workflow rule knowledge base. The knowledge base is the key to all the workflow.

3.1.2 Context Manager

Context Manager is responsible for maintaining context-awareness through the entire collaboration process. The core of context manager is a context reasoning engine, which reasons context information through a set of well-defined common ontologies stored in the context knowledge base. It provides context services such as context adding, updating, querying and reasoning to other components. Context manager enables the EkSarva framework to adapt to the changes of contexts. Therefore the changes of collaboration contexts such as heterogeneous computing environments are made invisible to collaboration participants.

After the EkSarva collaboration engine starts running and the collaboration process is initialized, collaboration participants can join the collaboration session through a registration process. During this registration process, the client tells the collaboration server what are his/her interests and other context information. Meanwhile, the client also registers his/her computing environment, including computation power, display size, connection type and speed, etc. This information is then assigned to the current computing environment property of the client and stored in the context knowledge base

3.1.3 Transcript Manager

Transcript manager is responsible for generating and maintaining Smart Transcripts. Transcripts are not necessary scripts. They can be database entries as well. Transcripts are used to record information about the activity of the collaboration, such as states, action items, important milestones, all the user activities, etc... Transcripts could be used to make references to past events. All past collaboration documents and files are stored in the transcript knowledge base.

3.1.4 EkSarva Collaboration Engine

EkSarva Collaboration Engine is responsible for enacting the collaboration embedded in the workflow rules by providing coordination, synchronization and communication. Thus the two-stage process is analogous to a compiler of specification and a run-time engine for instructions generated. Mechanisms that are required to implement the above mentioned architectural components can be seen in a layered component model. Each layer corresponds to a software module providing the necessary services.

3.1.5 Database

The database in EkSarva plays a vital role. It consists of all the user data, files and their privileges in the form of database. Also the database is used to store the various artifacts involved in the process of collaboration.

3.2 Scenario Description

To bring into light the effective usage of context awareness principles and generic workflow structure, collaboration work in three different environments is projected. Collaboration work between different projects with respect to varying organizations is the best scenario where EkSarva goals can be portrayed. Individuals in an office have different roles and perform

different tasks. And a user may work in more than one organization. Individuals may work on their own or work as part of a group under different organization. Users may be using a variety of applications based on their work, capabilities, availability and preferences. Users may have different access privileges and services depending upon the context of the user. Groups use different group ware systems to collaborate and get work done. Thus EkSarva framework fits in well with this scenario. However we limited the services of the groupware to Synchronous communication, Asynchronous communication, File Transfer, Personal help applications like To-do list, Favorites, Phone book. It is often seen that in domains like software, university, healthcare, military, engineering all individuals and groups use the same set of applications creating homogenous environments.

Collaboration work between different projects with respect to varying organizations is the best scenario where EkSarva goals can be portrayed. Individuals in an organization have different roles and perform different tasks. Individuals may work on their own or work as part of a group. Users may be using a variety of applications based on their work, capabilities, availability and preferences. The environment in which the users work may have varying privilege models say rigid, semi rigid, non rigid. As a result of which, groups of individuals are empowered with different type and level of awareness propagated. Thus the EkSarva TestBed model addresses the larges issue of collaboration with different groupware systems and individual applications.

3.2.1 Office Collaboration Scenario:

We employed Office Collaboration Scenario for simulation in this project. In this scenario, there are number of Employees/users who are located at different remote locations. Each employee may work in more than one organization and each has access to the all the organizations they are working through a single GUI assuming that all the organizations are adopting our EkSarva

TestBed concept. All of them work with their workstations at home or in the office and have access to the internet and are also connected to other organizations through dedicated networks or internet. They can communicate with others by using their instant messenger client, email and telephones for communication. All of the Employee participate in various activities. Each user can work on all the organizational projects same time and can access all the resources, services by using PIA.

Example scenario:

In our project we are considering the following organizations

Software Organization

University Organization

Nonprofit Organization

Considering the Software organization, it has three different projects namely "MSDN","JCP","COLLABORATION". There are different users in each of the projects. The list of users and the groups to which they belong to are listed. These users have different privileges and rights.

