MOG 2010

3rd Workshop on Multimodal Output Generation

PROCEEDINGS OF THE 3RD WORKSHOP ON
MULTIMODAL OUTPUT GENERATION

Trinity College Dublin, 6 July 2010

**Ielka van der Sluis, Kirsten Bergmann,
Charlotte van Hooijdonk and Mariët Theune (eds.)**

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Sluis, van der. I., Bergmann, K., Hooijdonk, C. van, Theune, M.

MOG 2010
Proceedings of the 3rd Workshop on Multimodal Output Generation
I. van der Sluis, K. Bergmann, C. van Hooijdonk, M. Theune (Eds.)
Trinity College Dublin, Ireland
6 July 2010

ISSN 0929–0672
CTIT Workshop Proceedings Series WP2010-02

trefwoorden: multimodality, natural language generation, multimodal generation, modality choice, human modalities.

Book orders:
Ms. C. Bijron
University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
Human Media Interaction
P.O. Box 217
NL 7500 AE Enschede
tel: +31 53 4893740
fax: +31 53 4893503
Email: c.g.bijron@ewi.utwente.nl

Druk- en bindwerk: Ipskamp Drukkers, Enschede

Preface

It is a pleasure for us to welcome you at Trinity College Dublin for the 3rd Workshop on Multimodal Output Generation (MOG 2010). Work on multimodal output generation tends to be scattered across various events, so one of our objectives in organising MOG 2010 is to bring this work together in one workshop. Another objective is to bring researchers working in different fields together to establish common ground and identify future research needs in multimodal output generation. We believe the programme of MOG 2010 meets these objectives, as it presents a wide variety of work offering different perspectives on multimodal generation, while there is also the opportunity to meet colleagues, exchange ideas and explore possible collaboration.

We are very pleased to welcome two invited speakers. Paul Piwek, from the Open University at Milton Keynes, UK, will argue for a change of emphasis in the generation of multimodal referring acts. In particular, he will talk about the consideration of two issues: first, neutral and intense indicating as two varieties of indicating, and second, the incorporation of pointing gestures into existing work on the generation of referring expressions. Regarding the latter Paul will present a novel account of the circumstances under which speakers choose to point that directly links salience with pointing. Gavin Doherty, from Trinity College Dublin at Dublin, Ireland, will talk about the nature of design problems of interactive computer systems and multimodal output. He will argue for the participation of end users in collaboration with technology developers and domain experts to face these problems. For the sake of illustration, Gavin will present work from the area of mental health care which makes extensive use of collaborative design methods.

This volume brings together the abstracts provided by our invited speakers and the papers presented at the MOG workshop. Five papers contribute to the challenge of multimodal output generation from different perspectives and are briefly introduced in the following.

Éric Charton, Michel Gagnon, and Benoît Ozell present preliminary results from a software application dedicated to multimodal interactive language learning. They investigate the problem of transition from a textual content to a graphical representation. The proposed system produces all syntactically valid sentences from a bag of words, and groups these sentences by their meaning to produce animations.

Michael Kriegel, Mei Yii Lim, Ruth Aylett, Karin Leichtenstern, Lynne Hall, and Paola Rizzo contribute with a paper on multimodal interaction in a collaborative role-play game. The game characters use speech and gestures for culture-specific communication. An assistive agent is used to enhance the user's perception of the characters' behaviour. Kriegel et al. report on an evaluation of the system and its interaction technology.

Kris Lohmann, Matthias Kerzel and Christopher Habel propose the use of tactile maps as a means to communicate spatial knowledge for visually impaired people. They present an approach towards a verbally assisting virtual-environment tactile map, which provides a multimodal map, computing situated verbal assistance by categorising the user's exploration movements in semantic categories.

Ian O'Neill, Philip Hanna, Darryl Stewart, and Xiwu Gu present a framework for the development of spoken and multimodal dialogue systems based on a dialogue act hierarchy. In their contribution O'Neill et al. focus on the means by which output modalities are selected dependent on a particular modality in a given system configuration as well as on the user's modality preference, while avoiding information overload.

Herwin van Welbergen, Dennis Reidsma, and Job Zwiers contribute with a paper on action planning for the generation of speech and gestures for virtual agents. Their approach applies a direct revision of bodily behaviour based upon short term prediction combined with corrective adjustments of already ongoing behaviour. This leads to a flexible planning approach of multimodal behaviour.

In addition to the above-mentioned paper presentations, the MOG 2010 workshop features two further presentations of work in progress. Margaret Mitchell will report on her work with Kees van Deemter and Ehud Reiter on natural reference to objects in a visual domain. Sergio Di Sano will present work on interactional and multimodal reference construction in children and adults.

Thanks are due to the programme committee members, to our guest speakers and the authors of the submitted papers.

