Parallel Processing of Interruptions and Feedback in Companions Affective Dialogue System

Jaakko Hakulinen\textsuperscript{1}, Markku Turunen\textsuperscript{1}, Raúl Santos de la Camara\textsuperscript{2}, Nigel Crook\textsuperscript{3}

\textsuperscript{1} Department of Computer Sciences, University of Tampere, Finland
\textsuperscript{2} Telefónica I+D, Spain
\textsuperscript{3} Computing Laboratory, University of Oxford, UK
{jaakko.hakulinen|markku.turunen}@cs.uta.fi, e.rsai@tid.es, nigc@comlab.ox.ac.uk

Abstract

Much interest has recently been given to making dialogue systems more natural by implementing more flexible software solutions, such as parallel and incremental processing. In the How-Was-Your-Day prototype, parallel processing paths provide complementary information and the parallel processing loops enable the system to respond to user activity in a more flexible manner than traditional pipeline processing. While most of the components work as though they were in a pipeline, the Interruption Manager is a component which uses the available information to generate the system responses outside of the pipeline and handles situations such as user interruptions.

Index Terms: spoken dialogue system, system architecture, incremental processing, multimodal dialogue

1. Introduction

During recent years there has been a lot of interest in more flexible and efficient processing models in dialogue systems. Incremental processing and other more flexible architecture solutions have been implemented in various systems. The incremental processing approach is not new in spoken dialogue systems; Hearsay-II system [1] already featured fundamentally incremental processing, enabled by the blackboard architecture introduced by the work. However, soon after, when speech and language technologies matured into self-contained components like ASR, TTS and NLU, a couple of decades of mostly pipeline based systems followed.

The main motivation behind the current interest in more flexible dialogue processing is to enable more natural and efficient [2] interaction than the strictly turn-based style more or less forced by pipeline architectures. Another motivation is the diversification of input processing components. For example, emotion analysis has provided parallel information to ASR. Similarly on output side, many dialogue systems include embodied conversational agents capable of facial, hand and body gestures which must be controlled continuously. Use of reactive GUIs, has also been successful [3]. Finally, multi-core CPUs have made parallel processing available on common hardware.

The basic components (ASR, NLU, TTS etc.) are still present in modern systems but their relations and communication is more flexible than a simple pipeline. There are various approaches to this, which have been devised in different combinations in various dialogue systems. One approach is to make the entire processing chain incremental so that initial results are sent forwards as soon as possible and better hypotheses can replace them when more evidence becomes available. The challenge in such cases is that each component must be able to re-plan its actions when the hypothesis changes. One incremental processing solution has been implemented by Skantzze and Schlangen [4]. In their system, components can $update$, $purge$ and $commit$. $Update$ is necessary when more information becomes available. When previous information is revoked, results based on the revoked information must be $purged$ from the system. $Purging$ may propagate through several components. To limit the amount of stored information, a component can $commit$ to a result, i.e., inform that the result will not be revoked.

Another way to move forward from the pipeline model is to expand the system to include parallel processing in some part of a pipeline or more thorough parallel processing of all the components. Examples of parallelism introduced into pipeline structures include use of multiple speech recognisers to process the same input. One may also process speech in parallel with different goals such as the detection of emotion. Multimodal systems have this sort of parallelism because of the need to process the different modalities, like speech and gestures, simultaneously. When parallelism is taken further, all the components can work in parallel. One example of such an approach was taken with the TRIPS architecture [5].

When parallelism is introduced, synchronisation becomes a challenge at some point. In TRIPS system, the synchronisation was implemented by including a context assumption in each processing request to make sure each message is processed correctly. If the assumed context information is not yet available to the component (it is being generated by a parallel processing path), a component knows to wait for the context to become available. Such a solution spreads the control over parallel processing to each component so that they must be aware of related issues. An alternative approach is to introduce an overseeing controller component, which takes care of overall synchronisation. Risk with such approach is that the controller can become very complex.