Similarly the University Scenario has three different projects namely "NASA"," ELEARNING","AIMS" and in the Nonprofit Organization "Tsunami","AID". People in each of the organization have different tasks, and use different services for collaboration. Based on the privileges assigned to a user they have rights on the artifacts for e.g.: User A has a privilege of say "3" and a file say "XYZ" has a access privilege of say "1" .User A will not be able to access the file "XYZ", which only a user of privilege equal to or greater than "3" can have access to.

When it comes to collaboration the users of one project within an organization may have to collaborate with other projects and sometimes collaboration may take place between organizations too. But restrictions exist here as well; Say the artifacts in a NASA project should not be passed to any other organization not even to the other projects within the same organization.

Our test bed was created such that it accumulates all these privilege models, artifact access models right at the creation of the projects itself. So context awareness is being established right from the beginning. The following the test or use cases we are implementing.

3.3 Use/Test Cases

3.3.1 Test Case1

User "A" ticks the rule for immediate notification of user "B" sending any artifact to user "C, the context here is set to A being a Manager in Software Organization. This makes A to get all trace of B's work with C in the Software Context, but he is not aware of anything that is out of his context.

User "A" ticks off the rule for immediate notification of C sending information to D. This makes him unaware of any action in Research context.

The intention of this scenario is to reflect the rule on Users B and C only in Software Development context. This application is giving the flexibility for the user to keep his private space organized only by changing his local rule option

3.3.2 Test Case2

User A ticks a rule for rerouting in Research domain; this will get the actions intended for him to be rerouted to his Graduate Assistant C. So a student who sends her homework to A ,will be rerouted to GA

The reply sent by the GA will be posted to Prof A as well as to the concerned student

If the GA is offline then the Synchronous communication prompts the user to give an Asynchronous message

The notification of Professors absence and the rule settings by the Prof will be sent to all the students automatically.

Things that are shown with this use case are a principle of Context sensitive rerouting.

3.3.3 Test Case3

Based on the sensitivity level associated with the item the sender of the item should be notified , if for e.g. an urgent item is not accessed or if there is no action associated with it in the receivers side for a particular interval of time

The time interval can be chosen by the sender of the item

Things proven here are automatic tracing for the actions performed. Manual configuration for the tracing time, Context aware traced notification

3.3.4 Test Case4

The software organization follows rigid user restrictions by setting rules on their privileges, on the other hand Non profit organization has no restriction s on the user actions So when an artifact or an action in the context of Software organization must undergo Context aware reasoning.

3.4 EkSarva TestBed Server architecture:

EkSarva TestBed architecture is based on client-server architecture as shown in **figure.** This figure gives the brief description of the architecture. The architecture is designed generic so that it can satisfy all the above EkSarva TestBed requirements. This architecture resembles the proposed EkSarva architecture in all the aspects. In this client and server communicate with each other using Web services. All the services and modules in this architecture are explained below:

