

Mobile Conversational Characters

Mohammed Waleed Kadous and Claude Sammut

Smart Internet Technology CRC
School of Computer Science and Engineering
University of New South Wales
{waleed, claude}@cse.unsw.edu.au

Abstract

InCA is a distributed personal assistant conversational character. The front-end runs on a handheld PDA and uses facial animation and natural speech input/output to interact with the user to provide services such as appointments, e-mail and weather reports. Existing conversational character research focuses on desktop platforms, but there are obvious differences when the platform is a mobile device, the two most obvious being the limited computational power and the restrictions on input modalities. This paper discusses the architecture and implementation of InCA, which addresses these two challenges.

1 Introduction

Most conversational characters are designed to run on desktop computers. The user is assumed to have several modes of input, such as keyboard, mouse and voice.

However, recent years have seen an explosion of mobile devices, such as personal digital assistants, in-car computers and high-powered mobile phones. Techniques for conversational characters on such devices are under-explored. There are two particular challenges:

- Limited computational power. In particular, these devices do not have hardware acceleration of 3D graphics, and are not likely to in the near future.
- Limited I/O options. These devices may be small, have low resolution, lack keyboards etc.

A further problem, shared with desktop characters, is making these characters seem intelligent. Due to the limited computational power, this is even harder on a mobile platform.

While Moore's law will inevitably increase the computing power of mobile devices, they will be roughly three iterations of Moore's law behind (i.e. typically one quarter to one eighth the computing power of) desktop machines. Therefore, using the network to provide additional computing power to mobile devices will allow, in certain circumstances, a circumvention of their limited computing power.

InCA (Internet-based Conversational Agent) is a mobile conversational character that runs on a PDA, but uses network infrastructure to overcome some of the above limitations. It is part of the program of research being undertaken by the Smart Internet Technology Cooperative Research Centre. A photograph of InCA running on a mobile device is shown in Figure 1. The current implementation of InCA has the following features:

- Provides the following personal assistant-type services: news headlines, email-reading, making and listing appointments (synchronised with the desktop), retrieving weather and exchange rates, and translations

from English to several European language (albeit badly pronounced in English).

- Spoken (but speaker-dependent) natural language input. Users can say things like "Can you get me the weather for today, please?" Or "Yo InCA! What's the exchange rate for US dollars, man?" or "I want the local news, InCA." Our system does not force them to adhere to a constrained grammar.
- Speech output with facial animation, but currently without emotional expression.



Figure 1: InCA running on a PDA

The rest of this paper will discuss the architecture used by InCA to provide these capabilities; in particular, it will focus on the two most interesting problems: dialog management and facial animations. It will then discuss some recent refinements, before presenting plans for future work.

1.1 Related work

Our work draws on the established study of embodied conversational agents. This includes the work of Cassell et al (1999) on REA, and also Cyberella (Gebhard, 2001). Both of these systems try to develop virtual characters that interact via speech and gesture.

What was said	Speech recognition
ok what about my appointments	that care about my point man
what's the weather going to be like	what the weather down to be light
uh francs please	a Frank's place
ok can you translate i'm tired to german	a cake can you translate I'm tied to German
no goodbye	know the by.

Table 1: Speech vs recognised words

The InCA project is also related to the work on TRIPS (Allen et al., 2001) and the CU communicator system (Pelom et al., 2000). Both of these projects focus on the process of collaborative interaction through speech.

The underlying technologies for mobile conversational characters are also currently developing; for example, work on facial animation for mobile platforms (Pandzic, 2002). There has also been some work on speech-based interaction with mobile devices of a limited form (Speereo, 2002). There is also the SmartKom Project at DFKI, which also aims to develop mobile conversational characters (Wahlster, 2002); however, at the time of publication, details are unavailable. To our knowledge, this is the first published work on functional mobile conversational characters.

2 InCA Architecture

To provide the above features, InCA employs the architecture shown in Figure 2. It operates within three domains: the client, which runs on a PDA, the server that coordinates the speech recognition, speech synthesis and dialog management, and finally a coordinator that is responsible for real-time retrieval of data such as weather, appointments, and so on from the Internet.

2.1 Client

The InCA client currently runs on a Compaq iPaq H3870¹. This device has the following specifications.

