The Connector - Facilitating Context-aware Communication

M. Danninger[1], G. Flaherty[1], K. Bernardin[1], H. K. Ekenel[1], T. Köhler[1],
R. Malkin[2], R. Stiefelhagen[1], A. Waibel[1,2]
InterACT Research, Interactive Systems Labs
Universität Karlsruhe, Am Fasanengarten 5, 76131 Karlsruhe[1]
Carnegie Mellon University , Pittsburgh, PA 15221, USA[2]

maria.danninger@ira.uka.de

ABSTRACT

We present the Connector, a context-aware service that intelligently connects people. It maintains an awareness of its users' activities, preoccupations and social relationships to mediate a proper connection at the right time between them. In addition to providing users with important contextual cues about the availability of potential callees, the Connector adapts the behavior of the contactee's device automatically in order to avoid inappropriate interruptions.

To acquire relevant context information, perceptual components analyze sensor input obtained from a smart mobile phone and --- if available --- from a variety of audio-visual sensors built into a smart meeting room environment. The Connector also uses any available multimodal interface (e.g. a speech interface to the smart phone, steerable camera-projector, targeted loudspeakers) in the smart meeting room, to deliver information to users in the most unobtrusive way possible.

Categories and Subject Descriptors

D.4.3 [**Information Systems Applications**]: Communications Applications.(http://www.acm.org/class/1998/)

General Terms

Management, Measurement, Design, Human Factors.

Keywords

Context-aware communication, multimodal interfaces.

1. INTRODUCTION

Computers are becoming more and more ubiquitous and seamlessly integrated into our everyday life. Considerable human attention is expended operating and attending to computers, and humans are often forced to spend precious time fighting technological artifacts rather than attending to human interaction and communication.

Having these computers anticipate our needs and provide us with relevant services and information would help humans to break the technological attention barrier and re-focus on meaningful human interactions. Such human-centered computational tools would especially be beneficial in meeting situations or technology-mediated communication.

Within the framework of the CHIL project --- Computers in the Human Interaction Loop --- we intend to develop context-aware, proactive computer services that aim to assist people during their daily interactions with others [1]. Rather than expecting a human to attend to technology, CHIL's goal is to develop computer services that are sensitive to and attend to human activities, interactions, and intentions. In order to act in a proactive, implicit way, these services should be able to identify and understand human activities.

In this paper, we describe the prototype implementation of such a CHIL service, the Connector. The goal of the Connector service is to ensure that two parties who wish to communicate with one another are connected at the right time and with the best media, whenever it is most appropriate and opportune for both parties. In order to address this task, the Connector service needs a detailed understanding of user state, activities, social relationships and communication urgency.

The remainder of this paper is organized as follows. Section 2 is an overview of related work. Section 3 introduces the Connector vision. A Connector prototype system comprising a typical scenario, employed perceptual components and frontend devices, as well as the underlying architecture, is presented in Section 4. A summary and conclusions are given in Section 5.

2. PREVIOUS WORK

There has been a significant amount of effort expended towards understanding user context to improve technology-mediated communication.

Both "Awarenex" [4] and "Live Addressbook" [3] are systems in which users can see others' location and availability status with an interface similar to today's instant messaging buddy lists. Users can consider this information in order to make smarter decisions about contacting others. The "Live Contacts" system [6] also provides preferences for communication channels. "Enhanced Telephony" [5] is a desktop-based design of an enhanced PC-phone. In all of these systems, users must either manually update their availability state, or context information is inferred automatically from sources such as login time, personal calendars,

ICMI'05, October 4–6, 2005, Trento, Italy.

messenger status, idle time of computer input devices and engagement in communication activities.

All of these systems provide the contactor with information about potential contactees. SenSay [7] is a mobile phone that follows a different approach. It adapts to changing user states by manipulating ringer volume, vibration, and phone alerts for incoming calls. SenSay uses a number of wearable sensors including accelerometers, light and microphones mounted on the user's body to provide context information.