MOG 2010 is endorsed by SIGGEN (ACL Special Interest Group on Generation) and has been made possible by financial support from the Science Foundation Ireland. The Cognitive Science Society sponsored a prize for the best student paper. Trinity College Dublin provided administrative assistance as wel as the venue for the workshop and the German Society for Cognitive Science provided the domain for our website (http://www.mog-workshop.org/). Finally, the research institute CTIT (Centre of Telematics and Information Technology) of the University of Twente kindly gave us permission to publish the proceedings of MOG 2010 in the CTIT Proceedings series. We are grateful to all these supporting organizations.

The organizers of this workshop,

Ielka van der Sluis, Kirsten Bergmann, Charlotte van Hooijdonk and Mariët Theune June 2010

TRINITY COLLEGE DUBLIN
COLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH
THE UNIVERSITY OF DUBLIN

Universität Bielefeld

vrije Universiteit amsterdam

UNIVERSITY OF TWENTE.

MOG 2010 Organizing Committee

Ielka van der Sluis	Trinity College Dublin, Ireland
Kirsten Bergmann	Bielefeld University, Germany
Charlotte van Hooijdonk	VU University Amsterdam, The Netherlands
Mariët Theune	University of Twente, The Netherlands

MOG 2010 Programme Committee

Elisabeth André	University of Augsburg, Germany
Ruth Aylett	Heriot-Watt University, UK
Ellen G. Bard	University of Edinburgh, UK
John Bateman	University of Bremen, Germany
Christian Becker-Asano	ATR (IRC), Japan
Timothy Bickmore	Northeastern University, USA
Harry Bunt	Tilburg University, The Netherlands
Justine Cassell	Northwestern University, USA
Kees van Deemter	University of Aberdeen, UK
David DeVault	USC ICT, USA
Mary Ellen Foster	Heriot-Watt University, UK
Markus Guhe	University of Edinburgh, UK
Dirk Heylen	University of Twente, The Netherlands
Gareth Jones	Dublin City University, Ireland
Michael Kipp	DFKI, Germany
Stefan Kopp	Bielefeld University, Germany
Emiel Krahmer	Tilburg University, The Netherlands
Theo van Leeuwen	University of Technology Sydney, Australia
James Lester	North Carolina State University, USA
Saturnino Luz	Trinity College Dublin, Ireland
Fons Maes	Tilburg University, The Netherlands
Mark Maybury	MITRE, USA
Paul Mc Kevitt	University of Ulster, UK
Mike McTear	University of Ulster, UK
Louis-Philippe Morency	USC ICT, USA
Radoslaw Niewiadomski	TELECOM ParisTech, France
Jon Oberlander	University of Edinburgh, UK
Cécile Paris	CSIRO ICT Centre, Australia
Paul Piwek	The Open University, UK
Ehud Reiter	University of Aberdeen, UK
Jan de Ruiter	Bielefeld University, Germany
Thomas Rist	FH Augsburg, Germany
Zsofia Ruttkay	Moholy-Nagy University of Art and Design, Hungary
Matthew Stone	Rutgers, USA
Kristina Striegnitz	Union College, USA
Marc Swerts	University of Tilburg, The Netherlands
David Traum	USC ICT, USA
Marilyn Walker	University of Sheffield, UK
Sandra Williams	The Open University, UK

Sponsors

Science Foundation Ireland, http://www.sfi.ie/

Cognitive Science Society, http://cognitivesciencesociety.org/

The German Society for Cognitive Science, http://www.gk-ev.de/

The Centre for Telematics and Information Technology, University of Twente, http://www.ctit.utwente.nl/

Endorsed by SIGGEN

ACL Special Interest Group on Generation, http://www.siggen.org/

Contents

Invited Speakers

Regular Speakers

Aspects of Indicating in Multimodal Generation: Intensity and Salience

Paul Piwek
Centre for Research in Computing
The Open University, United Kingdom
p.piwek@open.ac.uk

Abstract

Most extant models of verbal reference to objects in a shared domain of conversation, specifically in the field of Natural Language Generation, focus on description: the use of symbolic means to uniquely identify a referent. Generation of multimodal referring acts requires a change in focus. In particular, demonstratives combined with pointing gestures are primarily a form of indicating. In my talk, I will discuss two issues which this change in emphasis brings with it. Firstly, I will examine the evidence for two varieties indicating: neutral and intense indicating, which, I will argue, are associated with the distal and proximal form of demonstrative noun phrases, respectively. Secondly, I will examine how pointing gestures can be incorporated into existing work on the generation of referring expressions. I will show that in order to add pointing, the notion of salience needs to play a pivotal role. After distinguishing two opposing approaches: salience-first and salience-last accounts, I will discuss how a salience-first account nicely meshes with a range of existing empirical findings on multimodal reference. A novel account of the circumstances under which speakers choose to point is described that directly links salience with pointing. The account is placed within a multi-dimensional model of salience for multimodal reference.

