

An Interactive Humanoid Robot Exhibiting Flexible Sub-Dialogues*

Heriberto Cuayáhuil
DFKI GmbH
hecu01@dfki.de

Ivana Kruijff-Korbayová
DFKI GmbH
ivana.kruijff@dfki.de

Abstract

We demonstrate a conversational humanoid robot that allows users to follow their own dialogue structures. Our system uses a hierarchy of reinforcement learning dialogue agents, which support transitions across sub-dialogues in order to relax the strictness of hierarchical control and therefore support flexible interactions. We demonstrate our system with the Nao robot playing two versions of a Quiz game. Whilst language input and dialogue control is autonomous or wizarded, language output is provided by the robot combining verbal and non-verbal contributions. The novel features in our system are (a) the flexibility given to users to navigate flexibly in the interaction; and (b) a framework for investigating adaptive and flexible dialogues.

1 Introduction

Hierarchical Dialogue Control (HDC) consists of behaviours or discourse segments at different levels of granularity executed from higher to lower level. For example, a dialogue agent can invoke a sub-dialogue agent, which can also invoke a sub-sub-dialogue agent, and so on. Task-oriented dialogues have shown evidence of following hierarchical structures (Grosz and Sidner, 1986; Litman and Allen, 1987; Clark, 1996). Practically speaking, HDC offers the following benefits. First, modularity helps to specify sub-dialogues that may be easier to specify than the entire full dialogues. Second, sub-dialogues may include only relevant dialogue knowledge (e.g. subsets of dialogue acts), thus reducing significantly their com-

*Funding by the EU-FP7 project ALIZ-E (ICT-248116) is gratefully acknowledged.

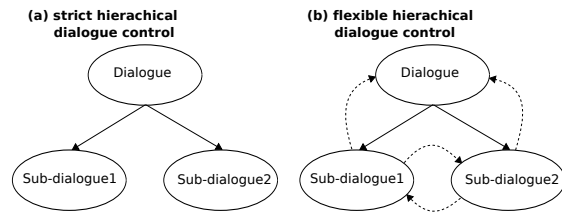


Figure 1: Hierarchies of dialogue agents with strict (top down) and flexible control (partial top down).

plexity. Third, sub-dialogues can be reused when dealing with new behaviours. In this paper we distinguish two types of hierarchical dialogue control: strict and flexible. These two forms of dialogue control are shown in Figure 1. It can be observed that strict HDC is based on a pure top down execution, and flexible HDC is based on a combined hierarchical and graph-based execution.

The main limitation of *strict* HDC is that human-machine interactions are rigid, i.e. the user cannot change the imposed dialogue structure. A more natural way of interaction is by relaxing the dialogue structure imposed by the conversational machine. The advantage of *flexible* HDC is that interactions become less rigid because it follows a partially specified hierarchical control, i.e. the user is allowed to navigate across the available sub-dialogues. In addition, another important property of the latter form of HDC is that we can model flexible dialogue structures not only driven by the user but also by the machine. The latter requires the machine to learn the dialogue structure in order to behave in an adaptive way. The rest of the paper describes a demo system exhibiting both types of behaviour, based on a reinforcement learning dialogue framework.

2 Hierarchical Reinforcement Learning Dialogue Agents with Flexible Control

Our dialogue controllers use hierarchical reinforcement learning as in (Cuayáhuitl et al., 2010). We extend such a formalization through a hierarchy of dialogue agents defined with the following tuples: $M_j^i = \langle S_j^i, A_j^i, T_j^i, R_j^i, L_j^i, U_j^i, \gamma_j^i, \delta_j^i \rangle$, where S_j^i is a set of states, A_j^i is a set of actions, T_j^i is a stochastic state transition function, R_j^i is a reward function, L_j^i is a grammar that specifies tree-based state representations, U_j^i is a finite set of user actions (e.g. user dialogue acts), γ_j^i is a finite set of models that subtask M_j^i is being allowed to transition to, and $\delta_j^i = P(m' \in \gamma_j^i | m \in \gamma_j^i, u \in U_j^i)$ is a stochastic model transition function¹ that specifies the next model m' given model m and user action u . Although the hierarchy of agents can be fully-connected when all models are allowed to transition from a given particular model (avoiding self-transitions), in practice, we may want our hierarchy of agents partially-connected, i.e. when γ_j^i is a subset of subtasks that agent M_j^i is allowed to transition to.