Extensive context-aware and pervasive computing environments have been developed at a number of research institutes, e.g. Xerox Park's ParcTab system [15] and Carnegie Mellon's Aura project [8].

The Connector service presented in this paper is an approach that combines many of the features mentioned above. It utilizes a variety of sensors, such as cameras and microphones built into specially equipped rooms or mobile devices, to acquire context information. Upon attempting contact, Connector aims to provide users with relevant context information about potential contactees. Furthermore, Connector adapts the behavior of the contactee's device automatically in order to avoid inappropriate interruptions. Finally, Connector uses a number of output devices to deliver information to users in the most unobtrusive way possible.

This paper outlines the Connector vision and introduces a prototype system integrating all the perceptual resources and addressing all aspects of the communication problem: including frontend devices, context modeling, and the sensory infrastructure for the smartroom and mobile scenarios.

3. CONNECTOR VISION

Most cell phone owners have received unwanted, interrupting phone calls in inconvenient situations. On the face of it, we would like these calls to be rejected automatically. But what if a rejected phone call was of unexpected importance? What if the call is closely related to the user's current activity? From the contactor's point of view, how many missed contact attempts must be made before success? How much time must be lost leaving messages, listening to messages, attempting to locate the contactee, or communicating over low-bandwidth channels, before the value of the communication is exceeded by the cost of getting in touch?

Figure 1. Most cellphone users have experienced inappropriate phone calls, e.g. in a meeting situation.

The Connector service aims to solve these and similar problems by facilitating context-aware communication for mobile users. It attempts to intelligently connect people at the best time, using the best media. To accomplish this goal, the Connector must maintain an awareness of its owners' activities, preoccupations, and social relationships.

Such a service is not possible without a detailed understanding of user context. The service must know who is doing what with whom, why and where in a given space to consider appropriate actions. To answer these questions, we must continuously track human activities using all perception modalities available and afterwards combine the perceived multimodal information to provide pertinent assistance. To reduce the complexity of the task, we focus on office environments and in particular meeting situations. For this reason, a smart meeting room is equipped with a variety of sensors including cameras and microphones. Perceptual components include computer vision and audio processing techniques to provide information about the current situation in the smartroom and the users' identities, activities and locations. To capture the context of mobile users outside the smartroom, wearable audio sensors are integrated into mobile clients. A similar setup can be applied to different environments or common social spaces, such as living rooms or theatres.

In order to connect users in the most appropriate and unobtrusive way, Connector must use many frontend devices to interact with the user. If mobile phones and notebook or desktop computers are not available or inappropriate for communication, the Connector could instead use visual displays projected onto some convenient surface or directed audio.

The Connector architecture includes a proactive personal agent for every user, which acts as a digital secretary. The personal agent controls and manages all interaction with its user. It knows about the user's current location, availability, personal preferences and social relationships to potential contactors; thus it can find the best time and medium to reach its user. Since all logic is located on the server, users are able to communicate from any network-connected client device. Clients with displays, like mobile phones or PCs, provide an adequate graphical interface. Devices like conventional telephones offer a speech interface which allows communication between the user and the personal agent

Personal agents in the Connector environment must be able to process different types of communication requests. Besides standard person-to-person connections, a user will be able to specify a topic for the request. In this way, a contactee could specify that the Connector block all incoming communication unless the call is related to the current task. In addition, Connector will allow users to establish time-dependent connection links (e.g. "I want to talk to my boss before the next meeting") and will find an appropriate time for both partners to establish the connection. Group connections will deal with multiparty conversation or broadcast of information (e.g. "Inform all meeting participants that I will be late"). One-of-many connections will allow the user to convey information to at least one person in a group; rather than requiring serial attempts to reach group members, the Connector can automatically select the most-available group. For all types of connections, Connector will find the best communication medium, using whatever devices are the most appropriate and convenient.

