

A State Model for the Realization of Visual Perceptive Feedback in SmartKom

Peter Poller, Norbert Reithinger

German Research Center for Artificial Intelligence – DFKI, Saarbrücken, Germany

{poller, reithinger}@dfki.de

Abstract

User acceptance of multimodal dialog systems in general depends not only on the quality of the system output but also significantly on the quality of the process feedback the user gets during interaction. We show an approach that permits the system to present simultaneously - and synchronously to the ongoing processing - adequate detailed and adaptive immediate feedback in various ways. For example, in multimodal dialog systems corresponding graphical presentations and/or appropriate behaviour patterns of an animated agent during user input processing significantly contribute to a more natural and expressively more powerful system. Additionally, in case of processing errors the approach provides sufficient information to be able to present the user detailed descriptive feedback on the stage of processing at which the error occurred. The approach is mainly based on a module working state model that is used to determine in real time general processing state information in terms of relatively fine grained global system working states. Evaluations of the overall system show that simultaneous processing feedback is highly appreciated by users.

1. Introduction

Today's intelligent multimodal dialog systems are moving to cover more and more technically and functionally complex tasks within one system. The multimodal dialog system SmartKom¹ [1] understands imprecise, ambiguous, or incomplete multimodal input and generates coordinated, cohesive, and coherent multimodal presentations for three scenarios, currently addressing more than 50 different functionalities of 14 applications [2].

Unfortunately, this trend is inevitably accompanied by limits of real-time processing. On the other hand, communication pauses are naturally acceptable if the communication partners know what's currently going on, e.g., someone is signaling intensive thinking about his answer to a question. In conjunction with corresponding feedback signals, it is even natural and polite to provide the communication partner with enough time to think about his next utterance. If we transfer this behaviour to human-computer interaction, systems should constantly

provide adequate and adaptive feedback about the system's internal processing state.

In this paper we describe a working state model that can be used to provide this feedback during processing of the user input. The main idea of our approach is that all individual modules permanently publish their respective working state to a central reasoning unit whose task, in turn, is to constantly determine and publish meaningful global working states in parallel to the ongoing user input processing. These global working states can then, in turn, be used to trigger working state dependent system feedback routines of various types using different output modalities (if available) during processing. Flashing lights of different colors or smart progressing bars are simple realization ideas for such feedback. Visual perceptive processing feedback could also consist of meaningful behaviour patterns of an animated agent [3]. Furthermore, the model can easily be extended to infer and provide situation specific information about the stage of processing at which a processing error occurred. The approach only requires that all modules correctly and completely publish their working state (incl. the exact time at which it was reached) in real time.

The set of applicable module working states and global system working states naturally depends on different features of the individual modules and the overall system itself. Basically, a module is either active or inactive. But for example in a distributed system one module might process input data in close cooperation with other modules. Here it becomes relevant to distinguish normal inactivity of a module from expecting intermediate results that are needed to finish the current task.

2. Module working states

In general, in modular systems like those based on Galaxy Communicator [4], or MULTIPLATFORM [5] that consist of many modules, meaningful information about the current processing state of a module is not accessible unless the module itself publishes that information. Therefore, we need a uniform predefined set of module working states that allows each module to identify its current working state and the exact time at which this working state is reached. Of course, every module must constantly publish this information to the reasoning

¹<http://www.smartkom.org>

unit which, in turn, infers and publishes the corresponding global system working state. Since permanent publishing of the same working state generates a lot of redundancy, the modules publish by convention only their module working state as soon as it changes.

First, we characterize the two basic module working states that are relevant for all different kinds of modules:

waiting: The module is currently expecting input data that are needed to start or continue processing or the module is currently performing internal operations (e.g., loading a knowledge source).

processing: The module currently processes user input or input data that were sent by another module.

In addition to these basic module working states, some modules might have more complex functionalities which should, in turn, be reflected appropriately by more fine grained module working states they publish, e.g.:

deactivated: The module has explicitly been deactivated.

expecting: As part of the processing of a task, the module has published the request to perform a subtask and is currently waiting for the publishing of the corresponding response as an intermediate result which is needed for further processing. This state is also used if the module waits for user input that is needed to fulfill a specific task.

Module working state information of this granularity is sufficient within a system in which the end of a turn is equivalent to inactivity of every module. But if there are modules that are capable to perform tasks that are persistent after finishing an interaction with the system, the module working state information must be supplemented. In this case, information about the task that is/was performed and the state that is/was finally reached thereby is also needed in order to complete the state information for the reasoning unit. For example, a video tape controlling module should also publish information about the state of the device it controls, e.g., “playing,” “recording” or “pausing.” In SmartKom there are two more application control modules falling in that category, i.e., a telephony module and a television control module. Again, corresponding device specific state information must be published to the reasoning unit.

3. Global system working states

Based on constant publishing of module working states by the modules themselves, a central reasoning unit determines and publishes a corresponding global system working state. Of course, the set of relevant global working states that is used is system specific because it depends on characteristics of the modules and the system itself.

Our set of global system working states that has shown to be reasonable and sufficient for a multimodal dialog system is as follows:

sleeping: The system is inactive (i.e., there is no user).

idle: The system is awaiting user input.

listening: The system is currently receiving input from the user.

understanding: The system is currently analyzing the meaning/intention of the user input.

understood: The system has comprehended the meaning of the user input and is currently planning an appropriate system reaction/answer.

notUnderstood: The system could not analyze the meaning of the user input.

clarification: The meaning of the user input is unclear. But the system has recognized that and is therefore currently planning a corresponding clarification request.

performing: The system is currently performing actions within applications.

presentationPreparing: The system output presentation is currently being planned.

presentation: The system output presentation is currently being performed.

stranded: The processing of input data (that were initiated by the user or by an application) stopped unexpectedly.

These global system working states form a basic repertoire that covers more or less all potentially relevant global system working states of a multimodal dialog system. Of course, other systems may require other global system working states but the idea behind them remains the same, anyway.

Depending on the applications that can be used within a system and the modes of interaction that are possible with these applications or the system itself, we could imagine an additional layer of global system state information that is used to represent the current “interaction mode” between the user and the system or one of the integrated applications. Corresponding global system states are needed for systems in which the user may also interact with system devices or applications besides the core system functionality. For example, in a multimodal dialog system the user could be mediated to a telephone application to make a phone call via the system. During this call the user does not interact with the system at all – although he might make use of system devices at this time, e.g., a microphone. We identified the following additional global system states to represent different user interaction modes. They are published independently of the global system states defined above.

normal: Normal interaction mode

mediatingActive: The system is currently brokering user input to a third party.

mediatingPassive: The system presents the user an output that does not influence the interaction with the system at all (e.g., television, music, ...)

bargeIn: A user input occurs during the processing of another user input. Of course, this state must be associated with the respective user input stream.

The task of the state reasoning module is to constantly infer and publish in parallel to the ongoing processing the corresponding global system working state from the individual module working state information of every module. It needs a knowledge source that reflects the modular system architecture as well as information about the functionalities of the individual modules. Technically, the knowledge base of our reasoning module is realized as a finite state network, with global working states at its nodes. The transitions depend on logical conditions that are imposed on the working states of the modules. These conditions are expressed by formulas in disjunctive normal form whose predicates are module working states.

4. Visual perceptive feedback in the SmartKom system

All global system states can be used to trigger appropriate and immediate system feedback during input processing. Especially multimodal dialog systems open a large number of facilities to present the system feedback appropriately. There are first the different output modalities themselves that offer various feedback realization options and second, there is the option of combining them elegantly (e.g., a flashing light accompanied by appropriate sound signals). If there is an animated agent, even non-verbal, perceptive and reflective agent behaviour could be triggered in order to visually reflect the current system state in parallel to the ongoing processing. In such systems, the realization of appropriate perceptive feedback may be a real design task. The multimodal presentation module should react meaningfully and immediately to the constantly published global system working states and it needs detailed knowledge how to e.g. match the states to corresponding gestures of an animated agent.

Independently of the overall system in which our approach is realized, the different options to realize perceptive feedbacks during processing (e.g., graphically, acoustically, agent behaviour, etc.) are never intended to address or even interrupt the user directly. They are nothing else but realizations of immediate, non-interruptive feedback signals to the user.

In SmartKom, the user communicates with a life-like character that is realized as an animated agent. Not every global system working state must be communicated

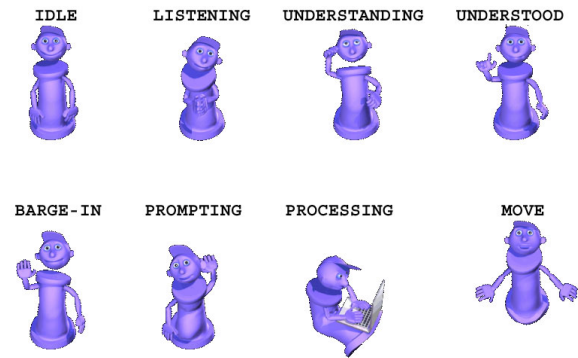


Figure 1: *Some state reflective behaviour patterns of an animated agent*

with a perceptive feedback action. The 8 different agent behaviour patterns presented in figure 1 with their main characteristic posture are those that are important for our system. As soon as the global system working state changes or the system output presentation gets ready to be started, the agent immediately adapts its behaviour by smoothly fading to the newly applicable behaviour pattern. Thereby, homogeneous and natural agent behaviour constantly reflects the current processing state with smooth and fluent state transitions between them. The agent behaviour patterns are assigned as follows:

IDLE: The agent shows breathes regularly and blinks with the eyes.

LISTENING: The agent purposefully turns towards the user in order to give him its best attention while he is, e.g., speaking to the system.

UNDERSTANDING: The agent signals that the system is currently analyzing the input by moving the right hand to the temple and scratching with the index finger.

UNDERSTOOD: The upraised index finger is used to signal that the input was comprehended.

BARGE-IN: The upraised hand is used to show the user that the system is still processing the last input and won't accept barge-in data.

PROMPTING: If there were communication pauses, this gesture is used in the 'idle' state to prompt the user implicitly to continue his interaction with the system.

PROCESSING: The agent types on a laptop to indicate that the system is currently accessing its knowledge sources to fulfill the user's request.

MOVE: This gesture is used as part of the output presentations to enable the agent to move around on the screen.

In addition to system state compatible agent behaviour, multimodal systems can also make use of appropriate graphical output to give perceptive processing feedback. An example is not only to simply indicate the occurrence of an error but also to reason on the stage of processing at which the error occurred. This significantly contributes to a robust system that includes immediate and appropriate system reactions in case of errors. From our experience the following different error categories can be distinguished easily from nothing but synchronously reasoning on the correctly published module working states in a multimodal dialog system:

input errors: Errors that occur because the system did not understand the user input or its intention. These can be further subdivided into: recognition errors due to acoustical conditions and errors occurring during analysis of the user intention

processing errors: Errors that occur during the processing of a correctly interpreted user's intention. These can be subdivided into errors during the processing of the user intention, errors during an interaction with an application, and errors occurring during the presentation of the system output

dialog errors: Errors that occur due to functionality limitations of the system itself, i.e., the user requested a task the system isn't capable of.

Of course, after identification of the above error categories, corresponding error output messages can be triggered in the output presentation module the same way as perceptive feedback is shown during processing. This way, the approach significantly contributes to robustness and user acceptability of the system it is integrated in.

5. SmartKom evaluations and conclusion

The development of SmartKom was constantly accompanied by system evaluations focusing on different viewpoints, including module specific evaluations, ergonomic evaluations and finally end-to-end evaluations [6]. Although there was no specific state reasoning module evaluation, the intermediate evaluations of the overall usability of the system by naive users provided interesting results. The evaluation of the questionnaires filled by each subject shows that especially the non-verbal reflective behaviour was judged as being very intuitive and helpful to understand what's currently going on within the overall system. This can be compared to a former prototype in which the feedback routine wasn't realized yet and the agent behaviour was judged significantly worse there.

To the extent shown above, our model can already be used to robustly keep the user informed even in case of an error. So, future work is the extension of the model in such a way that it naturally becomes possible to inform the user about what the error was caused by and

maybe also a suggestion about how the dialogue could be continued successfully. Consequently, we currently explore first the identification of more detailed, more concrete and more problem centered error categories in order to improve the robust error handling procedure as outlined above. It is an open question how and to what extent the working states must be more fine-grained for that purpose. In this context we are aware of the danger to loose generality in favor of potentially system-dependent refinements. So, we aim to end up with an optimally balanced general working state repertoire that is general enough to be applicable for as much systems as possible without loosing the granularity of the underlying working states that is needed to ensure the realizability of the different benefits of the approach shown in this paper.

6. Acknowledgments

This research was funded partially by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the SmartKom project under grant 01 IL 905 A5 and partially by the European Union in the 5th framework project Comic under grant IST-2001-32311. The responsibility for the content lies with the authors.

7. References

- [1] W. Wahlster (Ed.), *SmartKom - Foundations of Multimodal Dialogue Systems*. Berlin, New York, Tokyo: Springer, 2004, to appear.
- [2] N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pflieger, P. Poller, M. Streit, and V. Tschernomas, "SmartKom - Adaptive and Flexible Multimodal Access to Multiple Applications," in *Proc. of the Fifth International Conference on Multimodal Interfaces (ICMI-PUI 2003)*, Vancouver, Canada, 2003.
- [3] R. Cole, S. van Vuuren, B. Pellom, K. Hacioglu, J. Ma, J. Movellan, S. Schwartz, D. Wade-Stein, W. Ward, and J. Yan, "Perceptive Animated Interfaces: First Steps Toward a New Paradigm for Human Computer Interaction," in *Proc. of the IEEE*, vol. 91, no. 9, September 2003, pp. 1391–1405.
- [4] "Galaxy Communicator Web Page," 2002, <http://communicator.sourceforge.net>.
- [5] G. Herzog, H. Kirchmann, S. Merten, A. Ndiaye, and P. Poller, "MULTIPLATFORM Testbed: An Integration Platform for Multimodal Dialog Systems," in *Proc. of the HLT-NAACL03 Workshop on The Software Engineering and Architecture of Language Technology Systems (SEALTS)*, Edmonton, Canada, 2003.
- [6] F. Schiel, "Evaluation of multimodal dialog systems," in [1].