Hermes: A Platform for Context-Aware Enterprise Communication

Ajita John
Avaya Labs Research
New Jersey, USA
ajita@avaya.com

Reinhard Klemm
Avaya Labs Research
New Jersey, USA
klemm@avaya.com

Ankur Mani
Arizona State University
Arizona, USA
ankur.mani@asu.edu

Doree Seligmann
Avaya Labs Research
New Jersey, USA
doree@avaya.com

Abstract—This paper describes our vision for next-generation context-aware enterprise communications that achieves the integration of backend business processes with user communications. We envision a new class of enterprise applications in which communications between users in response to a variety of enterprise events will be an automated process in which the right group of users will be selected for communicating at the right time on the right media. To achieve this goal, the applications must exploit a variety of context information such as enterprise knowledge, user knowledge and application knowledge. This paper describes a platform named Hermes that enables the creation and execution of such context-aware applications. In addition, it briefly describes experiences with using the system in several demonstrations.

Keywords: Context, context-aware communication, workflow, presence and availability.

I. INTRODUCTION

In this paper, we present our vision of context-aware enterprise communication and our platform for enabling creation and execution of context-aware communication applications. Communication among people in an enterprise has been a means for knowledge transfer, knowledge generation, and decision making. Significant shifts have been occurring in the recent past that impact communication in an enterprise. The virtualization of enterprises through globalization and expansion has necessitated the increasing dependence on technology for communication. The proliferation of communication devices on different media, the use of multiple devices of varying capabilities by people, the mobility of people, the permeation of work into different environments such as the home, hotel, and the road which are conducive to varying degrees of the use of specific types of media, and the increasing specialization of people have made it difficult to manually identify who, when and how to contact people in response to an enterprise event. Additionally, there is a strong need to bridge the gap between backend business processes that compute, update databases and push content to web pages and the decision making processes that users constantly engage in through communications in an enterprise. Context aware communication in such an enterprise cannot be independent of the knowledge about the structure and processes of the enterprise.

A. Our Vision

We envision a modern enterprise that integrates context-aware computing and communications with the knowledge management and business processes in the vastly expanding landscape of possibilities. To address this, we have built a platform named Hermes for creating and executing rich context-aware applications that tightly integrate backend business processes with communications. The communication applications on Hermes are modeled using workflows as defined by the Workflow Management Coalition (WFMC) [1]. The nodes of the directed graph in the workflows represent communication tasks that use the context of the enterprise, current task and the users in a reasoning process to make the effective selection of the media, time, and people for a communication interaction. The tasks, at the selected time, connect through a sessions and connectors infrastructure to the selected media servers and set up communication sessions such as a notifications, responses and conferences among selected groups of people.

Hermes integrates the applications with backend business processes thus providing for closed loop enterprise systems where processes are controlled and monitored. The closing of the loop is essential context modeling as context is a dynamic construct [3]. This is in sharp contrast to open loop systems where communications such as an email or voice call may be initiated by a backend process, however, little is done to monitor and react to the outcome of the communication event. The integration of communication with the business processes and context-awareness provides:

• the tractability of the work threads in an enterprise and guarantee of delivery of multimodal messages and connections among people.
• better auditing of enterprise processes leading to enhanced reporting.
employee satisfaction through reduced interruptions and balancing of load

B Related Work

Platforms for creating context-aware applications have been discussed in [2,5,8]. These works consider context as independent pieces of information. However, something is context because of its relationship to a larger set of interrelated facts and thus affects decision making [6]. Thus, while these works provide support for collecting, disseminating and processing contextual information, the application developers themselves have to provide for processing and using context. Similar efforts in [4] use pieces of information to make decisions about connecting people. We focus on a set of communication applications that are embedded in the working environment of the modern enterprise. We use the knowledge in the enterprise about the work practices, hierarchy, user preferences and others and are able to associate the information collected with the given knowledge. This association provides the required context to make decisions. There has been some effort in using such knowledge such as the workflow for multimedia conferencing [9] but it is restricted to a specific application.

The organization of the rest of the paper is as follows. In section II, we present an enterprise communication scenario that motivates the need for a context-aware platform. Section III discusses our context-aware platform, Hermes and the representation, acquisition and use of context in Hermes. Section IV discusses our experiences with Hermes and conclusions are given in section V.