- StrongARM 206MHz processor.
- 32MB Flash ROM, 64MB RAM.
- 320x240 65,000 colour screen.
- Internal Microphone/Speaker.
- Linux operating system with Qt/Embedded GUI.
- 802.11b Wireless ethernet (WiFi).

The StrongARM processor is designed for low-power consumption, and not computing power – it consumes no more than 400 milliwatts – two orders of magnitude less than a desktop processor. It does not have a floating-point unit. Obviously, its 3D capabilities are extremely limited.

The software that runs on the client is very “thin”; it streams audio (hopefully the speaker’s voice) to the server and plays back audio and facial animation scripts once they have been downloaded to the client. To simplify detecting silence, a button on the side of the device – usually used as the “voice record” button – is used to signal when the user is speaking to InCA. It communicates to the server over WiFi. Audio is currently transferred uncompressed (22kHz 16-bit PCM), since this occupies less than 5 per cent of the maximum available bandwidth.

¹It also works with other StrongARM-based Linux devices, e.g. the Sharp Zaurus SL-5500.

2.2 Server

The server coordinates several different components. It currently runs on a Linux workstation (Pentium III 800MHz, 256MB RAM). Firstly, it takes the audio coming from the client and reassembles it into a continuous audio stream (since the audio is streamed to the server, but it only goes to the speech recognition engine, delays in communication can be easily handled). It sends this data to the speech recognition engine. Currently, we are using IBM ViaVoice to provide speech recognition. It takes the the speech recognition engine’s guess of the utterance and passes this to the dialog manager, which generates a reply. The InCA server then takes the reply and passes it to the Text-to-Speech (TTS) engine to generate both the audio and facial animation instructions. Currently, we are using IBM ViaVoice TTS for this, however we are evaluating other alternatives, such as Rhetorical’s rVoice, and ScanSoft’s RealSpeak. This information is conveyed back to the InCA client, and once the data is downloaded, the InCA client is told to begin playing the response.

Local TTS and speech recognition systems were evaluated, but did not have the required features, nor were they computationally viable. For instance, CMU’s Flite TTS system (Lenzo and Black, 2002) takes up 90 per cent of CPU while synthesising; leaving little processor time for animation; and with somewhat inadequate speech quality. Similarly, although there are mobile speech recognition engines, these require strictly specified grammars and do not even offer an unconstrained dictation mode. IBM’s ViaVoice, by comparison, allows natural language, enabling users to utter unconstrained requests, such as “Can you translate *where’s the nearest restaurant* into Spanish?”

2.3 Dialog management and Coordinator

Most dialog management systems are designed for text-based interaction. The approaches required for dialog management with speech are significantly different, mainly because of the unreliability of speech recognition. Table 1 shows user utterances and the speech recognition engine’s best guess of that utterance.

For this reason, we employed Probot (Sammut, 2001) for dialog management. Probot has a number of features that make it well-suited to this problem. Probot is implemented as a rule-based system embedded in a Prolog interpreter. The rules consist of patterns and responses, where each pattern is matched against the user’s utterance and the response is an output sentence. Both patterns and responses may have attached Prolog expressions that act as constraints in the patterns and can invoke some action when used in the response.

The pattern-response rules are grouped into contexts, which represents the current topic of conversation. A typical script consists of several contexts. Changing contexts

