Perceptual Collaboration in Neem

P. Barthelmess and C.A. Ellis
Department of Computer Science
University of Colorado at Boulder
Campus Box 430, Boulder, CO 80309-0430, USA
Phone 303 492-8709 Fax 303 492-2448
{barthelm,skip}@colorado.edu

Abstract

The Neem Platform is a research test bed for Project Neem, concerned with the development of socially and culturally aware collaborative systems in a wide range of domains.

In this paper we discuss a novel use of Perceptual Interfaces, applied to group collaboration support. In Neem, the multimodal content of human to human interaction is analyzed and reasoned upon. Applications react to this implicit communication by dynamically adapting their behavior according to the perceived group context. In contrast, Perceptual Interfaces have been traditionally employed to handle explicit (multimodal) commands from users, and are as a rule not concerned with the communication that takes place among humans.

The Neem Platform is a generic (application neutral) component-based evolvable framework that provides functionality that facilitates building such perceptual collaborative applications.

Keywords: distributed multimodal interfaces; group perceptual interfaces; distance collaboration; platforms and tools;

1. Introduction

Project Neem at University of Colorado is concerned with the development of socially and culturally aware collaborative systems. The project explores issues surrounding such systems in a variety of different domains, targeting different group cultures and interaction styles.

The Neem Platform is a research test bed for this project. It provides a generic (application neutral) evolvable framework upon which socially and culturally aware applications are developed. In the present work, we concentrate on describing the issues surrounding the use of Perceptual Interfaces to support human to human communication in the context of the Neem Platform. The complexity and challenges involved in building applications is only hinted at here (see [9] for a more in depth overview).

Central to the design of the platform is the focus of the project on the dynamic nature of group work in general, and social and cultural aspects in particular, as expressed in multimodal interactions that take place among users as they collaborate using verbal and nonverbal behaviors. Neem applications are designed to take advantage of this rich contextual information to elicit timely and appropriate responses that conform to situated social and cultural norms of a target group.

In contrast to this philosophy, the vast majority of systems that tackle multiple modalities are geared towards handling direct commands, and mostly ignore eventual interaction among the users. Neem’s opportunistic approach to human interaction is more closely related to the work presented e.g. by: Jebara et al [13] in which the system acts as a meeting mediator, offering feedback and relevant questions to stimulate further conversation; Isbister et al [11], whose prototype mimics a party host, trying to find a safe common topic for guests whose conversation has lagged; Nishimoto et al [17] whose agent enhances the creative aspects of the conversations by entering them as an equal participant with the human participants and keeping the conversation lively; CMU’s Janus project [24, 12] is somewhat related, in its aim to make human-to-human communication across language barriers easier through multi-lingualtranslation of multiparty conversations and access of databases to automatically provide additional information (such as train schedules or city maps) [24]. While Neem shares the interest in human-to-human mediation, its goals are more ambitious than keeping a bi-party conversation going. Neem targets social and cultural aspects and is therefore concerned with a more detailed view of how groups work, and how collaborative systems can contribute.

The Neem platform is a software infrastructure that supports and incentives application designs that are based on
dynamic, context-appropriate system reaction, situated to specific groups of users. The platform is designed to ease the integration of functionality for capture and reification of user actions over a variety of modalities (e.g., speech, text, gestures), as well as analysis and reasoning based on this reified context. Also considered are facilities that allow applications to present themselves as a set of virtual participants, perceived as having personalities and opinions, through the use of multiple output modalities such as animated characters that emote, voice, natural language textual output as well as conventional widget-based communication.

The goal of the platform is to facilitate the development of real-time distributed multipoint perceptual-based applications that focus on context-appropriate dynamic reactions. To this end, an evolvable component-based infrastructure is provided. Functionality embedded in the platform makes development of group-aware perceptual applications roughly equivalent to the development of singleware (single user applications). Development is also supported by embedded support for Wizard of Oz experiments, that we extend to explore issues on group interaction.

The platform incorporates reusable components that provide extensible support for basic real-time collaboration, such as session management, consistency control, data distribution, communication and coordination among components. Its focus is on functionality that allows the actions of a group of interacting users to be captured, reasoned upon, and reacted to, under a perceptual perspective that explores multiple input and output modalities.

From a Perceptual Interface research perspective, Neem differs from typical work because of its focus on implicit interactions of a group of users, as opposed to the focus on single users, as is more typical. Even in systems that target groups of users (e.g., [6, 15, 19]), the focus is on explicit multimodal command, in which speech and pen, for instance, are used to replace more conventional interface devices.