II. AN ENTERPRISE COMMUNICATION SCENARIO

We present an enterprise communication scenario that illustrates the key motivations for Hermes. This scenario has been drawn from an extensive case study that was conducted with several insurance companies. It evolves around a scenario in which person-to-person barriers are reduced and users with access to multiple media across a global enterprise (see Figure 1) are brought together to solve a specialized and time-sensitive issue like the resolution of a claim in an insurance company by people such as a claim appraiser and an expert. We first discuss how such a scenario in handled by an intelligent context-aware middleware and then provide the contrast in a manually handled scenario. Similar scenarios have been observed in other domains such as health-care where appropriate resident doctors, nurses, and specialists are connected to respond to a patient emergency.

A Intelligent Communications using Context

A Claims Handling business process in an insurance company detects that the deadline for an unresolved car accident claim from a customer is near and triggers a communication application named Flow 1 (Figure 2). This flow contacts the claim appraiser via phone according to the appraiser’s user rules. If the appraiser does not respond, Flow 1 proceeds to escalate the problem by sending invitations for a conference, to be started in say, 2 hours, to the appraiser, appraiser’s boss, and the customer. It sends the invitations using a combination of presence and availability information, user rules and enterprise policies. User rules include preferences that specify how users would like to be contacted at different times of the day. This may mean that, for instance, the boss receives the invitation via email, the appraiser via voice, and the customer via IM. The invitation can be derived from a template that includes information about the claim. The recipients may respond through the phone or through a user portal. If a quorum of people accepts the invitation, Flow 1 establishes a conference with the three invitees at the scheduled time by reaching out to them. The medium of the conference (E.g. voice or IM) may be determined based on the needs of the collaboration, the presence and availability information of the participants and the media capabilities of the devices of the participants.

Figure 1: Users in a global virtual enterprise with access to multiple media may need to be drawn into an interaction

Figure 2: An example of a Communication Flow

During the conference, the boss needs an auto collision expert. There are many such experts in the global locations of the company and the boss does not know which experts have the necessary skills and are available. Experts with prior interaction with the customer or appraiser or boss may be preferred. Experts may be mobile with a Personal Digital Assistant with voice and text chat capabilities. However, the noise level in the expert’s environment may preclude any productive voice interaction or the interaction may require high attention levels that necessitates that the expert should not be participating in any other session. To automate the
expert finder process, the boss initiates Flow 2 which selects a possible group of experts based on skills, cohesiveness with participants of the conference, presence, availability, location and environment and sends out invitations to them. Depending on whether or not an expert responds within a time period, Flow 2 alerts the boss about the availability of an expert. Since the boss is present on an ongoing conference, this context information is used to send the alert. If the conference is a voice conference, the alert is rendered as a whisper and audible only to the recipient. If the conference is an IM conference, the whisper is an IM message. If an expert is available, the expert is bridged into the conference.

After the completion of the conference, the boss is sent an alert to update of the status of the claim. The update can be done either by voice interaction software or through a user portal. This step will update the appropriate backend databases in the enterprise resulting in the resolution of the claim and providing the bridge between the enterprise backend and human communications.

B  Comparison of Manual Process with Intelligent Process

The above scenario shows how the a system, that can intelligently connect the right group of people at the right time on the right media by reasoning about contextual information such as presence, availability, cohesiveness, past and current communication sessions, user rules, skills, location, environment, enterprise organizational information and policies can result in effective communication to address needs of the enterprise. Contrast the above study with trying to connect people manually where human delays and errors can be costly to the enterprise especially when the pool of people for selection is large, capabilities of the media used by the people is varied, and the people are geographically distributed. Intelligently automating the process leads to reduced latency in decisions and enhanced enterprise productivity and customer satisfaction. From a user perspective, it allows the protection of personal information such as presence, availability, and location from exposure to other users by allowing a trusted system to access the information. We now discuss the middleware platform we developed that provides support for creating context-aware enterprise communication applications.

III. HERMES COMMUNICATIONS MIDDLEWARE

We developed a communications middleware called Hermes that allows the design and execution of communication flows such as those presented above. The overall architecture of Hermes is shown in Figure 3. It consists of various layers that define, gather and use context for communication applications. In this section, we first describe each of these layers. Next, we describe the execution of a flow in Hermes. Finally, we describe the context information and their classification.

A  Design Environment: Providing Application Context

The design of a flow follows a visual workflow model in a graphical user interface. A flow designer selects nodes from a palette of nodes, drops these nodes onto a canvas, connects them according to the desired control flow, and configures each node for the use of context information by right-clicking on it and filling out a node type-specific property sheet. For example, the property sheet for an alert node will allow the designer to specify, among other things, whether presence, availability, and/or user preferences should be used to contact a user. It allows the specification of application context such as the customer claim information in Section II.A.