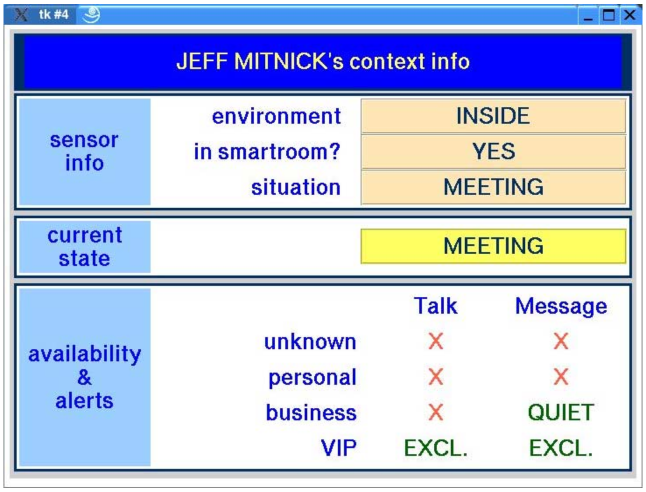
4. CONNECTOR PROTOTYPE

In this section we present our current prototype implementation of the Connector service. The focus of this prototype was the integration of all perceptual resources and communicating front- and backend modules. We first present a typical use case, and then introduce the perceptual components, frontend devices, and system architecture.

4.1 A typical scenario

Our main user, Jeff, realizes that he will be late for a meeting. He uses his mobile phone to inform the other meeting participants of the delay. His personal agent knows that most meeting participants are already in the meeting room and notifies them all together via the smartroom's communication interface. The message "Jeff will be ten minutes late" is projected onto an empty wall and an audible cue notifies the participants that new information has appeared on the display.

A co-worker, Bob, wants to contact Jeff. Jeff's personal agent discerns that Jeff is currently driving and therefore only available for business telephone calls. Bob's phone tells him that he can either place a call immediately or leave a voice message. As Bob chooses to call Jeff, the connection is established and Jeff's phone rings at high volume so that he can hear it over the ambient car noise.



JEFF MITNICK's context info

sensor info	environment	INSIDE	
	in smartroom?	YES	
	situation	MEETING	
current state		MEETING	
availability & alerts		Talk	Message
	unknown	X	X
	personal	X	X
	business	X	QUIET
	VIP	EXCL.	EXCL.

Figure 2. A visualization of Jeff's context information as stored in his personal agent. Jeff is currently in a meeting. He is only available for communication with VIP contacts and accepts messages from business contacts.

As Jeff arrives at the meeting, his personal agent is informed that Jeff is now in the smartroom where a meeting takes place. Jeff's personal agent has learned that Jeff does not take phone calls or messages from personal or unknown contacts when he is in a meeting (see Figure 2). When Bill, a friend, tries to reach Jeff, he is notified that Jeff is currently engaged in a meeting that is scheduled for another half hour. Bill asks to be called back later. Also during the meeting, Joe, who happens to be one of Jeff's VIP contacts, wants to talk to Jeff. Joe is informed that Jeff is in a meeting, but he decides to place the call anyway. Jeff's personal agent notifies him about the incoming call via targeted audio; in this way Joe's communication request does not disturb the other meeting participants. Jeff decides the call is important enough to take it, and he quietly leaves the meeting to speak with Joe. Finally, after the meeting, Jeff's personal agent proposes that he call Bill friend back.

4.2 Proactive agents and data model

The Connector system consists of proactive personal agents which act as digital secretaries for their users. The personal agents store user data (i.e. address books, personal settings, and pending calls), context information and available frontend devices. They are thus always able to determine the best mode of interaction with their users.

Context information currently includes the address book with contact information, user preferences and settings and perceptual information provided by three perceptual components. The first detects whether or not the user is in the smart room. The second detects whether or not a meeting is taking place in the smart room given that the user is in the smart room. The third detects gross user environment (currently “inside,” “outside,” or “in vehicle”) given that the user is not in the smart room. The outputs of these sensors can be fused to imply the current user state: unknown, in vehicle, outside, inside but not in the smart room, in the smart room, or attending a meeting in the smart room.