References

Piwek, P. (2009). Salience in the Generation of Multimodal Referring Acts. In *Proceedings of the 2009 International Conference on Multimodal Interfaces (ICMI-MLMI)*, pages 207–210, Cambridge, MA. ACM Press.

Piwek, P., Beun, R., and Cremers, A. (2008). 'Proximal' and 'Distal' in language and cognition: evidence from deictic demonstratives in Dutch. *Journal of Pragmatics*, 40(4):694–718.

Collaborative Design of Multimodal Output

Gavin Doherty
Trinity College Dublin
Dublin, Ireland
Gavin.Doherty@scss.tcd.ie

Abstract

A major theme of Human-Computer Interaction (HCI) research has been on facilitating user participation in the design of interactive computer systems, including the use of participatory design processes in which users form part of the design team. A further development is the emergence of a range of informative systems in which the content or information being delivered is generated by end-users or domain experts. This content may be delivered in a multimodal fashion, and hence we must consider the future of multi-modal output generation (MOG) technologies to be one in which the final design depends on the technology developers, domain experts and end users.

To illustrate, I discuss work in the area of technology support for mental health care, where we have made extensive use of collaborative design methods. The systems developed made use of virtual characters in 3D computer games, video (including animations), mobile phones, Internet charts, and (importantly) paper. The model used was one in which development of the platform was separated from development of content, but each was a collaborative process, one led by the technology developers, the other by domain experts. The user experience emerges from the combination of the two, but the focus of each design effort is different. While I reflect on the potential use of MOG in this area, the focus of the talk will be on the nature of the design problem facing those trying to develop produce informative, affective and engaging experiences using multimodal output, and how this may impact on the future of MOG.

A Preprocessing System to Include Imaginative Animations According to Text in Educational Applications *

Éric Charton, Michel Gagnon, Benoît Ozell
École Polytechnique de Montréal
2900 boulevard Édouard-Montpetit, Montréal, QC H3T 1J4, Canada
{eric.charton|michel.gagnon|benoit.ozell}@polymtl.ca

Abstract

The GITAN project aims at providing a general engine to produce animations from text. Making use of computing technologies to improve the quality and reliability of services provided in educational context is one of the objectives of this project. Many technological challenges must be solved in order to achieve such a project goal. In this paper, we present an investigation on the limitations of text to graphics engines regarding imaginative sentences. We then comment preliminary results of an algorithm used to allow preprocessing of animations according to a text for a software application dedicated to multi-modal interactive language learning.

Keywords: Generation of animations

1 Introduction

In a long term perspective, The GITAN project[1] (Grammar for Interpretation of Text and ANimations), which started at the end of 2009, aims to solve the problem of transition from a textual content to a graphical representation. Discovering those mechanisms implies exploration of intermediate steps. As this project is generic and not domain dependent, we specifically need to explore the limits of computability of a graphic animation, regarding to a sentence, into a wide acceptance. In particular, we need to investigate the limits of existing graphic rendering techniques, regarding the potential complexity of semantic meaning obtained through a free, on the fly, sentence acquisition.

To illustrate this, we present preliminary results of a system dedicated to build a language learning software application. This system involves the capacity of a student to produce a semantically and syntactically acceptable sentence using a limited bag of words defined by a teacher, while observing a graphical animation of the sentence. The difficult aspect of this work is that the learning software has to display an animation for any syntactically correct sentence constructed from the bag of words. The idea is to allow the student to compare the animation that results from his own arrangement of words with the one that conforms to the visual representation of the target sentence chosen by the teacher (see figure 1). An intuitive advantage of such a tool is the capacity given to the student to understand instantly, with the help of animations, misinterpretations and confusions resulting in some sentence constructions. From a theoretical perspective, this application is an opportunity to investigate specific cases appearing in animation generation, driven by a non-constrained natural language.

This paper is organized as follows. First, we describe the proposed application, and investigate the theoretical challenge arising from its specificities. Then, we describe the previous attempts made in the research field of text to animation systems, and put them into perspective with

[*] This work is granted by Unima Inc and Prompt Québec.

[1] www.groupes.polymtl.ca/gitan/

the specific problem encountered with open sentences generated from a bag of words. In the fourth section, we present a system and its algorithms whose purpose is to anticipate the types of sentences that a student can produce from a bag of words and limit the amount of animations to be preprocessed. Then we present the results of an experiment where we produce a delimited set of sentences extrapolated from a bag of words and evaluate how those sets can be used to preprocess animations. We conclude with future work.