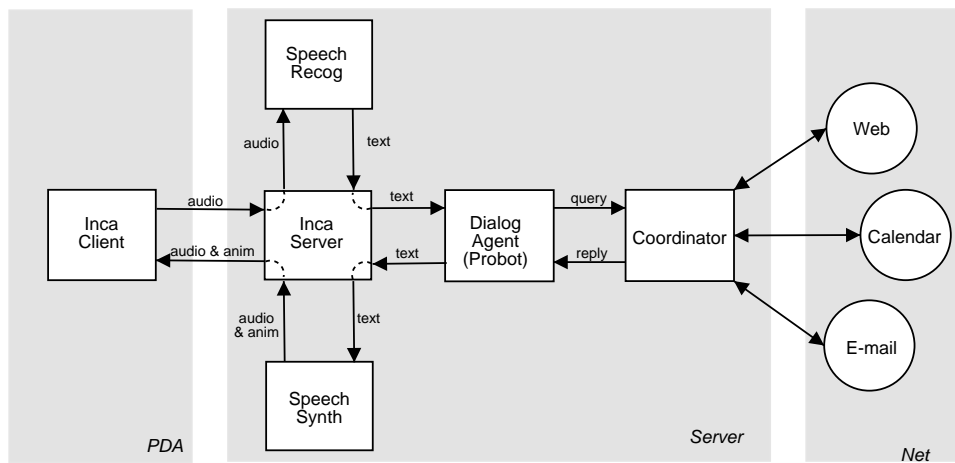


Figure 2: InCA architecture

is managed by making a Prolog call. To add further flexibility and conversational agility, Probot also employs two other lists of pattern-response rules: filters and backups.

Filters are used to detect utterances that require an instantaneous change of context. Utterances are first checked against filter patterns before being checked against patterns in the current context. Backups are used to handle utterances that did not cause any filter or context rules to fire.

This makes InCA very agile in conversation, while still allowing it to use context to constrain the recognition problem.

```
InCA: Hello. How can I help you?
User: I want the local news.
InCA: Headlines are: [...]
User: can you help me a bit?
InCA: I can tell you local, international
      or computer news.
User: well can you list my emails then?
InCA: You have 3 messages. [...]
User: help me.
InCA: To read a message, just tell me the message
      number. Or you can ask me to list your messages.
```

In the above conversation, InCA is able to offer context-specific help, while still being able to change context from news to e-mail in a single statement – no explicit indicators such as “can we do e-mail now?” are required.

One particular set of Prolog commands that is extensively used in our scripts is to retrieve information from dynamic information sources, such as the web, e-mail and calendaring systems. This is accomplished through the Coordinator; a program which accepts instructions from the Probot and retrieves the information from the network.

The current implementation of the Coordinator is a Perl script. It uses RSS (Rich Site Summary) to retrieve headlines; SOAP (Simple Object Access Protocol) to make the remote procedure calls to retrieve exchange rates and translations (through Babelfish) and POP3 to retrieve e-mail.

These queries can form part of InCA’s replies, as demonstrated below. An example of a rule employed by our system is:

```
c_language :: { french | german | spanish | italian };
* translat~ * {into | to } <c_language> ==>
[
  ^coord_query([translate, ^4, ^2])
]
```

The above rule would fire on an utterance such as “could you please translate where is the nearest hotel into italian”. The response generates a coordinator query asking to translate the second expression matched (in this case, “where is the nearest hotel”) into the fourth expression matched (“italian”).

2.4 Facial animation

Clearly, for current generation mobile platforms, three-dimensional texture-mapped heads are out of the question, at least if any other processing is to be done. One possible approach would be to use a 2D polygonal face, as Perlin (1997) does; however, for many mobile devices, even this small number of calculations may be excessive.

For the current version of InCA, we have used a very simple, “cartoon” approach. A face was generated using the commercial character animation package Poser. In addition to the face, we also manipulated the face to generate the mouth positions described in Nitchie (1979) as conveyed in Parke et al (1996). In total, 18 mouth positions are generated. Each phoneme produced can be mapped to one of these mouth positions. Rather than the whole image being retained, only a rectangle including the mouth is kept.

When the TTS generates the audio for InCA, it also generates the phonemes and the corresponding timing information. This can be used to construct a “playlist” of which mouth position should be shown and for how long. The playlist and the audio are both transmitted to the client. Once both are received by the client, synchronised playback begins.

When playback begins, the whole face is drawn. When it is time to change the mouth position, the mouth area is “overwritten” by the mouth image corresponding to the phoneme.

A similar idea is used to implement random blinking.

3 Refinements

In addition to the basic plan above, we have applied several refinements.

3.1 Alternatives for speech

We evaluated the accuracy of our system by having several conversations, and recording correct, confused and wrong responses. A “confused” response occurs when none of the

Probot rules fire; and consequently InCA asks the user to repeat his last statement. A “wrong” response occurs when an incorrect rule fires, e.g. the user says: “List my e-mails”, and InCA replies with “One Australian dollar is 0.55 US dollars”, which obviously did not meet the user’s request.

We found that with a sample of 99 utterances, from a user who had trained the speech recognition system, InCA was wrong only 2 per cent of the time, and confused 31 per cent of the time.