In this paper, we present the How-Was-Your-Day prototype, a multimodal dialogue system, which can react to speech in a timely manner, provide constant feedback via an animated character and handle user interruptions. From the software architecture point of view, this is implemented using a message passing communication framework, parallel processing paths on selected parts of the system and an overseeing component called the Interruption Manager, which controls other components to provide timely feedback and handle interruptions but which does not control the normal processing pipeline. In the rest of the paper we first describe the How-Was-Your-Day demonstrator and the ‘social’ dialogue it has with users. The system description is followed by the explanation of the system architecture. The paper ends with a more theoretical analysis of the architecture.

2. How-Was-Your-Day demonstrator

In the Companions project, the goal is study new type of interaction, where we move from task based interaction to a
long term relationship between the dialogue system and a user. In the "How-Was-Your-Day" scenario we have been working with ‘social’ conversations, where emotions and the communication itself are key. The How-Was-Your-Day prototype (HWYD) listens to the user’s description of their working day. The system can ask questions to elicit more information from user and occasionally attempts to positively influence user’s emotional state with longer tirades. Both the user’s descriptions and the tirades can be rather long (around 50 words). Naturally, users are likely to interrupt some of the system’s tirades, so the interaction is rather far from traditional task-based dialogue consisting of short, strict dialogue turns. Rather, the dialogue implemented in HWYD prototype is a particularly good example of interaction, where the timely responses to the user’s emotional speech and interruptions are vital for satisfying interaction.

The following example dialogue illustrates some of the interesting features of the system including tirades, user interruption and emotional analysis of user input.

**User:** Worst day ever. I arrived late because of the traffic.

**System:** You have my sympathy. What happened next?

**User:** I missed the meeting on project responsibilities; I got more work dumped on me. Just what I don’t need. I’m already likely to miss the deadline. [ECA looks worried when negative tone is detected in user’s voice]

**System:** I’m sorry to hear that. Who was at the meeting?

**User:** Only my team. They gave me a set of PowerPoint slides from the meeting. I spent the rest of the day reading them but my computer kept crashing because of the updates.

**System:** Oh dear! It is understandable how you must be feeling. It’s right to be unhappy about such a bad situation. Try not to worry about missing the deadline because of the technical problems. [User starts speaking] It’s sad that your motivation could be affected by them. [System interrupts its tirade]

**User:** Well, it’s not the technical problems that worry me, but the amount of extra work on this project and the deadline.

**System:** What is the project about?

The system appears to users as an Embodied Conversational Agent capable of facial and body gestures. Emotional speech can be found both on TTS, which includes, e.g., paralinguistic elements, user input, which goes through emotional classification. ASR results are also analysed to detect sentiment on the linguistic level.

The system is designed to be used in the home environment with a wireless microphone for input and a TV screen as display. Figure 1 is a screenshot of the ECA and textual and graphical visualisation of the system’s internal activity for demonstration purposes (for a video demonstration, see http://www.companions-project.org/demonstrators/english/).

### 3. System Architecture / Processing Model

HWYD prototype consist of 15 different modules connected by Inamode, a hub based message passing platform. The modules and their abbreviations are:

- **ASR:** Dragon Naturally Speaking is used for real time, large vocabulary speech recognition. Per user acoustic training is used to maximise recognition rates.
- **AA:** Acoustic Analysis module provides low level analysis of user speech for ATT module.
- **ATT:** Acoustic Turn-Taking module detects start and end of user speech and user interruptions. It also separates back-channeling from interruptions.
- **AEC:** Acoustic Emotion Classifier uses EmoVoice [6] to categorise users’ speech segments as neutral, negative-active, negative-passive, positive-active, or positive-passive.
- **DAT:** Dialogue Act Tagger and Segmenter module compiles ASR results and when the ATT reports an end of user turn, segments the input into semantic units and applies dialogue act tagging on them.
- **SA:** Sentiment Analyser module uses the AFFECTiS service to mark each word in and the entire segment of recognised user input as neutral, positive or negative valence.
- **EM:** Emotional model fuses the emotional information from Acoustic Emotion Classifier and Sentiment Analyser into one category per user utterance.
- **UM:** User model contains systems knowledge of the user.
- **DM:** Dialogue manager takes care of general dialogue management such as deciding what to ask of the user and whether to attempt to influence the user’s emotional state.
- **ASM:** Affective Strategy Module generates a strategy and resulting plan to influence user’s emotional state.
- **NLG:** Natural Language Generator generates tirades based on the plans generated by Affective Strategy Module.
- **MFM:** Multimodal Fission Manager controls the ECA component according to the requests it receives from the other components and sends messages so that other components know when the system is speaking.
- **TTS:** Text-To-Speech by Loquendo includes emotional and social expressions implemented for the prototype.
- **ECA:** Embodied Conversation Agent component is based on Haptek player and features a lifelike female character with facial and body gestures and speech output.
- **IM:** Interruption Manager manages the system to make it more responsive by sending messages to MFM while the system is processing user input, e.g., and at the moment the emotional tone of user input has been detected, and by controlling different components on interruptions situations.