We define four interaction levels, divided into two broad interaction classes. “Available for talk” and “available for instant message” are synchronous interaction classes; “available for email” and “available for voice mail” are asynchronous interaction classes.

Contacts in the address book are divided into four classes: unknown, personal contacts, business contacts, and V.I.P. contacts.

For example, consider the context information summary for user Jeff in Figure 2. Jeff does not want to accept any communication from unknown or personal contacts if he is in a meeting, but he wants business contacts to be able to send text messages. V.I.P. contacts are allowed to send text messages or initiate voice contact, but Jeff wants notification of V.I.P. contacts to be sent privately. Jeff is always available for asynchronous communication, not shown in Figure 2.

The Connector may supply contactors with information about the contactee’s current state, depending on the relationship between contactor and contactee. In the scenario outlined in Section 4.1, for instance, Jeff’s V.I.P. contact is told that Jeff is in a meeting, but that he can still initiate voice contact if he wishes. A personal contact will be given neither state information nor the option to initiate voice contact.

The Connector can also manage schedule information and pending contact requests. If a contactee becomes available, the contactee’s personal agent communicates with other personal agents to find out if any of the pending contacts are available, in which case the connection is established. The personal agents can also schedule meetings using this information.

The smart meeting room also has an interface agent, which communicates in the same way with the personal agents.

We use distributed, persistent object storage to ensure data consistency among the personal agents and the various frontend devices. The core object store tracks client subscription to system objects (i.e. address book entries, messages, emails or calendar entries) and distributes changes and updates appropriately.

This model, though simple, is scalable and will be made more detailed in the future.

4.3 Frontend devices and multimodal interfaces

Communicating with users in appropriate and unobtrusive ways is a key feature of the Connector service. Our prototype implementation includes many frontend devices with multimodal interfaces to deliver this feature.

A smart phone with graphical and speech interfaces is the main mobile device. In addition, our smart meeting room is equipped with interaction devices intended for use when the user is in the smart room.

4.3.1 The Connector smart phone

Connector's mobile frontend client is a smart phone. We control incoming calls, outgoing calls, messages, and phone alerts.

The smart phone graphical interface, shown in Figure 3, offers Connector-specific features such as message broadcasting. This interface runs on the Sony Ericcson P900 Symbian phone (using direct frame buffer access), Windows CE devices (using the Simple Direct Media Library), Linux devices, and Macintosh OSX devices. Three generations of smart phones were developed and are shown in Figure 3.

The smart phones contact the user's personal agent to determine how to respond to incoming calls or messages. This communication takes place via Wifi or TCP over GPRS. It can seamlessly respond to the presence of unreliable links from either network, and switch protocols when necessary.

Figure 3. Three generations of smart CHIL phones running the Connector graphical interface: Sony Ericsson P900, HP iPAQ h6315 and i-mate PDA2K GSM/GPRS Pocket PC.

4.3.2 Automatic speech recognition (ASR) on a mobile device

In order to provide a natural user interface to the Connector, we implemented speaker independent automatic speech recognition (ASR) on our mobile devices. ASR is used to issue commands to the device, including dialing and sending short predefined messages. The ASR system is a PDA-optimized port of our the Janus Recognition Toolkit (JRTk) [10,11]. It is a three-state sub-phonetically tied semi-continuous recognizer composed of 2000 distributions over 2000 codebooks. Each codebook is a 16-component Gaussian mixture. The feature space consists of 32 linear discriminant analysis (LDA) features computed from 32 Mel Frequency Cepstral Coefficients (MFCCs). The system has a 250-word vocabulary and uses a semantic context-free grammar (CFG) for search. This configuration allows roughly real-time decoding on modern PDAs. In the future, the ASR system will be expanded to cover more commands, and to cover text message dictation as well.

Figure 4. Targeted audio device (left) and a camera-projector pair (right) mounted on motorized computer-controlled pan-tilt units are used as interaction devices in the smart meeting room.