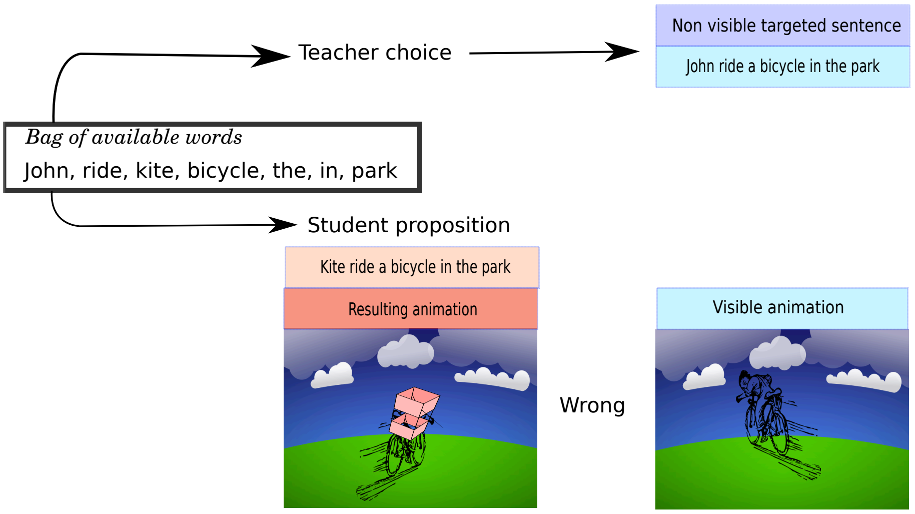

Teacher choice
Non visible targeted sentence
John ride a bicycle in the park
Bag of available words
John, ride, kite, bicycle, the, in, park
Student proposition
Kite ride a bicycle in the park
Resulting animation
Wrong
Visible animation

Figure 1: Synaptic representation of proposed application

2 Application principle and theoretical view

Chomsky investigated one aspect of nonsensical meaning in sentence construction with his famous sentence *Colorless green ideas sleep furiously.*[2] This is an example of a sentence with correct grammar (logical form) but potential nonsensical content. Our application is a typical case of the need for acceptance and interpretation of potential nonsensical sentences. It has been shown by Pereira (2000) that such a sentence, with a suitably constrained statistical model, even a simple one, can meet Chomsky's particular challenge. Under this perspective, this can be viewed as a metaphoric problem, but not only: it can also deal with unnatural communication intent, relevant to pure imagination. This problem investigated by the linguistic theory as the transformation mechanism of *conceptual-intention* into a linear sentence is not solved yet (Hauser et al. (2002); Jackendoff and Pinker (2005)).

In the generic field of graphical representation, Tversky et al. (2002) claim that correspondences between mental and graphical representations *suggest cognitive correspondences between mental spaces and real ones.* In the perspective of transforming a *conceptual-intention* into a visual representation, Johnson-Laird (1998) considers that visual representation of mental models *cannot be reduced to propositional representations*[3] *[as] both are high-level representations necessary to explain thinking.*[4] Johnson-Laird considers also that *mental models themselves may contain elements that cannot be visualized.* According to this, it appears in the perspective of a text to animation computer application, that the correspondences between semantic abstractions extracted from free text and visual representations are not always relevant to a simple sentence parse and rendering in a graphic engine. In pictorial arts, the correspondences for mental representations permitted by imagination, are obtained by a cognitive transformation of physical law, natural spaces and transgression of common sense to adapt an animation or a static image to the mental

[2] In *Syntactic Structures*, Mouton & Co, 1957.

[3] Defined by Johnson-Laird (1998), page 442 as *representations of propositions in a mental language.*

[4] Johnson-Laird (1998) page 460.

representation. Finally we can consider that animated results of those specific transformations are equivalent to the creative ones observed in artistic and entertainment applications like computer games, movies, cartoons. This particular aspect of natural language driven image generation and the role of physical limitations has been investigated by Adorni et al. (1984) who consider that such a cognitive transformation should be relevant to a computer AI problem.

2.1 Three cases of syntactically correct nonsensical sentences

To illustrate this, let us consider a bag of words, including the 10 following terms: ***{Jack, rides, with, bicycle, park, the, kite, runs, in, his}***. According to the rules of our application, the learner is allowed to build any sentence including a subset of those words. Those sentences can be for example *Jack rides his bicycle in the park. The kite runs in the park.* But they can also be *The bicycle rides Jack. The kite rides the bicycle.* If we mentally imagine the scenes expressed by these four sentences, we intuitively know that each one can be animated. Some of them violate common sense or physical laws, but can still be animated. For example, we can produce an animation representing a *bicycle riding its owner*, and thus revealing to the student a misinterpretation of relations between dependencies in a sentence. This is a **position case**. We will see that such semantic cases can be represented by a graphic engine.