In the rest of this paper, we describe in further detail the functionality of the Neem Platform. We begin by presenting Perceptual Interfaces in the context of group collaboration (Section 2). The architecture of the platform is presented in Section 3. Section 4 describes the prototype that has been implemented. The paper ends with Summary and Future Work (Section 5).

2. Perceptual Interfaces for group collaboration

Perceptual Interfaces are based on a paradigmatic shift from the structured, command-based GUI interfaces to a more natural one based on how humans interact among themselves. These new kinds of interfaces have been extensively studied (e.g., by [21, 20]). Findings support the notion that provided that an interface mimics real life, even if imperfectly, principles that explain perception in real life can be applied straightforwardly to computers, i.e., people’s reactions to computers are fundamentally social and perceptual [21]. The reaction to animated characters, for instance, tend to be similar to real participants, and even gender, ethnicity and similar factors play similar roles independently of the obvious artificial nature of such characters, and their imperfections in movements, and voice.

The use of such a paradigm is particularly relevant and appropriate in the context of group collaboration. The bulk of group communication is already performed by humans among themselves. The objective of a group system is in fact to support such human to human interaction. It is therefore natural to employ a similar communications-based paradigm to integrate augmentation functionality in a seamless and transparent way. It is thus desirable on one hand for a system to extract information from an ongoing interaction among humans, rather than by direct commands given through conventional GUIs, and on the other hand to present system’s contributions through similar mechanisms as the ones employed by human participants, i.e., through a complex combination of speech, gestures, facial expressions, gaze, etc. that comprise virtual participants.

Neem differs from Perceptual-based applications such as the Microsoft Paper Clip and similar agents that target a one-to-one interaction. While the Paper Clip interacts directly in a one-to-one fashion with users that are most of the time trying to get some “heads-down” work done and might resent the systems intrusion [23], in Neem the environment in which the system interjects is already communication-based and an application just adds a few extra (virtual) participants that try to blend into the group by following to the best of their possibilities the groups social norms.

Because most of the time participants are engaged in conversations among themselves, applications are not required (or even allowed!) to interject often, and might therefore wait for occasions in which higher confidence on perception and reaction are perceived. In contrast, direct human-computer dialogs require prompt and precise responses and may quickly degenerate as cues are misinterpreted and an application loses track of the topic of conversation, which can be upsetting in a Perceptual-based interaction, due to the high (human-like) expectations that might be associated with anthropomorphized systems.

Figure 1 depicts information flow in Neem’s Perceptual Interface. Participants interact in a distributed collaboration environment and their actions are sensed and interpreted; multiple interpreted streams are combined during fusion. Reasoning analyzes the events in context, i.e., it takes into consideration past interaction. A response is generated (which includes doing nothing), fission determines
the most effective way to react given available modes and taking into consideration user characteristics. This potentially complex sequence of multiple modality actions are finally rendered at one or more participants stations (reaction).

![Perceptual information processing in Neem](image)

Fusion and fission are mechanisms traditionally associated with multiple modality systems. Fusion and fission offer a potential to liberate users from having to adapt to system mandated input and output mechanisms. Users can be free to choose the modality that best fits their styles both when producing information as well as when being presented with information produced by a system. There is therefore a potential for adaptation to user needs and styles.

Briefly, **Fusion** is the process that combines individual modality streams into a single one, based on time and discourse constraints [14]. Fusion unleashes the full power of multimodal communication, allowing information on each mode to complement each other, for example, combining speech recognition with gesture analysis to allow users to point to objects and locations and issue verbal commands to operate on them, e.g. in the pioneering system Put-that-there [3]. If fusion combines different modes, **fission** does exactly the opposite (e.g. [7]). Given a message that needs to be conveyed, it is possible and desirable to pause and consider what is the most effective way of conveying it, either through some conventional user interface mechanism, or through a voice message, or through the use of an animated character that emotes [4].

Notice that what is sensed and reacted to are not direct commands from users to an application as is most common, but rather the content of ongoing interactions among human participants themselves, in the form e.g. of their spoken and textual exchanges and widget activations.

Neem applications may for instance “listen” to participants voice exchanges to extract contextual knowledge. Repeated references to voting, for instance, might prompt the system to initiate a dialog to determine if participants want to start a voting tool, by having a virtual participant, represented as an animated character, talk to the group, or to selected members. And in case of perceived agreement, a voting tool might be remotely launched on each of the participants stations. Summarized results might then be presented to a meeting chair or to the whole group, either through a conventional interface window or through a virtual participant.