We implemented a modified version of the *HSMQ-Learning* algorithm (Dietterich, 2000) to simultaneously learn a hierarchy of policies π_j^i . This algorithm uses a *stack* of subtasks and operates as illustrated in Figure 2. If during the execution of a subtask the user decides to jump to another subtask, i.e. to change to another sub-dialogue, the flexible execution of subtasks allows each subtask to be interrupted in two ways. In the first case, we check whether the new (active) subtask is already on the stack of subtasks to execute. This would be the case if it was a parent of the current subtask. In this case, we terminate execution of all intervening subtasks until we reach the parent subtask, which would be the new active subtask. Notice that termination of all intervening subtasks prevents the stack from growing infinitely. In the second case, the current subtask is put on hold, and if the new active subtask is not already on the stack of subtasks to execute, it is pushed onto the stack and control is passed to it. Once the new subtask terminates its execution, control is transferred back to the subtask on hold.

¹This is a very relevant feature in dialogue agents in order to allow users to say and/or do anything at anytime, and the learning agents have to behave accordingly.

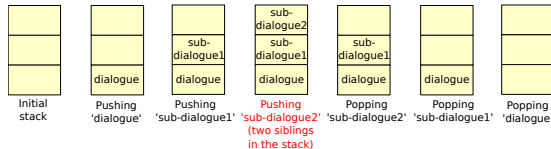


Figure 2: Hypothetical operations of stack-based hierarchical dialogue controllers. Whilst the fourth operation from left to right is not allowed in strict HDC, all stack operations are allowed in flexible HDC.

These kinds of transitions can be seen as high-level transitions in the state space. They can also be seen as the mechanism to transition from any state to any other in the hierarchy. To do that we maintain an activity status for each subtask M_j^i , where only one subtask is allowed to be active at a time. We maintain a knowledge-rich state that keeps the dialogue history in order to initialize or reinitialize states of each subtask accordingly. Since there is learning when new subtasks are invoked and no learning when they are interrupted, this algorithm maintains its convergence properties to optimal context-independent policies.

3 A Hierarchy of Dialogue Agents for Playing Quiz Games

We use a small hierarchy of dialogue agents—for illustration purposes—with one parent agent and two children agents (‘robot asks’ and ‘user asks’). Thus, the hierarchy of agents can ask the user questions, and vice-versa, the user can ask the robot questions (described in the next section). Both conversants can play multiple rounds with a predefined number of questions.

Due to space restrictions, we describe the hierarchy of agents only briefly. The set of states and actions use relational representations (they can be seen as trees) in order to specify the state-action space compactly, which can grow as more features or games are integrated. Dialogue and game features are included so as to inform the agents of possible situations in the interaction. The action sets use constrained spaces, i.e. only a subset of actions is available at each state based on the relational representations. For example, the action $Request(PlayGame) \leftarrow x_0$ is valid for the dialogue state x_0 expressed as $Salutation(greeting) \wedge UserName(known) \wedge PlayGame(unknown)$. The sets of primitive actions (80 in total) assume verbal behaviours

with a mapping to non-verbal ones, some sample dialogue act types are as follows: requests, apologies, confirmations, provide information, acknowledgements, feedback, non-verbal expressions, game-related actions. The transition functions use pre-defined parameters, their training from data is left as future work. The reward function addresses efficient and effective interactions by penalizing dialogue length and encouraging to continue playing. The dialogue agents learnt their behaviour by interacting with a stochastic simulated user, where the user responses eventually required transitions across agents. A sample dialogue with flexible interaction is shown in Fig. 3.