IBM ViaVoice, however, has a capability of producing alternative “interpretations” of an utterance – typically as many as 16 alternatives are generated. If there are no patterns in the current context that match an utterance, the InCA server requests an alternative from the speech recognition engine, and tests that to see if any patterns were matched this time. This is repeated until all alternatives are exhausted, at which point the user is asked to repeat his or her utterance. For example, consider the third example from Table 1: “uh francs please”. The first guess, “a Frank’s place” doesn’t match any patterns, so a second alternative is requested. The second alternative is “francs Place”, which – while not totally correct – is still recognised because the context is currently exchange rates, and one of the patterns picked up “francs” as a currency. Using this technique the confusion rate was reduced to 22 per cent – a 29 per cent reduction in the number of times the user was asked to repeat themselves.

3.2 Multiple characters

It is relatively easy to generate new characters to be used with InCA – another set of 22 or so images must be generated, and the speech synthesis must be modified to generate a different voice. We have generated several interchangeable characters; in particular we have a male character as well. These characters are interchangeable with minimal effort.

3.3 Facial gestures as state indicators

Some of the InCA queries can take a few seconds to perform; since retrieval of information over the Internet is sometimes slow. The conventional way to convey this to the user might be to get InCA to say “Please wait”. However, we have a facial expression that involves raising the eyebrows and looking up in a manner that is associated with thinking or contemplating. This facial expression is maintained until just before InCA is ready to speak. At this point, normal eye contact is restored. Exploring such use of facial expressions to express states such as thinking, confusion and so on is something we plan to explore.

4 Further work

Obviously, InCA is in her infancy, and the avenues of research are many. Our plans for further work include:

- Evaluating how important the face is; would the device be equally useful without the face?
- Evaluating the “3D cartoon” face against a real-time 2D face similar to Perlin’s (1997).
- Adding a phone interface to InCA, so that instead of interacting via a PDA, the interaction could occur using a standard phone line.
- Learning user’s preferences.

- Integrating more tightly between the speech recognition engine and the dialogue management system. In particular, the current model of taking alternatives from the speech recognition engine after recognition has been done is simplistic. We are looking at techniques for extracting probable words in the current context and using them to inform the search performed by the speech recognition engine.

- Adding multimodal capabilities to InCA, so it could, for example, display maps.

5 Conclusion

InCA is a mobile conversational character that uses speech I/O and addresses some of the unique challenges of the mobile environment. Simple facial animation techniques may be adequate; we are hoping to test this statistically. Further, the network can be used as a means to obtain additional computing power to effectively add features such as speech recognition and synthesis.

6 Web page

Movies, photographs, conversation transcripts, etc are available from:

<http://www.cse.unsw.edu.au/~inca/>

References

- James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–37.
- J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjalmsón, and H. Yan. 1999. Embodiment in conversational interfaces: Rea. In *Proceedings of the CHI’99 Conference*, pages pp. 520–527.
- P. Gebhard. 2001. Enhancing embodied intelligent agents with affective user modelling. In *UM2001: Proceedings of the Eighth International Conference*, Berlin. Springer.
- Kevin A. Lenzo and Alan W. Black. 2002. <http://www.speech.cs.cmu.edu/flite/>.
- E. B. Nitchie. 1979. *How to Read Lips for Fun and Profit*. Hawthorne Books, New York.
- Igor S. Pandzic. 2002. Facial animation framework for the web and mobile platforms. In *Web3D 2002*, pages pp. 27–34. ACM Press.
- Frederic I. Parke and Keith Waters. 1996. *Computer Facial Animation*. A K Peters.
- B. Pellom, W. Ward, and S. Pradhan. 2000. The CU Communicator: An architecture for dialogue systems. In *International Conference on Spoken Language Processing*, Beijing China.
- Ken Perlin. 1997. Layered compositing of facial expressions. In *SIGGRAPH 1997 Technical Sketch*.
- Claude Sammut. 2001. Managing context in a conversation agent. *Electronic Transactions on Artificial Intelligence*, 6(27). <http://www.ep.liu.se/ea/cis/2001/027/>.
- Speereo. 2002. Speereo web site. <http://www.speereo.com/>.
- Wolfgang Wahlster. 2002. Multimodal interfaces to mobile webservices. In *ICT Congress, Den Haag*, <http://smartkom.dfki.de/>.