![Figure 1: HWYD demonstration interface](image)

**Figure 1:** HWYD demonstration interface

![Figure 2: Data flow in HWYD prototype](image)

**Figure 2:** Data flow in HWYD prototype.
The components communicate via the Inamode platform which provides hubs to pass messages to connected modules over a network. Multiple hubs can be used to optimise the number of messages passed around in larger systems.

For message passing, an XML based enveloped has been specified. It contains metadata about the message including a unique id, sender module id, a timestamp and turn id, when relevant. Metadata also includes a “why_id” field, which is a list of previous messages, that summarise reasons for the generation of this message.

**Figure 3: IM model of system status.**

### 3.1. Processing paths

The HWYD prototype implements parallel processing paths and uses the Interruption Manager to control other components in situations where regular processing procedure must be altered. In addition, there are two processing “loops” from user input to system output; Main Dialogue Loop and Feedback and Interruption Loop. The loops are not optional; instead they both operate on each user input. The feedback and interruption loop generates immediate reactions to user activity and includes acoustic analysis components (ASR, AA, ATT, AEC), Interruption Manager and output Components (MFM, TTS, ECA). The main dialogue loop, which is the normal processing path, includes all the components in the system except Interruption Manager. Except for Interruption Manager, the two loops are not explicit to components, the same input and output messages are used in both. These two loops will be described in more detail in the next section.

#### 3.1.1. Main Dialogue Loop

The Main Processing Loop, which is depicted with the black arrows in Figure 2, starts with parallel processing of the audio signals. Three components process the audio stream independently: ASR takes care of speech recognition, the Acoustic Emotion Classifier classifies audio segments into one of five emotion categories and the Acoustic Turn-Taking module analyses the audio to detect the user’s speech and separates barge-in from back-channel. The Acoustic Turn-Taking module also uses ASR output as an additional indication of voice activity. Each of the audio components segments the audio themselves, which leads to possibly three different segmentations. These segmentations are combined in the components, which fuse the results.

The outputs of the ASR and the Acoustic Turn-Taking module are fused by the Dialogue Act Tagger and segmenter. ASR provides results incrementally, i.e., partial utterance hypotheses are available. The full utterance hypotheses may also be of varying duration and may not cover the entire user turn. Since the HWYD domain includes long user turns, the ASR results must be combined and re-segmented. The DAT collects the ASR results, compiles them according to end of turn messages from Acoustic Turn-Taking and then segments them into semantic segments. Finally, the Dialogue Act Tagger and segmenter adds the tagging to the segments.

The Acoustic Emotion Classifier results are first fused with the ASR results by the Emotional Models module. The Emotional Model also receives text tagged by Sentiment Analyser and fuses the two sources into one emotional state, combining the different segmentations generated by the two processing paths.

The result of Emotional Model is integrated with the dialogue act tagged and segmented ASR results in Dialogue Manager and processed by natural language understanding component. At this point all the information has been fused. This is then used to plan the system activity. This planning can occur in the Dialogue Manager alone, when system has a question to ask, or it can involve Affective Strategy Module, when the system plans longer tirades for the user. These two processing routes are optional paths taken on different dialogue turns.

The output from the Natural Language Generator is processed by the Multimodal Fusion Manager, which controls the ECA. This component continuously runs in parallel with the rest of the system to keep the ECA alive. MFM also informs Acoustic Turn-Taking whenever the system starts or stops speaking.