Another case could be a sentence based on **action** verbs. If we consider a bag of words containing ***{cat, eats, on, the, chair, in, his}***, a teacher will be able to define a target sentence like *The cat eats on the chair.* But the *eating* verb can have various possible representations, according to the order of words, and can be organized in sentences like *The chair eats the cat. The chair eats on the cat.* Only a mental work can solve the problem posed by the visualization of these sentences, and this work implies attribution of an imaginative animation sequence describing a *chair eating.* We can imagine a metaphoric application using a classical graphic engine, where a *cat disappears when it is touching the chair.* But this is clearly a lack of realism, difficult to accept in our education application.

A third case will involve **transformations**: if we consider now a bag of words containing ***{prince, transforms, into, the, castle, in, his, toad, himself, a}***. The target sentence could be *The prince transforms himself into a toad.* But it becomes difficult to integrate in a graphic engine a transformation function compatible with constructions like *The toad transforms himself into a prince. The toad transforms the castle into a prince.* If we consider all the possible action verbs and all the objects which can receive the faculty to do the concerned action, we obtain a very difficult problem to compute, relevant to an AI system, like predicted by Adorni et al. (1984).

From the previous examples, we can divide this representation problem in three families of cases: a **position case** (*The kite rides the bicycle*), an **action case** (*The chair eats the cat*) and a **transformation case** (*The toad transforms the castle into a prince*).

3 Existing systems and previous work

Many experiments have been previously done in the field of text to animation processing. In this section we examine some of the previously described existing systems and investigate their capacities regarding our three text to animation semantic cases.

3.1 Capacities of existing animation engines

In Dupuy et al. (2001), a prototype of a system dedicated to visualization and animation of 3D scenes from car accident written reports written is described. The semantic analysis of the CarSim processing chain is an information extraction task that consists in filling a template corresponding to the formal accident description: the template constrained choices limit the system to a very specific domain, with no possible implication in our application context.

Another system, WordsEye, is presented in Coyne and Sproat (2001). The goal of WordsEye is to provide a blank slate where the user can paint a picture with words: the description may consist not only of spatial relations, but also actions performed by objects in the scene. The graphic engine principle of WordsEye, like most of graphic engines, is able to treat the **position case** like *A chair is on the cat* [5] but because of its static nature, offers no possibilities to treat neither the **action cases** nor the **transformation cases**. The authors of WordsEye consider that *it is infeasible to fully capture the semantic content of language in graphics.*[6]

In an academic context, the system e-Hon, presented by Sumi and Nagata (2006), uses animations to help children to understand a content. It provides storytelling in the form of animation and dialogue translated from original text. The text can be a free on-the-fly input from a user. This system operates in a closed semantic field[7] but uses an IA engine to try to solve most of the semantic cases. Authors indicate that some limitations have been applied: firstly, *articulations of animations are used only for verbs with clear actions*; secondly, this system constrains sentences *using common sense knowledge in real time* (using ontological knowledge described in Liu and Singh (2004)). It is interesting, regarding our targeted application, to observe that a system dealing with potentially highly imaginative interactions from children needs to restrict its display with a *common sense resource.*

Some applications like Confucius (Ma (2006)) are more ambitious. The animation engine of Confucius accepts a semantic representation and uses visual knowledge to generate 3D animations. This work includes an important study of visual semantics and ontology of eventive verbs. But this ontology is used to constrain the representation[8] to common sense[9] through a concept called *visual valency.* According to this, Confucius' techniques cannot fit with the studied cases of our application.

Finally, the main characteristics of most of those existing systems are that they operate in a closed semantic field, according to common sense and respecting physical laws. One of them (WordsEyes) can represent any spatial position for any object in a scene. But none of those existing systems has the capacity to produce realistic representations for usage of action verbs non-conform to common sense included in a syntactically correct sentence and none of them can manipulate a transformation of any concept to another. This establishes a clear limitation of actual technologies available for the text to animation task when they are used in an open semantic field.