Figure 3: The Hermes architecture highlighting the various aspects of context in the system

B  Workflows with Communication Tasks: Use of Context to make Decisions

Workflows that are created in the design environment are stored in XML definitions. Each workflow is a directed graph of communication nodes. The graph contains branches and loops and represents the control flow of a communication application. The Hermes task library contains the logic for the nodes in the workflow. Aside from typical workflow types of nodes such as decision nodes, Hermes encapsulates complex communication tasks such as conferencing and alert nodes. The communication tasks access contextual
information from different sources and use this information in reasoning processes to determine who, how and when to contact and connect people. Hermes abstracts away the inherent complexities of programming to low-level, often proprietary, and technology-centric communication server and services APIs. Hermes communication nodes provide simple, easy-to-maintain, vendor-agnostic, and converged control over different communication media (voice, instant messaging, Web, email, etc.) and communication activities (conferencing, alerting, messaging, collaboration, etc.).

Hermes flows can be easily integrated into business processes either at design time through a shared workflow design environment or at runtime by giving business processes access to Hermes flows through a Web Services interface. Once deployed, Hermes flows execute on a workflow engine that is connected with Hermes J2EE components. The workflow engine sequences the execution of nodes in Hermes flows. Tasks use the services of the Hermes J2EE components as well as LDAP enterprise directories, databases, and other servers that collect and store relatively static contextual information such as enterprise rules, policies, user profiles, preferences, and roles. When tasks launch communication activities such as phone calls, conferences, sending of messages, they issue abstract communication requests to the Hermes request manager which queues up requests, logs them, and ties them together with the services of the Hermes entity relationship manager.

C Entity Relationship Manager: Aggregating Contextual Information

The entity relationship manager administers much of the contextual information that Hermes tasks rely on to reason about user presence, availability, past and present participation in communication sessions. Hermes communication sessions capture all the information pertinent to establishing or modifying the sessions through functionality in the underlying communication infrastructure, user participation and current user session activity, delivery of whispers, user feedback in response to whispers and notifications, and more. It is designed to reflect all the user and session communication activity that Hermes initiates, that Hermes can obtain knowledge of. The contextual information feeds into an entity relationship model that includes session, user, interaction, device and other entities and relationships between them. The entity relationship model is implemented as persistent Enterprise JavaBeans backed by a relational database. Hermes’s entity relationship management is converged across different media and thus allows uniform access to all the contextual information that Hermes administers as well as uniform control over communication activities (setup, start, modification, termination of communication).

D Agents: Gathering Contextual information

Hermes communication agents are attached to communication servers and devices and feed raw data about users and their communication activities through a Web Services interface into the entity relationship manager. For example, a communication agent attached to an instant messaging server can relay user presence information about instant messaging clients and conferences to the entity relationship manager. In another example, a Hermes communication agent can signal the arrival of a voice mail for a specific user to Hermes. Hermes tasks can execute sophisticated query and navigation operations against the entity relationship model.

E Context in Hermes

In this section, we discuss the definition, acquisition and use of context in Hermes. Context forms a subset of the knowledge about the enterprise, its structure and its processes, users and the user environment, and the application that is important for the current decision making process for the communication. Knowledge about an entity (here the enterprise, the user or the application) is a set of inter-related facts about the entity. As for example, in the scenario discussed in section II, the scheduling of a conference for solving a critical customer issue is a decision to be made. The decision making process involves answering three questions as shown in Figure 4- who should be invited to participate, when should the conference be, and what media and devices should be used for the conference. The pieces of information needed for this decision to be made are the presence and availability information of the users, the environment of the users, the skills, roles and positions of the users in an organization and the requirements of the current task (here the particular customer issue). These pieces of information and their relationships with each other and with the knowledge they are embedded in, form the context.

For the ease of representation and storage in a relational database, acquisition and use, all this knowledge can be divided into three classes namely the Enterprise Knowledge, the User Knowledge and the Application Knowledge. Note that, these knowledge spaces are not totally independent of each other and an overlap exists. We now discuss these knowledge spaces and the associated contexts in detail.

III.E.1 Enterprise Knowledge

The enterprise knowledge consists of the enterprise directory and databases which store information about the processes, deadlines, states of the processes, policies, concurrent and independent communication sessions, needs of the sessions (e.g. document sharing), and social network of the enterprise. The dynamic data about the current sessions, the current processes and the enterprise social network is obtained from the entity relationship manager in Hermes which is persisted in a database. Analysis of the
communication logs are used to create knowledge about the social network in the enterprise as discussed in [7]. The enterprise context consists of the subset of the enterprise knowledge relevant for the current decision (e.g. subset of the enterprise social network consisting of the people suitable for the given conference) and the values of certain variables such as details of the current communication sessions in progress. The enterprise social context helps in choosing the cohesive sets of users from the set of suitable and available users for a given conference as in [7]. The enterprise context also provides information for the current state of connectivity in the enterprise and certain book-keeping processes that may define additional constraints on time and medium for communication.