4 A Humanoid Robot Integrated System

Figure 4 shows the robot’s integrated system, which equips the robot with the following capabilities: listening, talking, seeing and moving.² A sample interaction assuming wizarded behaviour is as follows. The user says something to the robot (e.g. ‘ask me a question’) and the wizard selects a user dialogue act for the corresponding utterance such as ‘Request(Question)’. Then the dialogue manager selects and executes the learned action by querying/updating the game move generator and user model components, and by passing a dialogue act to the language generation and motor control components. The former provides the corresponding text to the speech synthesizer and the latter provides the corresponding non-verbal behaviour. Then the dialogue manager observes its new dialogue state, and suggests the next actions—based on its dialogue policies and user simulations—in the wizard-of-Oz GUI, see Figure 5. From this setting, if we remove the wizard then the system can behave autonomously, where user inputs are derived from the language and gesture understanding components.

The attractiveness of this kind of set-up is the smooth transition from purely wizarded conversational behaviour to fully autonomous behaviour, where the dialogue policies can be tested and eval-

²We use off-the-shelf technologies such as Sphinx / Julius / HTK for speech recognition, OpenCV for gesture recognition, Acapela and Mary for speech synthesis, HRL-based dialogue management (as described above), OpenCCG for language parsing and language generation, Weka and JavaBayes for maintaining a probabilistic and personalized user profile of asked and answered questions, the URBI middleware (www.gostai.com), and the NAO humanoid robot (www.aldebaran-robotics.com).

SYS	Hi, my name is NAO, and you?
USR	<i>I am Peter</i>
SYS	Nice to meet you Peter
SYS	Do you want to Play a Quiz game?
USR	<i>Ask me questions [jump to 'robot asks']</i>
SYS	The first question is. What is a typical magic instrument? Your choices are: One, guitar. Two, drums. Three, violin. Four, flute. What is your choice?
USR	<i>Number two</i>
SYS	That is not quite right. Do you have another choice?
USR	<i>Maybe number three</i>
SYS	The correct answer is number four, flute. The next question is: which one of these is an important artificial channel of the central America? Your choices are: One, channel five. Two, Panama canal. Three, Suez canal. Four, English channel. What is your choice?
USR	<i>Now I want to ask you questions [jump to 'user asks']</i>
SYS	Ask me the first question.
USR	<i>What does water become at the temperature of zero degrees?</i>
SYS	What are my choices?
USR	<i>Your choices are: One, condensation. Two, ice. Three, steam.</i>
SYS	I choose number one
USR	<i>That is incorrect.</i>
SYS	So what is the right answer?
USR	<i>The correct answer is number two.</i>
SYS	Okay, ask me another question.
USR	<i>The next question is: through which breach did the Italian army entered into Rome in 1870? Your choices are: One, porta venezia. Two, porta pia. Three, porta pazienza. Four, porta nuova. What is your choice?</i>
SYS	Paris
USR	<i>Let's stop playing. [jump to 'root']</i>
SYS	Did you like playing the Quiz Game?
USR	<i>Yes</i>
SYS	I am glad to hear that.
SYS	It was nice playing with you, see you soon. Bye!

Figure 3: Sample dialogue with non-strict hierarchical control. The groups of utterances represent activity in different sub-dialogues in the hierarchy of agents. Note that the user is allowed to override the imposed dialogue structure. For instance, when the user says ‘Ask me questions’ the agent interrupts the execution of the root subtask and transitions to the subtask ‘robot asks’. Similarly, the sub-dialogues do not need to follow their imposed structure and the user is allowed to talk about previous and unaddressed sub-dialogues such as the sudden switch from ‘robot asks’ to ‘user asks’.