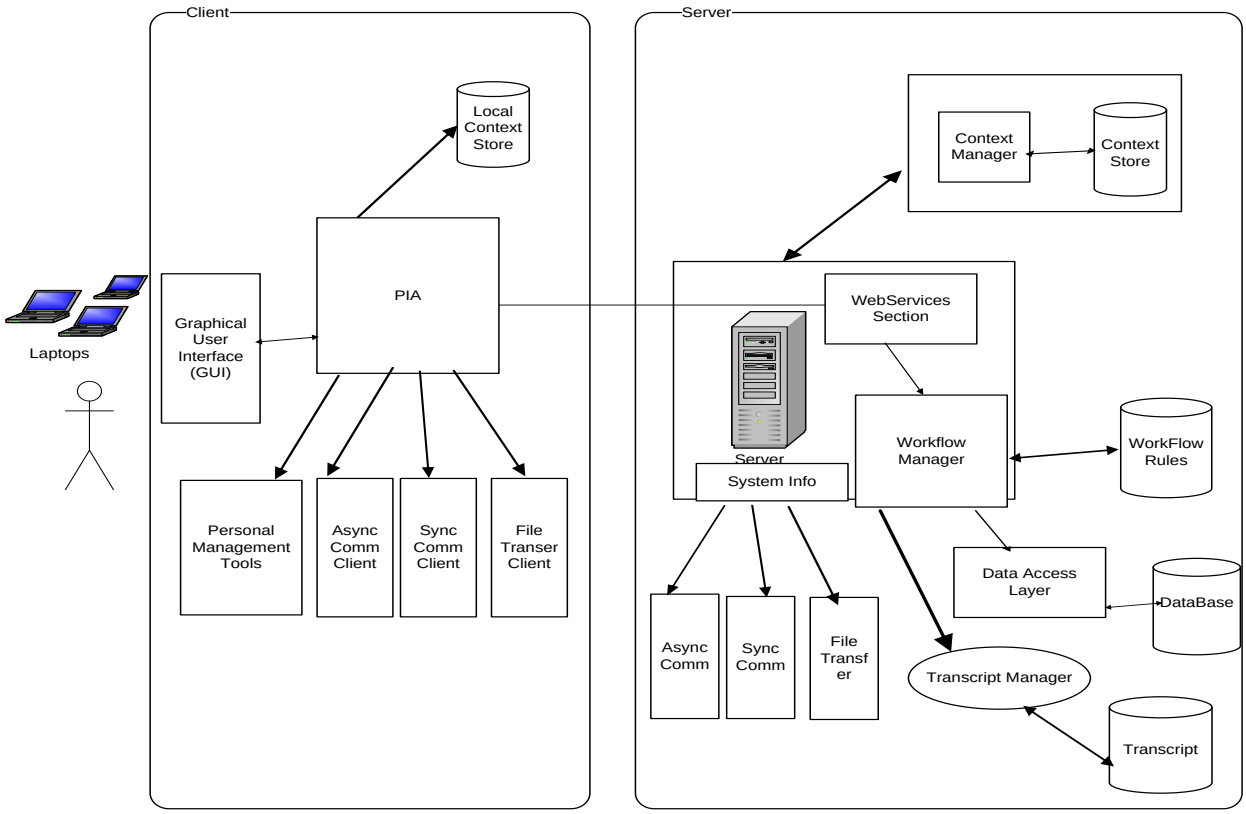

Fig: Client-Server Architecture

Server and client Architecture Modules: Server architecture consists of the following components:

Context Manager:

Context Manager is responsible for maintaining context-awareness through the entire collaboration process. This is the key module in EkSarva TestBed architecture. This module is used to store the context of users in client and servers. The core of context manager is a context reasoning engine, which reasons context information through a set of well-defined common ontologies stored in the context knowledge base.

Web Services:

Web services play a vital communication role among client and servers. Web services are application components and communicate using open protocols. They are self-contained and self-describing. Also they can be used by other applications. XML is the basis for Web services. So all the communication in EkSarva is carried back and forth by Web services. They take the context requests from client side PIA and sends necessary information back to the client PIA. The web services establish connection with database to get the data requested by PIA. Each web service is dedicated for each task like web service for user authentication, web services for workflow… .

Workflow Manager:

The workflow manager manages all the workflow in server and client. Workflow at its simplest is the movement of documents and/or tasks through a work process. More specifically, workflow is the operational aspect of a work procedure: how tasks are structured, who performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked.

Database:

All data pertaining to EkSarva is stored in the database tables. A number of tables are stored in the database to contain user information, context information. For each collaboration activity that occurs in EkSarva a call to the database is made.

Collaboration manager:

The collaboration manager is the command center of a collaboration project. It drives the process forward to next phase until the final goal is reached. It is responsible for enacting the collaboration embedded in the workflow rules by providing coordination, synchronization and communication.

Transcript Manager

Transcript manager is responsible for generating and maintaining Smart Transcripts. They can be database entries as well. Transcripts are used to record information about the activity of the collaboration, such as states, action items, important milestones, etc... Transcripts could be used to make references to past events.

PIA:

PIA is the acronym for Personal Information Agent. PIA plays a vital role on the client side of EkSarva. It renders the GUI and monitors all the clients' activities. It is an intelligent agent working for the client and manages all the clients' information, reducing clients burden.

GUI:

GUI is the interface through which client interacts with the PIA. GUI consist of all the services ,users , projects , groups and all information a user wants when he logs in to a particular project. It provides features that are flexible for the user to work in a collaboration event. Miscellaneous information is also provided for the users.