When participants have been working for a long time, and their video images match a pattern that might correspond to tired facial expressions, a Neem application might suggest that participants take a break, or adjourn the session to some other time. Participants react to that either by agreeing, in which case an application might arrange for the next meeting to be scheduled, or by rejecting the suggestion.

Neem thus explores the potential of Perceptual Interfaces in the context of collaborative system for extraction, analysis and reasoning about human-to-human interactions in support of social and cultural awareness.

Availability of human to human interaction information is also used in the context of Wizard of Oz experiments, which form an integral part of the platform. Wizard of Oz is a traditional technique for testing new features in the field by having a human participant masquerade as a virtual one, thus allowing for faster development cycles than possible if everything needs to be coded.

This technique is usually employed by natural language based systems, and has been in use at least since the first speech recognition projects were attempted (e.g. during development of Hearsay [10]) even if perhaps under different names.

Here we adopt and extend it to support experiments in group collaboration. Our use differs from typical experiments, which as a rule involve a single user at a time, to whose commands the (human) wizards react. Our use scenario involves a group of users, and the reaction to implicit communication that is directed from one user to one or more other users, in the course of a meeting. Wizards have been explored in multimodal settings (e.g. in [8, 22, 4]), but again in the context of a single user (even if multiple wizards can share the load of reacting to a single user’s commands).

Wizard of Oz functionality is quite naturally integrated in Neem applications through the use of available multimodal contextual information that is extracted by platform components, and by the already available means to display information through animated characters and voiced output.
The implementation of such functionality consists of a display that lists events of interest and allows interface elements to be remotely controlled by a human operator.

3. Architecture

We now turn our attention to the platform’s architecture. In this paper we just briefly overview Neem’s architecture which is presented in more detail in [2]. Details of Neem’s coordination model can be found in [1]. The platform’s design is based on a 2-tiered approach:

1. The platform’s foundation is a generic coordination model that can in principle be used to implement just about any distributed component-based system. It is at this core level that flexibility and extensibility are introduced into the platform.

2. A second stage specializes this generic infrastructure to support collaboration and dynamic reaction based on a perceptual paradigm, as is the goal of the project.

We now discuss in turn each of these three layers in further detail.

3.1. Coordination model

In the coordination model adopted by Neem, decoupled components interact indirectly through message exchanges. A meta-level mechanism mediates access to a Linda-like tuple-space [5]. This functionality, that we call mediation, allows for explicit control over component cooperation through message-rewrite rules.

A tuple-space is a shared dataspace, through which cooperating processes communicate among themselves only indirectly, by posting or broadcasting information into the medium, and by retrieving information by removing or just copying information off the medium. Retrieval is based on content-addressing capabilities of the medium [18].

Messages generated by the components and channeled through mediators can be seen as events, some of which are interpreted as requests for service that will cause one or more components to be in turn activated thus establishing indirect communication between the component that generated the original message and the one(s) that got activated as a result of mediator internal actions. Components react to messages by performing some computation and eventually sending out one or more messages.

Changing communication aspects therefore implies exclusively in modifying mediator definitions to “rewire” component communication patterns, without having to touch any of the existing components. The writers of the components are in fact largely unaware of the actual contextual use of their components beyond required handling of messages that are received and responses that are generated. The ability to compose existing behaviors in unanticipated ways results in clear flexibility in their potential for reuse.

Components are uniformly employed both for the development of core platform functionality and application layers. The only difference from a developer’s point of view is the potential for reusability of components at each of these levels - while platform components are expected to have general use, application specific components are tied to a specific solution and therefore have less chance of reuse. In the following paragraphs, we describe mostly generic functionality, that therefore corresponds to reusable platform elements, rather than application level ones.

3.2. Framework

The coordination model we just examined, while flexible, offers no specific support for the Perceptual Interface based dynamic collaborative applications that are targeted by Project Neem. We now briefly examine how the generic foundation is specialized to offer necessary services.

Given Neem’s focus on real-time distributed multipoint perceptual interface-based applications, the framework embeds the following functionality: Multimodal input and output - extraction and presentation of multiple modality information; Multimodal processing - such as fusion, fission, parsing of natural language streams; Multimedia processing - audio mixing and video switching; Session management - creation, joining, leaving, meetings or sessions.

Application development support covers the areas: User interface components - for building shared artifacts; Support for reasoning - for implementing dynamic context-based reaction; Support for Wizard of Oz experiments - to allow experimentation without coding.

