Multimodal Generation in the COMIC Dialogue System

Mary Ellen Foster and **Michael White**
Institute for Communicating and Collaborative Systems
School of Informatics, University of Edinburgh
{M.E.Foster,Michael.White}@ed.ac.uk

Andrea Setzer and **Roberta Catizone**
Natural Language Processing Group
Department of Computer Science, University of Sheffield
{A.Setzer,R.Catizone}@dcs.shef.ac.uk

Abstract

We describe how context-sensitive, user-tailored output is specified and produced in the COMIC multimodal dialogue system. At the conference, we will demonstrate the user-adapted features of the dialogue manager and text planner.

1 Introduction

COMIC[1] is an EU IST 5th Framework project combining fundamental research on human-human interaction with advanced technology development for multimodal conversational systems. The project demonstrator system adds a dialogue interface to a CAD-like application used in bathroom sales situations to help clients redesign their rooms. The input to the system includes speech, handwriting, and pen gestures; the output combines synthesised speech, a talking head, and control of the underlying application. Figure 1 shows screen shots of the COMIC interface.

There are four main phases in the demonstrator. First, the user specifies the shape of their own bathroom, using a combination of speech input, pen-gesture recognition and handwriting recognition. Next, the user chooses a layout for the sanitary ware in the room. After that, the system guides the user in browsing through a range of tiling options for the bathroom. Finally, the user is given a three-dimensional walkthrough of the finished bathroom. We will focus on how context-sensitive, user-tailored output is generated in the third, guided-browsing phase of the interaction. Figure 2 shows a typical user request and response from COMIC in this phase. The pitch accents and multimodal actions are indicated; there is also facial emphasis corresponding to the accented words.

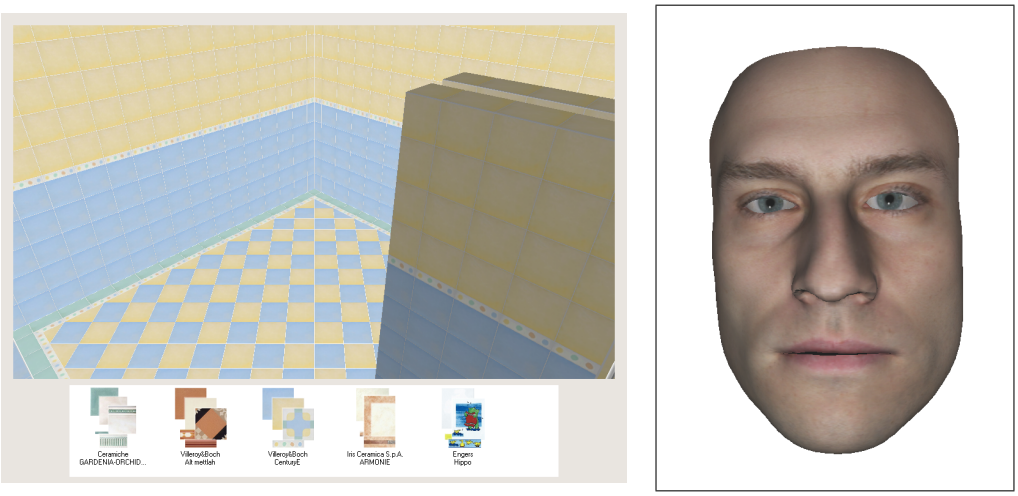
The primary goal of COMIC's guided-browsing phase is to help users become better informed about the range of tiling options for their bathroom. In this regard, it is similar to the web-based system M-PIRO (Isard et al., 2003), which generates personalised descriptions of museum objects, and contrasts with task-oriented embodied dialogue systems such as SmartKom (Wahlster, 2003). Since guided browsing requires extended descriptions, in COMIC we have placed greater emphasis on producing high-quality adaptive output than have previous embodied dialogue projects such as August (Gustafson et al., 1999) and Rea (Cassell et al., 1999). To generate its adaptive output, COMIC uses information from the dialogue history and the user model throughout the generation process, as in FLIGHTS (Moore et al., 2004); both systems build upon earlier work on adaptive content planning (Carenini, 2000; Walker et al., 2002). An experimental study (Foster and White, 2005) has shown that this adaptation is perceptible to users of COMIC.

2 Dialogue Management

The task of the Dialogue and Action Manager (DAM) is to decide what the system will show and say in response to user input. The input to the

[1] COnversational Multimodal Interaction with Computers; http://www.hcrc.ed.ac.uk/comic/.

Proceedings of the ACL Interactive Poster and Demonstration Sessions,
pages 45–48, Ann Arbor, June 2005.