Synchronous communication:

Services for synchronous communication are provided. Though in our problem report we are not focusing on fully fledged building of the service, we tried to provide minimum features to the services. Chat system is the synchronous communication we are providing to the users.

Asynchronous communication:

E-mail system is the asynchronous system that is provided in the EkSarva test bed. In this a basic email system is produced.

File Transfer:

Files can be transferred between users in projects. Peer to Peer file transfer as well as file transfer between client and server takes place. Transfer between client and server is restricted by the privileges the user has. Context awareness has be demonstrated through this communication.

Local Context:

Each user has an ability to customize his projects, form views. This information is stored in the local context store.

Privilege Model: Privilege model is defined on the basis of user roles. A user may have higher access levels than the other users.

Chapter 4

4.1 Introduction

In the previous chapter, the EkSarva test bed architecture was explained. The Client is a key part of this architecture. It is the interface through which User's interacted with the PIA and receives information from. The EkSarva Client block diagram is explained in detail in this chapter.

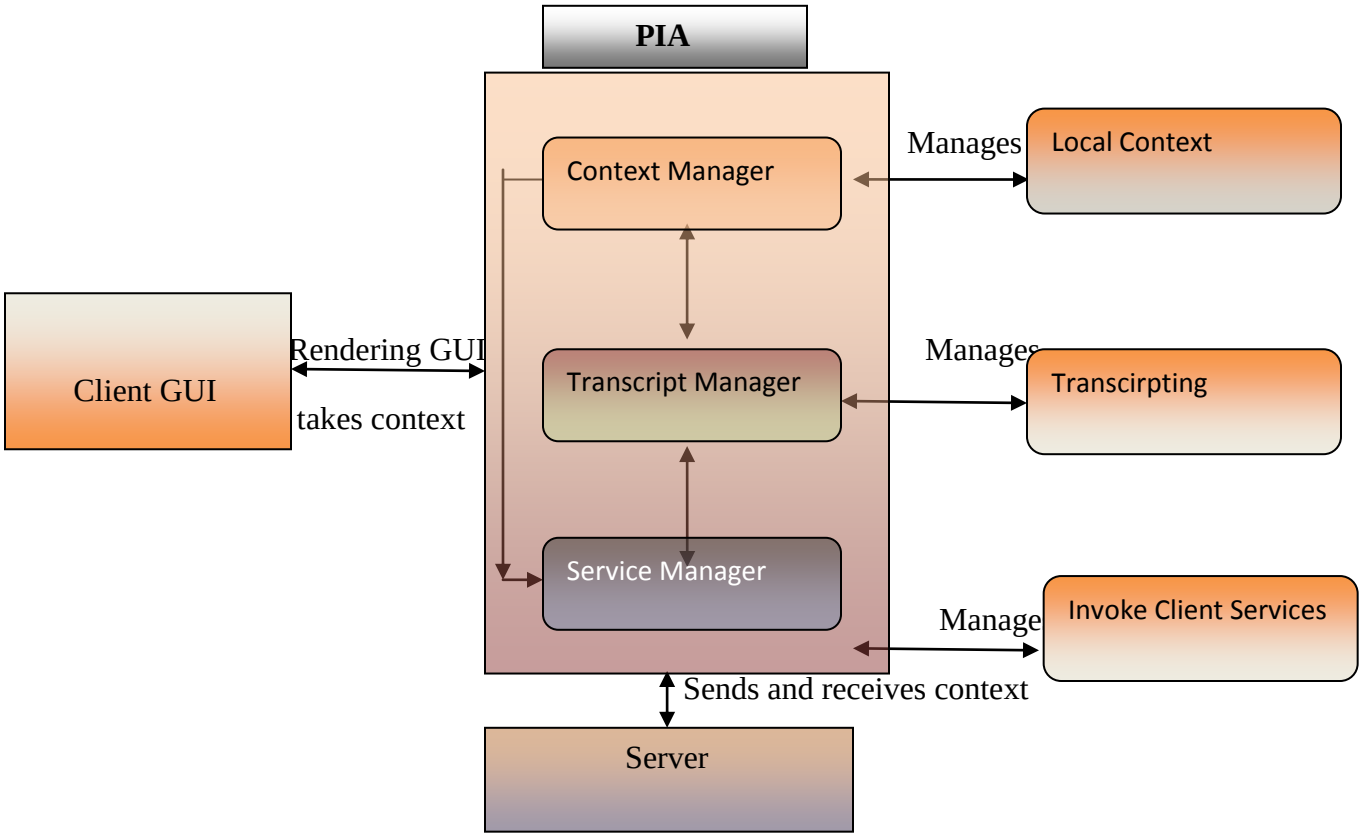

Figure: Client Block Diagram

4.2 PIA (Personal Information Agent)

PIA [Personal Information Agent] is the key component in the client architecture. It is the intelligent agent that works on the behalf of the user in his/her own personal space. It invokes the various services to the clients. The interface that comes to the user is managed by the PIA. It does the searching part on the behalf of the user. To do list activities, personal calendar activities, Google search are carried out by the PIA. The PIA is solely responsible for managing information at the user side. For example setting up the color of the user's choice is maintained by the PIA. Hence the user interface is very powerful with the PIA on its side.

4.3 Features Supported by PIA

4.3.1 Boundaries

To establish the feature of context awareness we came up with the concept of boundaries which provides a way to organize information resource and services around important privacy relevant context awareness. A boundary is one that delimits the owner's information space. Each information space has an owner who has permission on it.

The boundaries of information places can be identified by

Demarking physical boundaries through location awareness

Demarking social and activity-based boundaries through identity and activity awareness

For e.g.: Professor Bob can create an information space that contains all information and resources in his office. He doesn't want his family members to access it which might lead to unnecessary alterations.

Thus the concept of awareness can be established depending upon the context he /she is in. To implement the concept of "boundaries" "Access tags" is to be used. PIA looks for the access tags