4.3.3 Steerable camera- projector and targeted audio

A steerable camera-projector device (SCP, see [16]) is installed in the smart room and makes it possible to use any convenient horizontal or vertical surface as a display. The SCP consists of a camera and projector mounted on a motorized, computer-controlled pan-tilt unit (see Figure 4). Associating a camera with the projector creates an actuator-sensor pair which combines projection and vision capabilities. This association makes our display surfaces interactive and allows for personalized dialog with users in the smart room, even if they are not carrying any electronic devices. We currently concentrate on projecting simple "yes/no" dialog options to users and tracking users' hands to detect dialog choices (see Figure 5).

Since the messages are meant to be displayed in readable form to any user in the room, they can be rotated and rectified to appear undistorted at the projection's location. After rectification, the interaction buttons do not appear at a fixed position in the camera image; hence, their position has to be determined at interaction time. Position tracking is done by color histogram-matching using pre-trained color models. Once the dialog button is found, foreground-background segmentation allows us to detect when a button region has been occluded and a dialog option has been selected.

A relatively new technology, targeted audio, allows sound to be focused in a narrow beam in such a way that it is only audible within the path of that beam. This technology is useful for user-specific audio notification. Our targeted audio device consists of several directed audio speakers [1] mounted on a motorized, computer-controlled pan-tilt unit.

The commercially available pan-tilt units [12] carry payloads of up to 9 pounds with speeds of over 300º per second, and offer precise dynamic position control.

---

[1] The targeted audio device is developed by Daimler-Chrylser AG, Group dialogue Systems, a CHIL partner.

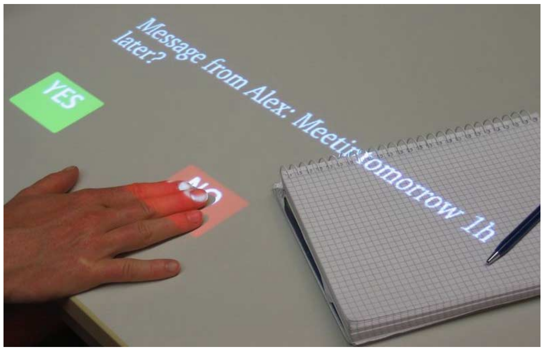


Figure 5. The camera associated with the projector makes it possible to carry out simple dialog interactions.

4.4 Perceptual components

To acquire user context, we focus on automatic detection rather than manually-operated context switches. In our prototype implementation of the Connector service we use face recognition, the detection of meetings, and acoustic environment detection.

Within the larger CHIL framework, many more perceptual components, including audio-visual person tracking, person identification, and speech recognition, are used to model users and describe user context. These modules will be integrated into the Connector service as they become available. In the following sections, descriptions are given for modules which have already been integrated into the Connector.

4.4.1 Face recognition

A face recognition/verification module, using a pan tilt zoom camera pointed at the smart room door, is used to detect and recognize meeting participants as they enter the smart room. This module first checks whether the person entering the room is in the list of participants; if so, the person is identified. The face recognition module is designed to optimize robustness over a small scale face verification task (i.e. one with approximately ten persons). Face recognition proceeds in two stages. The first stage contains specialized face and eye detectors for the people it should identify. This stage reduces the number of full identification attempts; although everyone entering the room is a potential meeting participant, the system should only recognize the actual meeting participants and ignore others. By using specialized detectors for the people in the list of known users, the detection rate of people not on the list decreases. The second stage contains the full recognition procedure. A feature vector is derived from the input face image using a local appearance based face representation method proposed in [9]. Afterwards, it is compared with the feature vectors in the database. If the minimum distance between the test feature vector and the ones in the database is higher than a threshold value, the face is ignored. Otherwise, the system hypothesizes the identity of the person whose stored appearance model matches the input features most closely.