(a) Bathroom-design application (b) Talking head

Figure 1: Components of the COMIC interface

User	Tell me about this design *[click on Alt Mettlach]*
COMIC	*[Look at screen]*
	THIS DESIGN is in the CLASSIC style. *[circle tiles]*
	As you can see, the colours are DARK RED and OFF WHITE. *[point at tiles]*
	The tiles are from the ALT METTLACH collection by VILLEROY AND BOCH. *[point at design name]*

Figure 2: Sample COMIC input and output

DAM consists of multiple scored hypotheses containing high-level, modality-independent specifications of the user input; the output is a similar high-level specification of the system action. The DAM itself is modality-independent. For example, the input in Figure 2 could equally well have been the user simply pointing to a design on the screen, with no speech at all. This would have resulted in the same abstract DAM input, and thus in the same output: a request to show and describe the given design.

The COMIC DAM (Catizone et al., 2003) is a general-purpose dialogue manager which can handle different dialogue management styles such as system-driven, user-driven or mixed-initiative. The general-purpose part of the DAM is a simple stack architecture with a control structure; all the application-dependent information is stored in a variation of Augmented Transition Networks (ATNs) called *Dialogue Action Forms* (DAFs). These DAFs represent general dialogue moves, as well as sub-tasks or topics, and are pushed onto and popped off of the stack as the dialogue proceeds.

When processing a user input, the control structure decides whether the DAM can stay within the current topic (and thus the current DAF), or whether a topic shift has occurred. In the latter case, a new DAF is pushed onto the stack and executed. After that topic has been exhausted, the DAM returns to the previous topic automatically. The same principle holds for error handling, which is implemented at different levels in our approach.

In the guided-browsing phase of the COMIC system, the user may browse tiling designs by colour, style or manufacturer, look at designs in detail, or change the amount of border and decoration tiles. The DAM uses the system ontology to retrieve designs according to the chosen feature, and consults the user model and dialogue history to narrow down the resulting designs to a small set to be shown and described to the user.

3 Presentation Planning

The COMIC fission module processes high-level system-output specifications generated by the DAM. For the example in Figure 2, the DAM output indicates that the given tile design should be shown and described, and that the description must mention the style. The fission module fleshes out such specifications by selecting and structuring content, planning the surface form of the text to realise that content, choosing multimodal behaviours to accompany the text, and controlling the output of the whole schedule. In this section, we describe the planning process; output coordination is dealt with in Section 6. Full technical details of the fission module are given in (Foster, 2005).

To create the textual content of a description, the fission module proceeds as follows. First, it gathers all of the properties of the specified design from the system ontology. Next, it selects the properties to include in the description, using information from the dialogue history and the user model, along with any properties specifically requested by the dialogue manager. It then creates a structure for the selected properties and creates logical forms as input for the OpenCCG surface realiser. The logical forms may include explicit alternatives in cases where there are multiple ways of expressing a property; for example, it could say either *This design is in the classic style* or *This design is classic*. OpenCCG makes use of statistical language models to choose among such alternatives. This process is described in detail in (Foster and White, 2004; Foster and White, 2005).

In addition to text, the output of COMIC also incorporates multimodal behaviours including prosodic specifications for the speech synthesiser (pitch accents and boundary tones), facial behaviour specifications (expressions and gaze shifts), and deictic gestures at objects on the application screen using a simulated pointer. Pitch accents and boundary tones are selected by the realiser based on the context-sensitive information-structure annotations (theme/rheme; marked/unmarked) included in the logical forms. At the moment, the other multimodal coarticulations are specified directly by the fission module, but we are currently experimenting with using the OpenCCG realiser's language models to choose them, using example-driven techniques.

4 Surface Realisation

Surface realisation in COMIC is performed by the OpenCCG[2] realiser, a practical, open-source realiser based on Combinatory Categorial Grammar (CCG) (Steedman, 2000b). It employs a novel ensemble of methods for improving the efficiency of CCG realisation, and in particular, makes integrated use of *n*-gram scoring of possible realisations in its chart realisation algorithm (White, 2004; White, 2005). The *n*-gram scoring allows the realiser to work in "anytime" mode—able at any time to return the highest-scoring complete realisation—and ensures that a good realisation can be found reasonably quickly even when the number of possibilities is exponential. This makes it particularly suited for use in an interactive dialogue system such as COMIC.

In COMIC, the OpenCCG realiser uses factored language models (Bilmes and Kirchhoff, 2003) over words and multimodal coarticulations to select the highest-scoring realisation licensed by the grammar that satisfies the specification given by the fission module. Steedman's (Steedman, 2000a) theory of information structure and intonation is used to constrain the choice of pitch accents and boundary tones for the speech synthesiser.