and sends them to the server where the privileged are checked from a database and workflow is carried out. A user can have different kind of privileges on items based on the context he is within in.

4.3.2 Access Tags:

Privacy tags are associated with objects which specify the objects privacy requirements. Each tag has two parts: an owner of the object and the reader of the object. A privacy controls readers are a set of principals who the owner permits to access the data item. It is implicitly understood that a privacy controls owner can have both read and write permissions on that data item.

4.3.3 Importance and Sensitivity Levels

EkSarva allows users to set the importance and sensitivity of messages. Importance refers to the priority of a message, and three levels of importance can be assigned: low, normal, and high. Messages assigned low importance display a down arrow. Messages identified as high importance automatically display an exclamation point.

Sensitivity defines the personal nature of a message, and the setting typically dictates how other users can respond to the message. The values for sensitivity are normal, personal, private, and confidential. If a user sets the sensitivity of a message to private before sending it, recipients cannot edit the message text.

4.3.4 Delayed Delivery and Auto-Expire

EkSarva supports two features that control the delivery and availability of your messages: delayed delivery and auto-expire. Delayed delivery allows the user to specify a date and time when PIA should deliver a message. Auto-expire allow the user to set the date and time a

message becomes unavailable. Both delayed delivery and auto-expire are available by clicking the Options button.

4.3.5 Personal Information Management

Personal information management tools like To-do lists and Calendar are managed by PIA. Users can add and remove appointments from the list. This feature serves as a reminder for the users of various activities they need to keep track of in the projects they are involved.

All these activities are carried out by PIA to bring out context awareness in the system.

4.3.6 Local Context

User is given privilege to makes customization on the GUI. Though we are limiting this feature only to the color setting, user customization can be extended in many ways. This information is stored on the client's local context and has nothing to do with the other users. PIA takes the information from the local context of the user and renders it to the client.

4.4 Client GUI

A lot of effort has been put in designing the GUI for the client. The collaboration has to be carried in the background, and workflows should be executed in an unobtrusive, manner without compromising the privacy and security constraints of the group. The user should be provided with appropriate information relevant to the user's sphere of activity. **Figure 4.2** Represents GUI for the user.

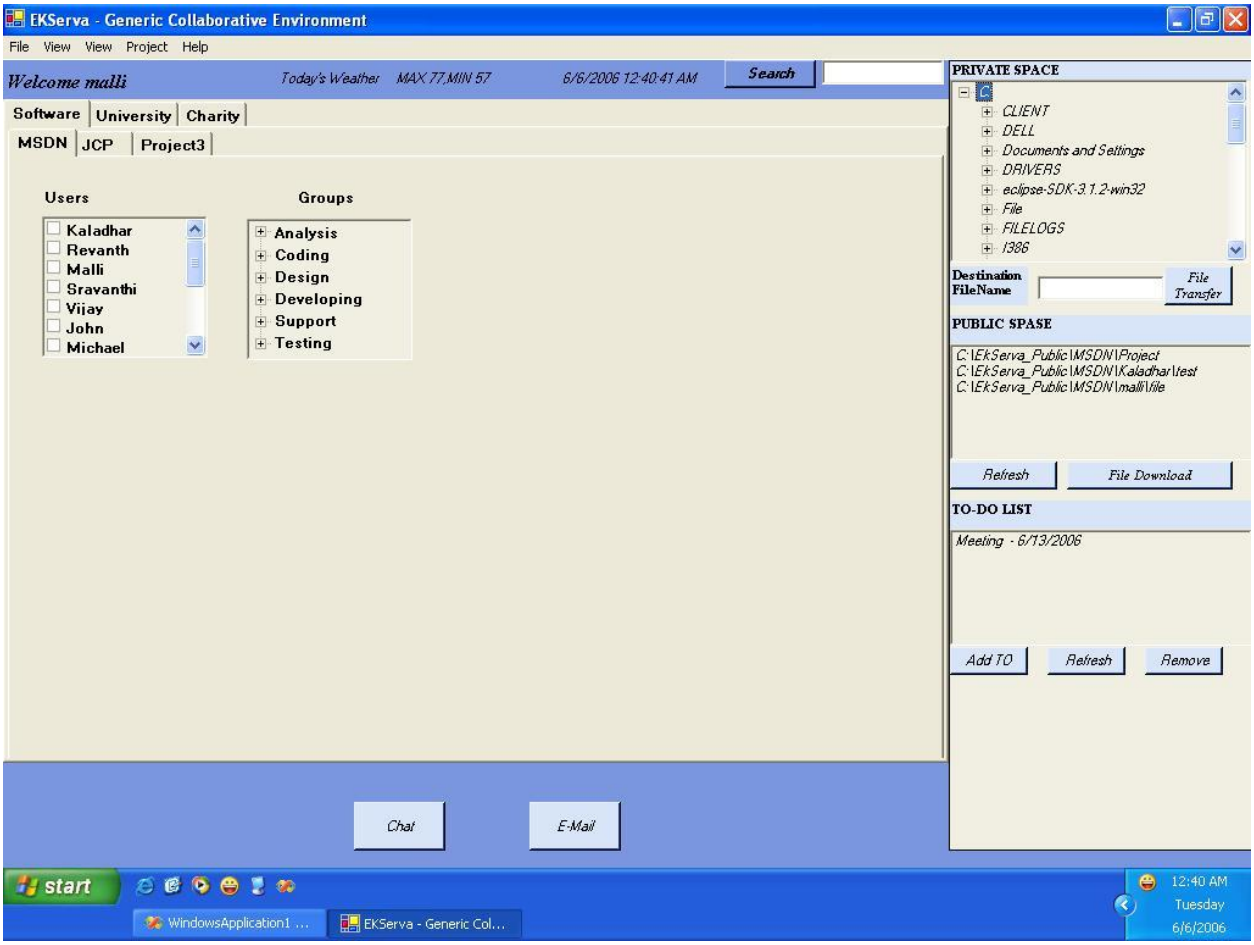

Figure: GUI screen shot

Organizations: Based on the user login the organizations that the user belongs to get populated. A user may work in more than one organization and can access all the organizations at the same time. Each Organization consists of Projects and Services. When user log into the GUI, all his organizations and projects are selected in the GUI.

Services: Users get access to a list of services that he is entailed to. The list of services are loaded based on the project he is present. We implemented basic types of services including

Email, Chat, File transfer (file upload and download)…. Depending the user context and preferences, services are invoked and user can customize the services.