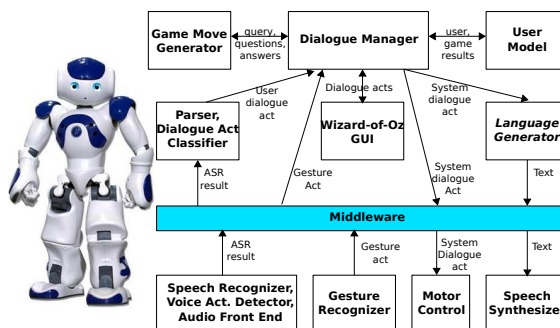


Figure 4: High-level architecture of our talking robot.

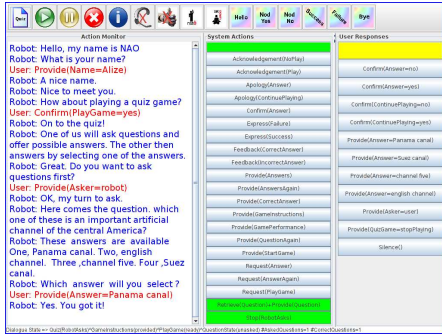


Figure 5: Screen shot of the wizard-of-Oz GUI, where the dialogue policies and user simulations suggest highlighted actions to the wizard. This setting allows fully-wizarded and (semi-) autonomous behaviour.



Figure 6: The Nao robot greeting a user prior to playing a Quiz game. The pieces of paper on the table are the Quiz questions the child asks the robot.

uated with (semi-) autonomous behaviour. We use this framework to investigate long-term human-robot interaction, in particular child-robot interactions for educational purposes. Figure 6 shows a scene from a pilot evaluation, where the robot and a child are visibly engaged with each other. A complete evaluation with simulated and real dialogues will be reported in a forthcoming paper.

5 Discussion and Summary

Typically, conversational interfaces impose a dialogue structure on the user. Even in dialogue systems with mixed-initiative interaction that give flexibility to the user in terms of providing more than one piece of information at a time, the user is hardly allowed to navigate flexibly during the interaction. Notable exceptions without dialogue optimization are (Rudnicky and Wu, 1999; Lemon et al., 2001; Larsson, 2002; Foster et al., 2006). We believe that Hierarchical Reinforcement Learning with global state transitions is an

interesting method to optimize (sub-) dialogues at different levels of granularity, where the design of action selection might not be easy to hand-craft. On the one hand, our HDCs can be applied to dialogues with user-driven topic shift, where the user can take control of the interaction by navigating across sub-dialogues and the system has to respond accordingly. On the other hand, our HDCs can be applied to dialogues with system-driven topic shift, where the system can itself terminate a sub-dialogue, perhaps by inferring the user’s emotional and/or situational state, and the system has to switch itself to another sub-dialogue.

We have described a conversational humanoid robot that allows users to follow their own dialogue structures. The novelty in our system is its flexible hierarchical dialogue controller, which extends strict hierarchical control with transitions across sub-controllers. Suggested future work consists in training and evaluating our humanoid robot from real interactions using either partially specified or fully learnt dialogue structures.

References

H. Clark. 1996. *Using Language*. Cambridge University Press.

H. Cuayáhuít, S. Renals, O. Lemon, and H. Shimodaira. 2010. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech and Language*, 24(2):395–429.

T. Dietterich. 2000. An overview of MAXQ hierarchical reinforcement learning. In *Symposium on Abstraction, Reformulation, and Approximation (SARA)*, pages 26–44.

M. E. Foster, T. By, M. Rickert, and A. Knoll. 2006. Human-robot dialogue for joint construction tasks. In *ICMI*, pages 68–71.

B. Grosz and C. Sidner. 1986. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

S. Larsson. 2002. *Issue-Based Dialogue Management*. Ph.D. thesis, University of Goteborg.

O. Lemon, A. Bracy, A. Gruenstein, and S. Peters. 2001. The WITAS multi-modal dialogue system I. In *EUROSPEECH*, Aalborg, Denmark.

D. Litman and J. Allen. 1987. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11:163–200.

A. Rudnicky and W. Wu. 1999. An agenda-based dialogue management architecture for spoken language systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 337–340, Keystone, Colorado, USA, Dec.