3.2 Semantic parsing and generation from bags

Besides, as discussed earlier, our application may meet situations where the animation does not respect physical laws and common sense. We have shown that there are many cases where it is not possible to simply parse an input sentence from the user and produce on the fly a semantic specification and give it to an animation engine. If the grammar does not contain common sense or physical laws, the semantic content of a syntactically correct sentence can correspond to a mental representation that does not respect common sense and that is not compatible with any actual existing animation engine. According to this, in our application context, one possible way is to try enumerating all the possible sentences that a bag of words can generate and to see if there is a way to cluster those sentences of similar meaning into sets small enough to be compatible with a preprocessing animation task. This is a typical sentence realization task, actively investigated in Natural Language Generation (NLG) (see Reiter and Dale (2000)). Text generators using statistical models without consideration to semantics exists. Langkilde and Knight (1998) present a text generator would take on the responsibility of finding an appropriate linguistic realization for an underspecified semantic input. In Belz (2005), an alternative method for sentence realization very close to our needs uses language models to control formation of sentences. However, our problem is specific and difficult to solve with a NLG module as we need to produce all possible sentences from a bag of words to preprocess animations, and not only a unique well-formed sentence, corresponding to a *conceptual-intention.* This specific aspect of exhaustive generation from bags of words has been first investigated by Yngve (1961). In this work, a generative grammar is combined to a

[5] Numerous examples are available on the website at www.wordseye.com.

[6] In Coyne and Sproat (2001) page 496.

[7] 18 characters, 67 behaviors, and 31 backgrounds.

[8] Ma (2006) page 109.

[9] *Language visualization requires lexicalcommon sense knowledge such as default instruments (or themes) of action verbs, functional information and usage of nouns.* Ma (2006) page 116.

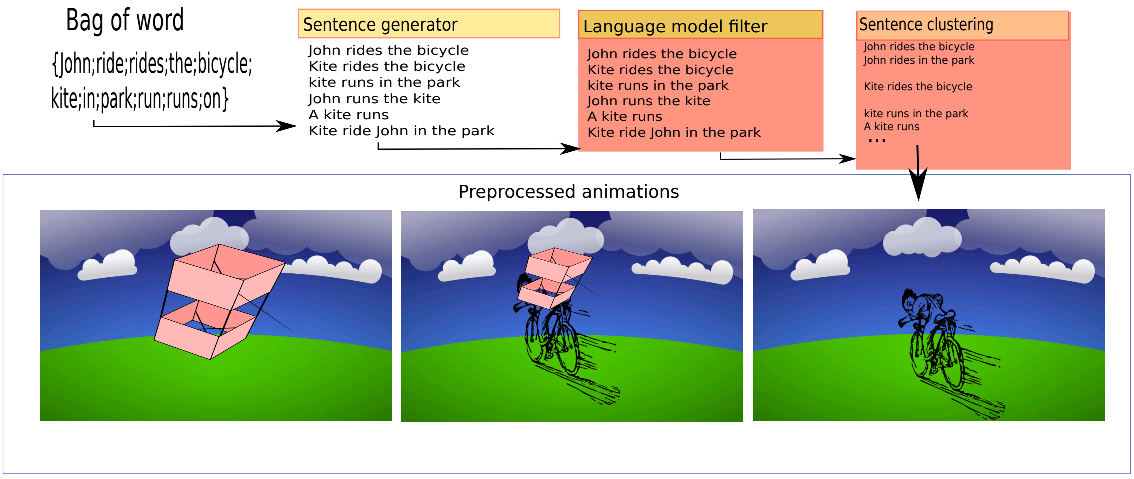


Figure 2: Architecture of the system and its successive algorithms

combinatorial random sentence generator applied to a bag of words. Most of the output sentences were quite grammatical, though nonsensical. Recently, Gali and Venkatapathy (2009) explored a derived work where models consider a bag of words with unlabeled dependency relations as input and apply simple n-gram language modeling techniques to get a well-formed sentence.

4 Proposed system

The given problem could be solved through enumeration of all the syntactically valid sentences that may potentially be produced for a given bag of words, without consideration to semantics, common sense or physical laws, followed by a clustering of those sentences into groups according to their meaning similarity. First, our system takes as input a bag of words and produces all syntactically valid sentences by means of a simple English rule-based sentence generator. Then, it uses a language model (as described in Song and Croft (1999)) to select, among the group of word combinations, only sentences that are valid according to a modeled language. Finally, a clustering algorithm groups these sentences by using a meaning similarity measure. At the end, we obtain for a given bag of words a restricted list of sentences, clustered by senses. We can produce for each cluster of sentences a unique animation. This unique animation is displayed when the student makes an attempt of sentence construction.

4.1 Sentence generator (SG)

The sentence generator (SG) is built with a limited set of flexible generative grammar rules implemented in Prolog. Those rules, which cover verbal phrases, noun phrases and prepositional phrases, allow the generation of sentences from a bag of words. The category of the words contained in the bag is also considered and added as a label to each word contained in the generated sentence. For example, the rules for verb phrases are the following ones:

```
vp(Features,BagIn,BagOut)-->
        lex(v,Features,BagIn,BagOut).

vp(Features,BagIn,BagOut,)-->
        lex(v,Features,BagIn,Bag1),
        np(_,Bag1,BagOut,).

vp(Features,BagIn,BagOut,)-->
        lex(v,Features,BagIn,Bag1),
        pp(Bag1,BagOut).
```

```
vp(Features,BagIn,BagOut)-->
        lex(v,Features,BagIn,Bag1),
        np(_,Bag1,Bag2),
        pp(Bag2,BagOut).
```

Note that the `lex` predicate refers to the lexical entry that specifies a word to be inserted in the sentence, whereas `np` and `pp` refer respectively to noun phrase and preposition phrase rules that will be recursively applied. We can see that the verb phrase rules cover about all verb arities without constraints. As we will see later, it is the language model that will constrain the generative expressivity. The rules also take as parameters the bag of words and the sequence of words forming the sentence currently generated. At each step in the execution of a rule, words are extracted from the bag of words and appended at the end of the sequence.