Personal Information Management: Various tools have been implemented to manage the user personal information. A user can manage his personal information according to his preferences. We implemented To-do list, Address book, My Favorites…. By using all these components a user can manage his information effectively.

Public space and Private space: This is shown on the right hand side of the GUI. To organize the files and folders relating to user and project are managed effectively by using personal and private work spaces. A Personal work space is located at the user personal space and Public work space is provided in each organization server. A user can share his files to other users by keeping them in the public space. Public folders access can be done using the user roles and privileges. All sharing of artifacts is done using public and private work spaces.

Users & groups: This is shown on the left hand side of the GUI. It lists all the users within the project. Based on the privileges the users within the list have different colors associated with them. The User will be able to see his own sources as well as the common sources like the calendar in the Source Super Set. Similarly all participating users within a group are listed out for performing group actions on users.

Miscellaneous: We provided the GUI with some services like Google search, Weather information, Date and Time….

User Actions: The User can login into different projects and perform actions. For example he can switch from one project to another project and see his friend's online working in that project. He can switch back to his workspace, with all the users and groups pertained to that project. The User has to switch on the source and specify the source when he intends to send a message.

Programming Language used: I have chosen VB.Net to implement the client GUI and MS SQL server 2003 to maintain the user data and files. I implemented 2 test cases, built the database and GUI.

Implementation issues:

I used VB.net 2003 and it does not support the concept of Winsock and hence working on peer to peer file transfer has become tough. So I used client server file transfer technique rather than peer to peer approach.

I had to learn the Oops concepts of VB.net for implementing the project. And it took some time for implementation of the project.

Chapter 5

Conclusions

Our TestBed and PIA model meets the design and concepts of EkSarva.

A PIA Client and Server is developed which meets all the requirements necessary to evaluate the proposed EkSarva TestBed Model.

A PIA Server works same as client PIA with additional client monitoring and managing all the global workflow and context awareness.

PIA client manages all the user local workflow.

Actors interacting with the client can be human or intelligent agents.

Multiple users can interact with the PIA at the same time because of the multi-threaded client server architecture.

All the work has been done using VB.net and MS SQL server to make the application more efficient and user friendly.

Web Services allows us to connect to server from anywhere in the world.

By using PIA, a user can connect and interact with any user in the world. One can organize his personal and office work using PIA.

Our PIA is so sophisticated that it can monitor and handle all the user work using work rules and context awareness.

Web Services made the application more portable, reusable and interoperable. These are used to exchange data between clients and servers irrespective of the platform.

Future Work

We created an EkSarva TestBed with simple services embedded. This can be improved to host fully developed services.

Current TestBed is developed based on the assumption that every organization is supporting the EkSarva concept and makes the TestBed more generic.

No security issues have been considered and this can be improved to more secure and efficient like authentication, encryption, etc...

Build more generic intelligent PIA which may helps the user more effectively.

Current PIA can handle only three organizations and make it handle any number of organizations.

PIA client and server GUI's has to be more user friendly.

The Graphical User Interface can be improved to represent more services and functions in an effective way.

The Actor type is currently is either human or intelligent agent. This can be improved to Human-Agent hybrid with the Human user being able to override the agent

Current work flow rules are simple and straight forward. So implement sophisticated work rules. This may help in context awareness and workflow.

Current PIA is working based on assumption that all the data and rules are predefined and exist in the organizations and its Projects. So develop more sophisticated setup wizard to create all the organization flow.

Improve Role and Privileges model and make it generic.

References:

1. Adaptive Collaboration in an Ever-Changing Environment - Jinqia Yu, Y.V.Ramana Reddy, Vijayanand Bharadwaj, Sumitra Reddy, and Srinivas Kankanahalli

2. Dr. Reddy's Research papers on EkSarva from IEEE Journal

3. A framework for collaborative systems by Charles Sheppard

4. Context Awareness by Gregory D. Adowd.

5. EkSarva: An Intelligent Collaborative Framework by Ramana Reddy, Senti1 Selliah, Vijaymand Bharadwaj, Jinqiao Yu, Srinivas Kankanahalli, Sumitra Reddy from IEEE journal.