III.E.ii User Knowledge

The User Knowledge consists of the inter-related facts about the user. This includes static and dynamic inter-related pieces of information such as user preferences regarding reachability, social networks, designation, roles, skills, properties of the physical environment of the user and presence and availability on different media and devices. The static information such as the user preferences, roles, skills and designations are obtained from the enterprise directory. The dynamic information about the user presence and availability are obtained from the entity relationship manager. When such information is missing, the estimate of the user presence and availability is made by the presence and availability server using a statistical model as in [4]. The user context provides value for the suitability of the user for a communication event such as a conference, the availability of the user over a time period and media.

III.E.iii Application Knowledge

The Application Knowledge consists of the facts about the different subjects such as the user roles that participate in a communication (E.g appraiser, boss in section II.A), customer issues, workflows for the customer issues, hyperlinks (E.g. claim documents in section II.A), content of messages in the application, the requirements such as maximum time for service and other fields depending on the Application. This knowledge in stored in the workflows provided by the application developer and the context is estimated by the workflow engine. The application context provides conditions to identify the suitable set of users for the conference, the time period for the meeting and the set of media elements required for communication.

F Execution of Tasks

This section briefly describes the execution of tasks in the Hermes system. When a task launches a communication activity, the request manager routes the request through the entity relationship manager to start a new communication session or modify an existing one. For example, adding a user to a conferencing session results in a modification to an existing communication session. From the entity relationship manager, requests for communication activities percolate down to the extensible Hermes connector framework. It contains connectors that join Hermes with its underlying communications infrastructure consisting of communication and media servers, services, and devices. The connector framework insulates the upper layers of the Hermes software architecture from the specifics and evolution of the underlying communication infrastructure. Typically, this is a distributed, multi-vendor, and multi-media infrastructure. Each Hermes connector is written to both the connector framework specifications and to the specifications of a specific component in the underlying communication infrastructure. It translates an abstract communication request into a component-specific sequence of API calls or requests. The connector framework provides the services common to all connectors and allows the easy addition and replacement of connectors as the underlying communication infrastructure evolves. In this section, we discussed the platform for creating and executing context-aware enterprise communication applications. We now discuss our experiences with the platform.

IV. EXPERIENCES

Communication applications such as the one presented in Section II.A have been demonstrated on Hermes extensively at Avaya and at various public forums such as tradeshows. This section briefly summarizes the experience with Hermes gathered from about 200 demonstrations on the system by users who are familiar with it. The experience is described from the perspective of (i) a developer to design context-aware communication applications and (ii) a user that is contacted by Hermes to participate in an interaction.

(i) Developer Perspective: The time it took to configure a communication node (E.g. conference node to connect 3 people) in the design environment varied between 2-3 minutes assuming that the user understood the application requirements beforehand and the variables for the application had been setup prior to configuring the node. The time it took to develop an entire application such as the expert finder flow in Section II.A varied between 15-20 minutes. (ii) User Perspective: The time it took a user to provide a response to a conference invitation through a user portal (E.g. the expert response in Section II) varied between 2-4 minutes. The time it took a user to join an interaction triggered by Hermes such as a conference (E.g. the appraiser, boss and customer receiving the actual conference calls in Section II) was of the order of seconds. For the application described in Section II, the execution times were always dominated by the time it took users to respond to an invitation (2-4 minutes for each response step).

The above experiences demonstrate the effectiveness of Hermes from the perspective of the application and the enterprise. We still need to do statistically significant experiments to verify the modeling and use of context for the reasoning for communication. We believe these experiments will give us insight into enhancing the reasoning in Hermes.

V. CONCLUSIONS

In this paper we presented the architecture of a platform that serves our vision of the modern global enterprise. The platform models and uses context to make effective decisions for selecting the users, media and time for a communication task in an enterprise. We observed that this context is a subset of knowledge about the enterprise, the users and the application workflow. While we describe the various sources of context information that we use in our work so far, there are many issues that we do not study. We list few significant
ones as candidates for future work: (i) Conduct extensive studies over large user groups and study the effectiveness of the reasoning processes that are used in Hermes. (ii) Identify optimum reasoning strategies for scheduling communication tasks such as conferences. (iii) Conduct user studies to understand user satisfaction in interacting with Hermes.

VI. REFERENCES