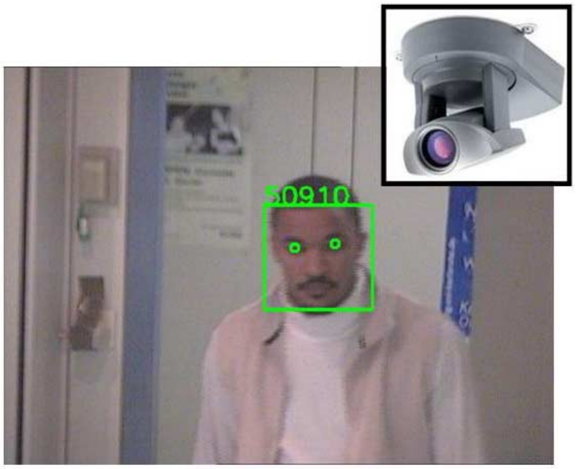


Figure 6. The face recognition module runs on images captured from a pan-tilt-zoom camera facing the door to the smart room.

4.4.2 Meeting detection

Detecting a meeting could imply tracking all users in the smart room, characterizing, their interactions, analyzing their speech, and any number of other complex perceptual procedures. For the prototype we describe here, a much simpler vision-based approach is used. A ceiling-mounted wide angle camera is used to detect activity in a specified area of interest around the meeting table (see Figure 7). Activity detection is done by adaptive background modeling using Gaussian mixture densities, as described in [13]. It is assumed that most of the foreground segments are caused by moving participants, with adaptation making sure that displaced objects such as chairs or notebooks are slowly integrated into the background. If a certain amount of activity is observed at the table over a fixed period of time, the occurrence of a meeting is hypothesized.

4.4.3 Acoustic environment detection

To identify the context of mobile users, we analyze acoustic signals captured by a user-mounted microphone. We use these signals to identify the user's current environment. Environment is an important context cue, as it can strongly bias the distribution of possible user activities. In [2], we detailed an approach capable of distinguishing eleven different environments with error rates less than 20%. We have since modified this system in two different ways.

The first modification to the system described in [2] was to expand the global coverage of the system while reducing the number of classes to a set that is both more easily separable by human listeners and is more useful to the target Connector user. That is, [2] described a system for distinguishing environments in one city. We have since undertaken a worldwide data collection effort, resulting in a corpus of environmental recordings from eight cities on four continents. This corpus was collected using a

Sony minidisk recorder and Sony ECM-717 stereo microphone. The classes covered in this corpus include airport, gallery (loosely any non-travel-purposed large indoor space), park (any outdoor, non-urban setting), restaurant, street, train platform (any space including train tracks), train station (not including the platforms), and vehicle (including automobiles, trains, buses, and airplanes). To distinguish between these classes, we record audio at 16kHz with 16-bit depth, and extract audio features at a rate of 100 frames per second. We model each class with mixtures of Gaussians (GMMs). We used one GMM per class with the number of Gaussians per mixture either optimized per class using the Bayesian Information Criterion (BIC) or set to a fixed number across all classes. We trained the GMMs on data from a balanced training corpus containing 26,880 one-second segments (3,360 per class) and tested the system on a balanced testing corpus containing 6,720 one-second segments (840 per class). Depending on the feature set used, error rates varied from 48.63% (a perceptually-motivated feature set) to 26.87% (Mel-frequency cepstral coefficients, or MFCCs). For the Connector application, we needed a more robust, if less informative notion of user environmental context. Hence, we conducted another experiment in which we limited the class set to three: inside, outside, and vehicle. Collapsing the confusion matrix from the best-performing eight-class system to a three-class system results in an error rate (using MFCCs) of 16.5%.

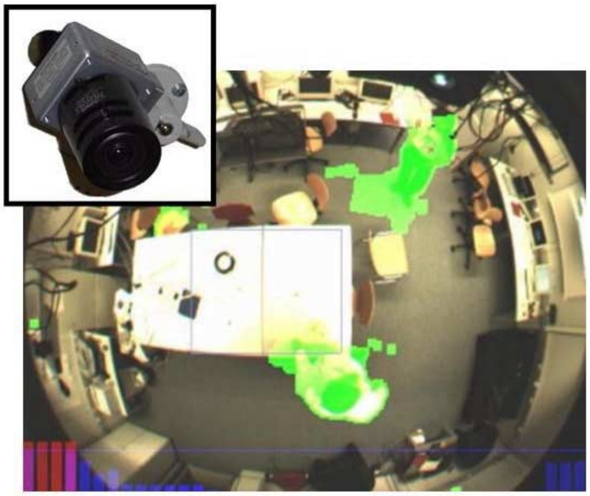


Figure 7. A perceptual component capable of recognizing meeting situations runs foreground segmentation on images from a ceiling mounted wide angle camera.