#### 3.1.2. Feedback in Interruption Loop

The system includes the feedback and Interruption loop alongside the main dialogue loop, to make the system more reactive and natural in its interaction. The loop uses information that has not been fully processed to make the ECA react in a timelier manner and control the system during user interruptions etc. Unlike the main dialogue loop, which works in an strictly turn based manner, the feedback and interruption loop can react to user activity any time.

At the heart of the loop is the Interruption Manager. The Interruption Manager receives information from the Acoustic Turn-Taking module so that it knows when the user is speaking and it sends messages to MFM to make ECA adopt a
listening pose. Other ECA reactions include such things as emotional gestures when a strong emotion is detected in the user’s speech and short, immediate responses after the user has stopped speaking and before the main dialogue loop has provided the full system response. In the example dialogue in chapter 2, such output includes phrases like “Oh dear!”

The Interruption Manager can also control the Dialogue Manager and the Affective Strategy Module, as signified by grey arrows in Figure 2, since those components may need to re-plan their actions when the user interrupts a long system tirade. In such cases MFM can tell what part of the output was not spoken out and Interruption Manager then asks the Dialogue Manager and the Affective Strategy Module to re-plan accordingly. The IM tracks the processing of the user’s utterances through the various modules using a System State Model (SSM - Figure 3) which is a two-level Finite State Machine. On the lower level, the SSM models the state of each module in terms of whether it is idle (Id), processing a message (Pr) or waiting for an incoming messages to complete its processing (Pe). On the upper level, the SSM tracks whether each agent in the conversation (system and user) is speaking (Sp), silent (Si), or in the case of the user, interrupting (In). The Interruption Manager also includes timers to act when no messages are being sent. For more details, see Crook et al. [7].

To summarise, the separation of the two loops provides both a conceptual model and a technical solution to separate the traditional pipeline processing and the processes required for timely reactions to user speech. It is noteworthy that from the point of view of individual components these different processing paths are invisible. They process input according to their specifications and the same output can be used in both loops.

4. Discussion

The How-Was-Your-Day dialogue system implements support for phenomena such as barge-in, emotional speech, back-channelling and multimodal feedback via an ECA during user turns by including parallel processing loops and an Interaction Manager to coordinate the system overall. This approach can be analysed in various ways. A generic model for describing dialogue systems with incremental processing was proposed by Shlangen and Skantze [8]. The HWYD scenario can be analysed using the concepts of the model.

Shlangen and Skantze define the connectedness feature for modules, meaning that if a module generates messages which are not based on any other messages, then they form an unconnected “island”. In HWYD we have modelled this relation at the InAmode message level using the why_id references within message envelopes. The why_id lists are used primarily for debugging and visualisation of log files, but they do represent the “grounded” relation. In the case of the HWYD prototype, the Interruption Manager is the only component which generates unconnected results: at times, it generates message based on time-out rules and generic system status.

The Completeness of a component refers to the level of completeness of component output compared to component input. If a component generates more complete results than its input is, it must make predictions. None of the HWYD components do such explicit processing, in particular, none of the results are later rejected when found incorrect, which is crucial to the concept in the model. In HWYD, only planned output utterances can be retracted due to user interruptions.

The Update frequency is the relation of the number of consumed and produced messages by a component. A simple case is when a single input message relates to a single output message. This is the case in Sentiment Analyser, Dialogue Manager, Natural Language Generator and Affective Strategy modules. Dialogue Act Tagger and segmenter, Emotional Model and Acoustic Turn-Taking are modules which combine multiple input messages to one output message. The Interruption Manager is once again a special case among the modules. It can generate fewer messages than it receives, which is the case in normal processing situations. In timeout situations etc. it can produce more messages than it receives. Therefore, it does not exactly fit into any of the update frequency categories of Shlangen and Skantze’s work.

In HWYD, support for natural interaction was implemented with a short processing loop alongside a more traditional main dialogue processing loop. Many benefits of the classical pipeline model were retained, e.g., components in most cases remain simple as the same messages are used in both processing loops and components do not need an explicit model of the parallel processes. The only component with any explicit model of concurrency (aside from base architecture) are those responsible for animating graphical agent and Interruption Manager, which contains all the explicit logic related to system level parallel processing.

5. Acknowledgements

This work was funded by the Companions project (www.companions-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

6. References