The used word categories are described by a standard morphosyntactic tag from Penn-Tree bank tag-set[10] like noun (NN), proper name (NP), verb (VBZ), conjunction (IN), article (DT), personal pronoun (PP). SG generates a sentence by combining phrases. For example, a sentence can be produced by combining a verb phrase with a noun phrase at subject position, as expressed by the following grammar rule (note that there are agreement constraints for person and number, and another constraint specifying that the verb phrase must be in declarative mode):

```
s(BagIn,BagOut,SeqIn,SeqOut)-->
      np(pers~P..number~N,BagIn,Bag1,SeqIn,Seq1),
      vp(mode~dec..pers~P..number~N,Bag1,BagOut,Seq1,SeqOut).
```

Taking as input the bag of words $\{the, is, a, Jack, bicycle, kite, park, in, rides, runs\}$, the system generates the following sentences:

```
Jack/NP rides/VBZ the/DT bicycle/NN
Jack/NP runs/VB the/DT bicycle/NN
Jack/NP runs/VB the/DT kite/NN
the/DT bicycle/NN rides/VBZ Jack/NP
the/DT bicycle/NN rides/VBZ a/DT kite/NN
the/DT bicycle/NN runs/VB Jack/NP
...
```

The flexibility of this very simple generative grammar is a deliberate choice to avoid the risk of non-generation of a valid sentence. In case of a non-valid sentence, the next module of our system is a language model filter that has been trained with a big corpus and achieves a final filtering that will remove all non-valid sentences.

4.2 Language model Filter (LMF)

The language model (LM) is trained from a corpus which domain is related to the targeted application. For the sample application presented in this paper (teaching English language), we used the *Simple Wikipedia* corpus.[11] This corpus uses simple English lexicon and grammar and is well-suited for our application needs. The language model is trained with the SRILM toolkit.[12] Each sentence proposed by the *Sentence Generator* is filtered by using an estimation of its probability, regarding LM. In our application, SRILM produces N-Gram language models of words.[13] With such a model, the probability $P(w_1, \ldots, w_n)$ to observe a sentence composed of words $w_1 .... w_n$ in the modeled corpus is estimated by the product of probabilities of the individual appearance of words contained in sequence $P(w_{1,n}) \approx P(w_1)P(w_2)...P(w_n)$. To obtain a more robust system, bi-Gram or tri-Gram models applied to a sequence of n words are adopted: $P(w_1, \ldots, w_n) \approx P(w_1)P(w_2|w_1)P(w_3|w_{1,2})...P(w_n|w_{n-2,n-1})$. In our application, we use a bi-Gram model, which can be represented by the following example ($< s >$ indicates beginning

[10] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/Penn-Treebank-Tagset.pdf

[11] See simple.wikipedia.org, and downloadable version on http://download.wikipedia.org/simplewiki/.

[12] Available on http://www.speech.sri.com/projects/srilm/.

[13] An n-gram is a subsequence of n items from a given sequence. The items can be phonemes, syllables, letters, words or base pairs, according to the application.

of sentence):

$$P(Jack, rides, the, bicycle) \approx P(Jack| < s >)P(rides|Jack)P(the|rides)P(bicycle|the)$$

For each sentence generated by SG, we estimate its probability of appearance. The non-existence of a bi-Gram sequence means a null probability for the complete observation sequence and rejection of generated sentence. It is also possible to define a threshold constant to reject sentences with low probability estimation.

4.3 Clustering algorithm (CA)

The clustering algorithm uses the chunking faculty of the Tree-tagger morphosyntactic shallow parser.[14] Chunking is an analysis of a sentence that identifies the constituents (noun phrases, verb phrases, etc.), but does not specify neither their internal structure, nor their role in the main sentence.