These issues are addressed by specializing components into interface and augmentation components. Even though conceptually similar (both are message enabled component types), Neem Interface Components (NICs) are characterized by their attachment to one or more interface devices, which makes them suitable for collecting and relaying interface events generated by each participant, in the form of standard messages. A Neem Augmentation Component (NAC), on the other hand, does not have this constraint and is purely a message processing device.

NICs provide means for the integration of multimedia devices, such as conventional monitor, keyboard, mouse, consoles, audio and video. Other less conventional devices (e.g. Virtual Reality (VR) goggles, haptik devices) can also be integrate through NICs. All that is required to integrate a new device is a set of NICs that interface with a device, extract events commanded by users and modify its state (for devices with output capabilities) according to commands re-
ceived as messages. A NIC may for instance attach to an audio source (e.g. microphone) and do speech-to-text conversion, or extraction of prosodic features, or attach to a video source and do gesture or facial expression extraction. NICs also react to messages they receive, causing changes to the associated state of the interface, for instance rendering at users stations of textual messages, graphics or full animations including gesture and/or voice.

Neem Augmentation Components (NACs) provide mostly back-end functionality, i.e., they are mostly responsible for processing the multiple modality streams, e.g. parsing natural language streams, fusion, fission of different streams and so on, as well as providing session management and multimedia processing capabilities.

NACs are also responsible for providing support for reasoning about the perceived context of an ongoing interaction and generating appropriate responses. Responses themselves are dependent on the specific application that is built on top of the platform.

4. Prototype

A prototype of the platform has been implemented. Actual development leverages as much as possible on existing, field tested technology, based on open standards (Figure 2).

A messaging infrastructure is implemented as two distinct environments - a collaboration and a multi-agent environment, currently realized by functionality offered by a DC-MeetingServer MCU and the DARPA Communicator architecture. These two environments are connected through a coupler component. The distributed collaboration environment provides support for participants’ interaction through NICs and the multi-agent environment supports back end augmentation functionality, such as multimodal processing and reasoning, which are typically provided by NACs.

DC-MeetingServer is a commercial H.323 Multipoint Conference Unit (MCU), produced by Data Connection Limited. H.323 is a family of multimedia conferencing protocols published by the International Telecommunication Union (ITU).

DARPA Communicator is based on MIT's Galaxy architecture [16]. It is an open source hub-and-spoke architecture that provides a distributed, scriptable message passing system with special emphasis on building language-enabled dialogue systems.

Coupler is the component that binds these two distinct environments together - it translates between message formats and is responsible for: 1) relaying collaboration events to the multi-agent environment for analysis and 2) propagating messages originated at the multi-agent environment among those components whose signatures comply to the messages. Coupler code accounts for the semantic differences between the conceptual architecture, based on tuple-spaces and actual push-based implementation.

About ten thousand lines of code (mainly C/C++) implement the connection, translation between environments, as well as a highly abstrscted API that is used to develop application layer components.

Multimodal support in this initial phase consists of console i/o (monitor, keyboard, mouse) as well as natural language through typed and spoken messages. Natural language text output and animation, including voice production can be employed as output modalities, besides the activation of conventional widgets. Natural language processing capabilities running on the back-end are provided by language processing modules produced by Colorado University’s Center for Spoken Language Research (CSLR) under the CU Communicator Project [25]. The open source CU Communicator system, is a DARPA Hub compliant system.

User interface components are developed in Visual Basic, Java and C++. Speech-to-text is built on top of Microsoft’s SAPI (Speech API). A variety of speech-to-text engines are compliant with SAPI. We currently employ IBM’s ViaVoice 9.0’s engine. Animation is currently built using Haptek’s VirtualFriends. Preliminary vision-based analysis is built on top of Intel’s OpenCV library and Entropic’s Hidden Markov Model Toolkit (HTK3).

White board, file transfer and application sharing, audio and video communication are provided directly by the H.323 infrastructure functionality.

A simple proof-of-concept application has been successfully developed on top of the platform. For details, please see [2].
5. Summary and future work

The Neem Platform is a generic framework for the development of collaborative applications. It supports rapid development of augmented real-time distributed multipoint group applications that employ a perceptual interface paradigm.

A prototype of the platform has been implemented. Actual development leverages as much as possible on existing, field tested technology, such as H.323 MCUs, the DARPA Communicator architecture and commercial speech and animation engines.

Future work will enhance the platform's capabilities by expanding its multimodal functionality, including facial analysis, gestures, including American Sign Language capabilities in a robust way.

Different applications are being developed and deal with the challenging aspects of building culturally and socially adequate tools and virtual participants. Planned applications include business meetings and distance education applications.

References