The second modification to the system described in [2] was to build a restricted prototype capable of running on a small portable device like the Connector phone. Using the same PDA-optimized engine as our ASR system, described above, we collected a new, pilot corpus on the PDA using the device's hands-free earbud and microphone. This data was sampled at 22kHz with 16-bit depth. We collected data from the three required classes --- inside, outside, and vehicle --- in one locale. We derived from this corpus a balanced training set of 3,000 one-second examples, a balanced test set of 600 one-second examples, and a balanced held-out set of 447 one-second examples. Using techniques similar to those for the system above, our best performance was a 6.6% error rate on the test corpus and an 8.0% error rate on the heldout corpus. This system used features based on Principal Component Analysis (PCA) on 5-frame windows of MFCCs.

4.5 Prototype architecture

In our smart room, multimodal data streams from various sensors have to be accessed by the perceptual components. These components analyze video or audio data and produce context events, which are passed to the Connector module. The Connector module contains all high level decision logic and employs proactive user-level and smart room-level agents, which choose appropriate frontend devices to interact with the users (see Figure 8).

In such a complex environment, where multiple perceptual components, actuators, filters and reasoning components, running on a distributed network, have to interact in an online fashion, a modular and dynamically reconfigurable architecture is of great importance. In our implementation, the problem was addressed at two levels: low-level high bandwidth data streaming and high-level message passing and control.

For the former, the NIST Smart Flow System middleware [14] is employed. Smart Flow allows for fast transfer of high quantities of sensory data over a network of components or clients running on several real machines. Each client produces a uniquely named output flow, which can be accessed by one or more receivers over the network or on the same machine. The dynamic relay of the data flows is handled by a series of servers that manage socket communications for distant receivers and shared memory accesses for local ones. All that has to be known by a new client to be integrated dynamically into a federation of running clients is the name and type of data flow it wishes to access. This makes the system extensible, quickly reconfigurable and easy to use.

At the higher level, a socket-based message passing scheme developed in our lab is used for sending messages and events between perceptual components, the Connector module and all of its agents. A central message server is responsible for registering all interacting modules and redirecting messages at runtime to the appropriate recipients according to a predefined rule base. The rule base is implemented as a finite-state machine; each state accepts a different set of messages and triggers different actions, including state-switching. Messages can be triggered and sent automatically by components which detect events, or manually. All of the implementation details for a given system are contained in the rule base; this allows for complete separation between application semantics and message-passing logic. The central server collects all messages as they are sent, checks the rule base to see if they are defined as triggers for new messages, and then generates new messages or redistributes the incoming messages (possibly changing message parameters) as necessary. Messages can be triggered automatically or by user interaction to switch between different states with different message passing logic, changing the network behavior to better cope with the new situations. Besides flexibility and modularity, another advantage of this centralized logic, decentralized semantics approach is fault-tolerance, as attempts to pass messages to nonexistent or nonfunctional components cause the server to take no action.

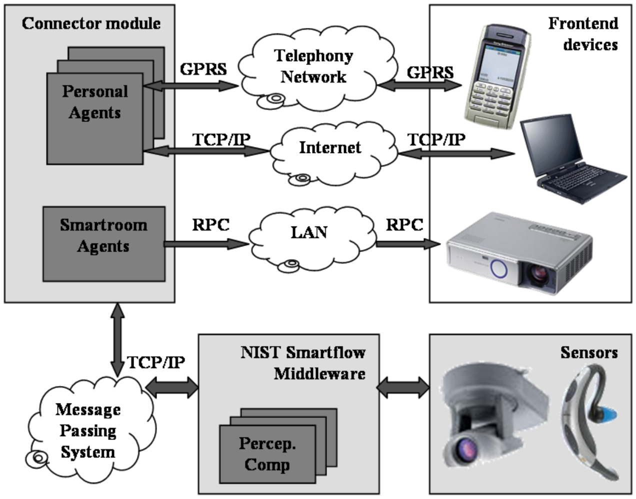


Figure 8. Connector server-side components (gray) communicate with client devices via their associated networks.

5. SUMMARY AND CONCLUSIONS

We have presented the Connector service, a system which uses state-of-the-art multimodal and perceptual technologies to acquire user context information and uses this information to provide supportive, non-burdensome communication protocols to establish intelligent human-to-human connections. We have introduced a prototype system in the form of a first Connector infrastructure including sensors mounted in a smart room and on a mobile device, a set of perceptual components, and several frontend devices. The proposed system is being extended to handle richer context models and user interfaces. Learning user preferences and behavior patterns, combined with more sophisticated fusion of context information from multimodal perceptual components will allow more detailed and accurate scene analysis and improved communication choices.

6. ACKNOWLEDGMENTS

This work has partially been funded by the European Commission under Project CHIL (http://chil.server.de, contract #506909).

7. REFERENCES

[1] http://chil.server.de/

[2] R. Malkin, A. Waibel, Classifying user environment for mobile applications using linear autoencoding of ambient audio, *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, Mar 2005.

[3] A. E. Milewski, T. M. Smith, Providing Presence Cues to Telephone Users, *Proc. of the Conf. on Computer-Supported Cooperative Work (CSCW)*, Philadelphia, Dec 2000.

[4] J.C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li and J. Bhalodia, ConNexus to Awarenex: Extending awareness to mobile users. In *Proc. of the Conf. on Computer Human Interaction (CHI)*, New York, 2001.

[5] J.J. Cadiz, A. Narin, G. Jancke, A. Gupta, M. Boyle, Exploring PC-telephone convergence with the enhanced telephony prototype. *Proc. of the Conf. on Computer Human Interaction (CHI)*, New York, USA, 2004.

[6] G. Henri ter Hofte, Raymond A.A. Otte, Hans C.J. Kruse, Martin Snijders, Context-aware communication with Live Contacts, *Proc. of the Conf. on Computer-Supported Cooperative Work (CSCW)*, Chicago, Nov 2004.

[7] D.P. Siewiorek et al., SenSay: A Context-Aware Mobile Phone, *Proc. of the Int. Symposium on Wearable Computers (ISWC)*, White Plains, NY, Oct 2003.

[8] Garlan, D., Siewiorek, D., Smailagic, A., Steenkiste, P., Project Aura: Toward Distraction-Free Pervasive Computing, in *IEEE Pervasive Computing*, April-June 2002.

[9] H.K. Ekenel, R. Stiefelhagen, "Local Appearance Based Face Recognition", Technical report, Universität Karlsruhe (TH), Feb. 2005.

[10] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries and M. Westphal, The Karlsruhe-Verbmobil Speech Recognition Engine, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 83-86, 1997.

[11] H. Soltau and F. Metze and C. Fügen and A. Waibel, A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment, in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2001.

[12] http://www.dperception.com/products.html#PTU-D46-70

[13] Stauffer, C. and Grimson, W.E.L., Adaptive background mixture models for real-time tracking, *Proc. CVPR*, v.2, pp.246-252, June 1999.

[14] http://www.nist.gov/smartspace/toolChest/nsfs/

[15] Bill N. Schilit, Norman Adams, Rich Gold, Michael Tso and Roy Want, The PARCTAB Mobile Computing System, in *Proceedings of the Fourth Workshop on Workstation Operating Systems (WWOS-IV)*, pages 34-39, Napa, CA, October 1993.

[16] G. S. Pingali, C. Pinhanez, R. Kjledsen, A. Levas, M. Podlaseck. User-Following Displays. *IEEE International Conference on Multimedia and Expo (ICME 2002)*, 2002.