5 Speech Synthesis

The COMIC speech-synthesis module is implemented as a client to the Festival speech-synthesis system.[3] We take advantage of recent advances in version 2 of Festival (Clark et al., 2004) by using a custom-built unit-selection voice with support for APML prosodic annotation (de Carolis et al., 2004). Experiments have shown that synthesised speech with contextually appropriate prosodic features can be perceptibly more natural (Baker et al., 2004).

Because the fission module needs the timing information from the speech synthesiser to finalise the schedules for the other modalities, the synthesiser first prepares and stores the waveform for its input text; the sound is then played at a later time, when the fission module indicates that it is required.

[2] http://openccg.sourceforge.net/

[3] http://www.cstr.ed.ac.uk/projects/festival/

6 Output Coordination

In addition to planning the presentation content as described earlier, the fission module also controls the system output to ensure that all parts of the presentation are properly coordinated, using the timing information returned by the speech synthesiser to create a full schedule for the turn to be generated.

As described in (Foster, 2005), the fission module allows multiple segments to be prepared in advance, even while the preceding segments are being played. This serves to minimise the output delay, as there is no need to wait until a whole turn is fully prepared before output begins, and the time taken to speak the earlier parts of the turn can also be used to prepare the later parts.

7 Acknowledgements

This work was supported by the COMIC project (IST-2001-32311). This paper describes only part of the work done in the project; please see `http://www.hcrc.ed.ac.uk/comic/` for full details. We thank the other members of COMIC for their collaboration during the course of the project.

References

Rachel Baker, Robert A.J. Clark, and Michael White. 2004. Synthesizing contextually appropriate intonation in limited domains. In *Proceedings of 5th ISCA workshop on speech synthesis*.

Jeff Bilmes and Katrin Kirchhoff. 2003. Factored language models and general parallelized backoff. In *Proceedings of HLT-03*.

Giuseppe Carenini. 2000. *Generating and Evaluating Evaluative Arguments*. Ph.D. thesis, Intelligent Systems Program, University of Pittsburgh.

Justine Cassell, Timothy Bickmore, Mark Billinghurst, Lee Campbell, Kenny Chang, Hannes Vilhjálmsson, and Hao Yan. 1999. Embodiment in conversational interfaces: Rea. In *Proceedings of CHI99*.

Roberta Catizone, Andrea Setzer, and Yorick Wilks. 2003. Multimodal dialogue management in the COMIC project. In *Proceedings of EACL 2003 Workshop on Dialogue Systems: Interaction, adaptation, and styles of management*.

Robert A.J. Clark, Korin Richmond, and Simon King. 2004. Festival 2 – build your own general purpose unit selection speech synthesiser. In *Proceedings of 5th ISCA workshop on speech synthesis*.

Berardina de Carolis, Catherine Pelachaud, Isabella Poggi, and Mark Steedman. 2004. APML, a mark-up language for believable behaviour generation. In H Prendinger, editor, *Life-like Characters, Tools, Affective Functions and Applications*, pages 65–85. Springer.

Mary Ellen Foster and Michael White. 2004. Techniques for text planning with XSLT. In *Proceedings of NLPXML-2004*.

Mary Ellen Foster and Michael White. 2005. Assessing the impact of adaptive generation in the COMIC multimodal dialogue system. In *Proceedings of IJCAI-2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. To appear.

Mary Ellen Foster. 2005. Interleaved planning and output in the COMIC fission module. Submitted.

Joakim Gustafson, Nikolaj Lindberg, and Magnus Lundeberg. 1999. The August spoken dialogue system. In *Proceedings of Eurospeech 1999*.

Amy Isard, Jon Oberlander, Ion Androtsopoulos, and Colin Matheson. 2003. Speaking the users' languages. *IEEE Intelligent Systems*, 18(1):40–45.

Johanna Moore, Mary Ellen Foster, Oliver Lemon, and Michael White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proceedings of FLAIRS 2004*.

Mark Steedman. 2000a. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689.

Mark Steedman. 2000b. *The Syntactic Process*. MIT Press.

Wolfgang Wahlster. 2003. SmartKom: Symmetric multimodality in an adaptive and reusable dialogue shell. In *Proceedings of the Human Computer Interaction Status Conference 2003*.

M.A. Walker, S. Whittaker, A. Stent, P. Maloor, J.D. Moore, M. Johnston, and G. Vasireddy. 2002. Speechplans: Generating evaluative responses in spoken dialogue. In *Proceedings of INLG 2002*.

Michael White. 2004. Reining in CCG chart realization. In *Proceedings of INLG 2004*.

Michael White. 2005. Efficient realization of coordinate structures in Combinatory Categorial Grammar. *Research on Language and Computation*. To appear.