Considering the list l of n sentences $s_1 \dots s_n$ kept by LMF, we generate a function *f_similarity* for the first sentence s_1 of l. This function contains, for each phrase chunk, a description of its nature and its position in s_1. Each phrase chunk is associated with its lexical content, with consideration to similarities (i.e. two similar verbs will be considered as unique). Next, we apply *f_similarity* to the remaining sentences $s_2 \dots s_n$. All sentences for which the function returned value 1 are selected to form a cluster together with sentence s_1. Finally we remove all the clustered sentences from l and iterate CA until l is empty. For the example *[Jack/NC] [rides/VC] [the bicycle/NC]* the similarity clustering function will be:

```
 f_similarity(sentence) = {
   if (sentence={1:NC{Jack};2:VC{rides;run};3:NC{bicycle}})) return(1)
else return(0) }
```

And clustering will be :

```
[Jack/NC] [rides/VC] [a bicycle/NC]
[Jack/NC] [runs/VC] [the bicycle/NC]
[Jack/NC] [rides/VC] [the bicycle/NC]
```

5 Experiments and preliminary results

In the preliminary experiments of our system, we used 10 bags of 10 words. Bags of words come from exercises included in an learning English student's book.[15] Those exercises include, for a given topic (i.e. *Talking about abilities*), a set of target sentences and a suggested vocabulary (i.e. *play, guitar, dance, swim, etc*).

Words	Generated sentences (SG)	Correct sentences (LMF)	Sentence clusters (CA)
6	25	23	7
10	460	280	20

Table 1: Evaluation of groups of sentences generated from a bag of words

We use 6 and 10 words from the bag and apply SG, LMF and CA. We count sentences generated in SG, sentences kept in LMF, and clusters returned by CA. Table 1 gives the arithmetic mean value of the results for each step of the test. This preliminary experiment confirms that for a given bag of words, it is possible to generate a limited set of semantics groups, compatible with a not expensive video preprocessing task. With a bag of 10 words, only 20 clusters are obtained, meaning only 20 animations have to be produced based on the limited set of objects delimited by the bag of words.

[14] The Tree-tagger is a tool for annotating text with part-of-speech and lemma information. It can also be used as a chunker. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[15] *Go For It! English for Chinese students*, series published by Thomson Learning.

Those preliminary results are sufficient to build an application prototype. With such results, our system can be used to preprocess and help to evaluate amount and specificity of potential animations according to a bag of words used to produce sentences. Our method allows to select, for a given bag of words, a limited set of semantic groups of sentences. The system can be used as a production tool to preprocess video in a text-to-animation multimodal application. It can also be used as a component of text-to-animation application software to evaluate its semantic field and produces automatically test sentences for evaluation purposes.

6 Conclusion

We presented an original component to support text to animation applications. The originality of this system is that it is not restricted to valid semantic productions that do not violate common sense and physical laws. This proposition investigates the specific situation of imaginative text to image applications. We showed that a generative grammar combined with statistical methods can extract a limited amount of potential sentences from a given bag of words. The advantage of such a structure is its ability to preprocess text to animation sequences in an open context application, with a low amount of misrepresentations of animated sequences regarding to text sense. The next step of our work is to try to introduce in our architecture a real-time text to image generator that accepts, in restricted semantic domains, scenes that do not respect common sense. This will be an attempt to evaluate the capacities of a system to elaborate imaginative-like text to animation system.

References

Adorni, G., Di Manzo, M., and Giunchiglia, F. (1984). Natural language driven image generation. In *Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics*, pages 495–500. Association for Computational Linguistics.

Belz, A. (2005). Statistical generation: Three methods compared and evaluated. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG'05)*, pages 15–23.

Coyne, B. and Sproat, R. (2001). WordsEye: An automatic text-to-scene conversion system. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 487–496. ACM New York, NY, USA.

Dupuy, S., Egges, A., Legendre, V., and Nugues, P. (2001). Generating a 3D simulation of a car accident from a written description in natural language. *Proceedings of the Workshop on Temporal and Spatial Information Processing*, pages 1–8.

Gali, K. and Venkatapathy, S. (2009). Sentence realisation from bag of words with dependency constraints. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 19–24. Association for Computational Linguistics.

Hauser, M., Chomsky, N., and Fitch, W. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569–1579.

Jackendoff, R. and Pinker, S. (2005). The nature of the language faculty and its implications for evolution of language (Reply to Fitch, Hauser, and Chomsky). *Cognition*, 97(2):211–225.

Johnson-Laird, P. (1998). Imagery, visualization, and thinking. In Hochberg, J., editor, *Perception and Cognition at Century's End*, pages 441–467. Academic Press.

Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and*

17th International Conference on Computational Linguistics, pages 704–710, Morristown, NJ, USA. Association for Computational Linguistics.

Liu, H. and Singh, P. (2004). Commonsense reasoning in and over natural language. In Negoita, M., Howlett, R., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 293–306. Springer.

Ma, M. (2006). *Automatic conversion of natural language to 3D animation.* PhD thesis, University of Ulster, