An Event-Based Conversational System for the Nao Robot*

Ivana Kruijff-Korbayová, Georgios Athanasopoulos, Aryel Beck, Piero Cosi, Heriberto Cuayáhuitl, Tomas Dekens, Valentin Enescu, Antoine Hiolle, Bernd Kiefer, Hichem Sahli, Marc Schröder, Giacomo Sommavilla, Fabio Tesser and Werner Verhelst

1 Introduction

Conversational systems play an important role in scenarios without a keyboard, e.g., talking to a robot. Communication in human-robot interaction (HRI) ultimately involves a combination of verbal and non-verbal inputs and outputs. HRI systems must process verbal and non-verbal observations and execute verbal and non-verbal actions in parallel, to interpret and produce synchronized behaviours. The development of such systems involves the integration of potentially many components and ensuring a complex interaction and synchronization between them. Most work in spoken dialogue system development uses pipeline architectures. Some exceptions are [2, 3], which execute system components in parallel (weakly-coupled or tightly-coupled architectures). The latter are more promising for building adaptive systems, which is one of the goals of contemporary research systems.

In this apaper we present an event-based approach for integrating a conversational HRI system. This approach has been instantiated using the Urbi middleware [4] on a Nao robot, used as a testbed for investigating child-robot interaction in the

Heriberto Cuayáhuitl, Bernd Kiefer, Ivana Kruijff-Korbayová, Marc Schröder
DFKI GmbH, Language Technology Lab, Saarbrücken, Germany

Piero Cosi, Giacomo Sommavilla, Fabio Tesser
Istituto di Scienze e Tecnologie della Cognizione, ISTC, C.N.R., Italy

Georgios Athanasopoulos, Tomas Dekens, Valentin Enescu, Hichem Sahli, Werner Verhelst
IBBT, Vrije Universiteit Brussel, Dept. ETRO-DSSP, Belgium

Aryel Beck, Antoine Hiolle
Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire, U.K.

Contact: e-mail: ivana.kruijff@dfki.de

* Parts of the research reported on in this paper were performed in the context of the EU-FP7 project ALIZ-E (ICT-248116), which develops embodied cognitive robots for believable any-depth affective interactions with young users over an extended and possibly discontinuous period [1].

ALIZ-E project . We focus on the implementation for two scenarios: an imitation game of arm movements and a quiz game.

2 Event-Based Component Integration

Due to the limited processing power and memory of the Nao's on-board computer, many of the system's components must run on one or more PCs in the network. As in other projects, we have pre-existing software in different programming languages running on different operating systems; an integration framework is thus required.

The open source Urbi SDK [4] has been chosen as the middleware in the ALIZ-E project. It aims at providing a universal programming environment orchestrating complex components. As a client-server architecture where the server is running on the robot, it is possible to integrate remote components written in C/C++ or Java with components that run directly on the robot. Urbi comes with a dedicated language, UrbiScript, which provides a number of interesting paradigms, for example for event-based and parallel programming, and can access and control the sensors and actuators of the Nao robot.

Similar to other interactive systems, we had the choice between different component integration paradigms; the most popular ones appear to be publish-subscribe messaging [5] and blackboard [6] architectures. Essential requirements include proper encapsulation of components to ensure maintainability of the software; the flexible rearrangement of information flow; and a notification mechanism allowing a component to initiate the flow of information.

Urbi's event objects provide a suitable mechanism for implementing a paradigm close to publish-subscribe messaging: an Urbi event can carry arbitrary values as a 'payload'. Components can trigger events whenever new data is available, a certain processing stage has been reached, etc. A controller script, written in UrbiScript, implements event handlers which pass the information on to the appropriate components. The advantage of this approach is that all event handlers for a given instantiation of the system are maintained in a single file; beyond mere message passing, they can also provide additional functionality such as centralized logging.

3 The Integrated System

Using the event-based approach introduced above, we have integrated the components shown in Fig. 1 and described in the following sections.

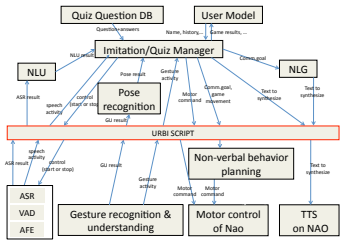

Fig. 1 System components.

3.1 Dialogue Manager (DM)

The DM is responsible for controlling the behaviour of the robot in the interaction. It is based on the information-state update approach [7]. Its action-selection mechanism is implemented as a non-deterministic Finite State Machine (SFSM), using three different types of machine states: (1) *dialogue states* describe different kinds of observations such as dialogue acts, gestures and timeouts; (2) *action states* are either dialogue actions (e.g. Ask(Name)) or high-level motor commands (e.g. Move(Hello)) with the ability of parallel execution to combine verbal and non-verbal behaviour; (3) *call states* are used to switch between FSM-based dialogue policies, and thus control the interaction in a modular way.

The current DM uses hand-written dialogue policies that encode the overall structure of the interaction, interpret conflicts between gesture and speech input, allow for simple clarifications and select gesture an peech output. The DM maintains a representation of the dialogue context in its information state and gets additional input from the User Model (e.g., user name, previous game scores, etc.). The questions for the Quiz Game come from the Question Database. Ongoing work consists in incorporating reinforcement learning for optimizing the robot's behaviour [8].

3.2 Audio Front End (AFE) and Voice Activity Detection (VAD)

The AFE component captures the speech signal from the microphones, makes some preliminary preprocessing like sample rate conversion, then sends the audio buffers to the VAD component.

The VAD module allows the robot to detect that dynamically varying sound sources which could be of interest for further analysis, such as human speech, are active, in order to trigger further processing. We examined the performance of different VAD algorithms for different background noise types and proposed a robust energy based VAD algorithm [9]. It uses the feature of smoothed energy contained in the frequency region of interest and can be configured for different background noise types. It outperforms conventional spectrum based VAD under certain noise conditions and for certain configurations [9]. The VAD implementation in ALIZ-E encompasses the functionality of recording the audio segments of interest. These audio segments are made available for further processing via Urbi event emission.

3.3 Automatic Speech Recognition (ASR)

We chose CMU Sphinx[2] for its robust coverage of the features important for speech recognition in the project: large-vocabulary continuous ASR, N-best list output and

[2] CMU Sphinx is an open-source project at Carnegie Mellon University, Pittsburgh.

adaptation techniques. We integrated the three main Sphinx modules in Urbi: feature extraction, main decoding (Viterbi forward search) and N-best lists generation.

To carry out interaction with children in Italian, we built an acoustic model using the Italian children voice corpus ChildIt [10]. We applied speaker adaptation techniques, using a small amount of data from individual speakers to improve the speaker-independent model. Using Vocal Tract Length Normalization and Maximum Likelihood Linear Regression the phonetic error rate in the ChildIt data improved from 29.8% for the baseline to 28.2% (cf. [11] for details).

The results are not good yet in the practical application of the system, but we plan to improve the score collecting more data from children and trying different online adaptation techniques: since each child will interact several times with the robot we can consider reusing data from previous interactions for adapting the models. Further planned improvements include incremental ASR implementation and dynamic language model creation.

3.4 Natural Language Understanding (NLU)

The N-best list from ASR is recombined into a compacted lattice before parsing proper to avoid re-analysing common subsequences. Several heuristics are applied to merge similar edges or nodes to reduce lattice size and thereby the parsing effort. The effects of these heuristics on parsing performance and accuracy still have to be measured with real data. The compacted lattice is then directly analysed by an agenda-based chart parser, which uses a hand-written competence grammar based on the Multimodal Combinatory Categorial Grammar framework [12] implemented in OpenCCG [13]. The agenda, together with an appropriate search strategy, allows to focus parsing on the promising parts of the input, and thus aims to find the best partial analyses. Currently, the scores from the speech recognizer are used to guide the search. It is planned to enhance this by using a statistical model based on the CCG grammar and on context information coming from the dialogue manager.

3.5 Natural Language Generation (NLG)

System output is generated either as canned text sent directly from the DM to the TTS component, or we employ a deep-generation approach, involving utterance content planning on the basis of a communicative goal specified by the DM as a logical form and grammar-based surface realization. For the task of utterance content planning we have implemented a graph rewriting component based on transformation rules. The power and the utility of this component is enhanced by additional functionality, such as function plugins to access external information sources during processing, e.g., the dialogue information state. In order to avoid repetition of the same output, the utterance planning rules provide a range of realization vari-

ants for each communicative goal. Selection among the variants is either driven by dialogue context or made at random. The utterance planning for the verbalization of arm movements in the imitation game also includes adaptation of the wording of system output to that of the user. Surface realization of the system utterances is obtained using the OpenCCG realizer [13] and uses the same handwritten grammar for Italian as mentioned above for parsing.

3.6 Text-To-Speech Synthesis (TTS)

For synthesizing the audio output, the commercial Acapela TTS system [14] available with the Nao robot is currently used. However, the ALIZ-E project requires a more customizable and flexible TTS system in the long run for the sake of emotionally expressive synthesis. We are therefore also integrating the open source Mary TTS platform [15]. The advantage of using Mary TTS is that it supports state of the art technology in the field of HMM-synthesis [16] , and enables us to experiment with the manipulation of para-verbal parameters (e.g. pitch shape, speech rate, voice intensity, pause durations) for the purpose of expressive speech synthesis, and the voice quality and timbre modifications algorithms [17] useful to to convert an adult TTS voice into a child like voice.

3.7 Gesture Recognition and Understanding (GRU)

The GRU component currently detects four types of events used in the imitation game: left hand up, left hand down, right hand up, and right hand down (and combinations thereof). Skin detection plays an important role in tracing hands. Although we can use a person-specific skin model by detecting the user's face and building a skin histogram from the face pixels, in practice it turns out that this strategy does not reliably detect the hands. The main reason is that some hand-skin pixels do not match the face-skin histogram due to differences in illumination/shadows. We therefore use a general skin model and a Bayesian skin detection approach [18]. The detection threshold is adjusted to reach a high detection rate, to make sure the hands are entirely present in the skin detection mask. As a consequence, we get an increased number of false positives. To overcome this issue, we obtain a skin-motion mask by combining the skin detection mask with a motion history image [19]. Ideally, only the moving pixels will be present in the skin-motion mask, thereby the skin-like background pixels being eliminated. Further, the position of the head is identified by a face detection and tracking algorithm [20] in order to define the vertical areas where the hands might move. By analyzing the number of pixels in these areas, we deterministically derive the nature of the event: hand up or down. The proposed GRU scheme achieves real-time operation and does not need a labeled video

database to learn simple hand gestures. However, in the future, we foresee the use of such a database for learning more complex gestures in a probabilistic framework.

3.8 Non-Verbal Behavior Planning (NVBP) & Motor Control (MC)

Displaying interpretable emotional body language during interaction should greatly improve the robot's acceptance. Since Nao cannot display facial expressions, body language is an appropriate medium to express emotions. Previous work shows that it is possible to display emotions using static key poses [21, 22]. The non-verbal feedback in our system currently includes empirically validated poses expressing anger, sadness, fear, happiness, excitement and pride [22].

The NVBP component decides whether to express an emotion non-verbally and which expression should be displayed. The decision is based on information regarding the situation. The MC component is responsible for implementing the physical movements of the robot. Future work will improve the decision mechanism and will address the use of movements to increase the acceptance of the robot.

4 Experience from Experiments and Conclusions

We presented an HRI system that recognizes input and produces output in the form of speech and gestures. We described the system components and their integration in Urbiusing an event-based approach. This approach supports integration of components in different programming languages, running in parallel, distributed on several computers. Components exchange information by values carried as 'payload' by the events. Components trigger events whenever new data is available, a certain processing stage has been reached, etc., resulting in a flexible processing model.

Parts of the system have been evaluated for component functionality. Technical evaluation of the intended functionality has been conducted for the system as a whole. This proves that the event-based control mechanism and the interfaces between components works as intended. It also shows certain limitations of the robot platform that future design will have to take into account (e.g., noisy input, instability, overheating). While the input recognition and interpretation for both speech and gestures work sufficiently for demonstration purposes, they are not robust enough for untrained users. The fully autonomous system is thus not yet mature enough for end-to-end usability evaluation. We are carrying out wizard-of-oz experiments where the wizard provides interpretations for speech and gesture input, and the rest of the system functions autonomously. Besides feedback on usability, these experiments serve data collection purposes for further system development.

References

1. "ALIZ-E website," (accessed 5.3.2011), http://aliz-e.org/.
2. O. Lemon, A. Bracy, A. Gruenstein, and S. Peters, "The WITAS multi-modal dialogue system I," in *EUROSPEECH*, Aalborg, Denmark, Sep 2001, pp. 1559–1562.
3. R. Stiefelhagen, H. Ekenel, C. Fugen, P. Gieselmann, H. Holzapfel, F. Kraft, K. Nickel, M. Voit, and A. Waibel, "Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot," vol. 23, no. 5, 2007, pp. 840–851.
4. J. Baillie, "URBI: Towards a Universal Robotic Low-Level Programming Language," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3219–3224.
5. M. Schröder, "The SEMAINE API: towards a standards-based framework for building emotion-oriented systems," *Advances in Human-Computer Interaction*, vol. 2010, no. 319406, 2010.
6. N. Hawes, J. L. Wyatt, A. Sloman, M. Sridharan, R. Dearden, H. Jacobsson, and G. Kruijff, "Architecture and representations," in *Cognitive Systems*, H. I. Christensen, A. Sloman, G. Kruijff, and J. Wyatt, Eds. published online at http://www.cognitivesystems.org/cosybook/, 2009, pp. 53–95.
7. S. Larsson and D. Traum, "Information state and dialogue management in the TRINDI dialogue move engine toolkit," *Natural Language Engineering*, vol. 5, no. 3-4, pp. 323–340, 2000.
8. H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira, "Evaluation of a hierarchical reinforcement learning spoken dialogue system," *Computer Speech and Language*, vol. 24, no. 2, pp. 395–429, 2010.
9. T. Dekens and W. Verhelst, "On the noise robustness of voice activity detection algorithms," in *Proc. of InterSpeech, Florence, Italy*, August 2011.
10. M. Gerosa, D. Giuliani, and F. Brugnara, "Acoustic variability and automatic recognition of children's speech," *Speech Communication*, vol. 49, pp. 847–860, Feb 2007.
11. M. Nicolao and P. Cosi, "Comparing SPHINX vs. SONIC Italian Children Speech Recognition Systems," in *7th Conference of the Italian Association of Speech Sciences*, Feb 2011, unpublished draft version.
12. J. Baldridge and G.-J. Kruijff, "Multi-modal combinatory categorial grammar," in *Proceedings of 10th Annual Meeting of the European Association for Computational Linguistics*, 2003.
13. "OpenCCG website," (accessed 5.3.2011), http://openccg.sourceforge.net/.
14. "Acapela website," (accessed 5.3.2011), http://www.acapela-group.com/index.html.
15. "Mary TTS website," (accessed 5.3.2011), http://mary.dfki.de/.
16. H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of ISCA SSW6*, 2007, pp. 294–299.
17. F. Tesser, E. Zovato, M. Nicolao, and P. Cosi, "Two Vocoder Techniques for Neutral to Emotional Timbre Conversion," in *7th Speech Synthesis Workshop (SSW)*, Y. Sagisaka and K. Tokuda, Eds. Kyoto, Japan: ISCA, 2010, pp. 130–135.
18. M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int. J. Comput. Vision*, vol. 46, pp. 81–96, 2002.
19. G. Bradski and J. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, pp. 174–184, 2002.
20. "OpenCV library website," (accessed 5.3.2011), http://opencv.willowgarage.com.
21. A. Beck, A. Hiolle, A. Mazel, and L. Cañamero, "Interpretation of emotional body language displayed by robots," in *Proceedings of the 3rd international workshop on Affective interaction in natural environments*, ser. AFFINE '10. New York, NY, USA: ACM, 2010, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/1877826.1877837
22. A. Beck, L. Cañamero, and K. Bard, "Towards an affect space for robots to display emotional body language," in *Proceedings of the 19th IEEE international symposium on robot and human interactive communication*, ser. Ro-Man 2010. IEEE, 2010, pp. 464–469.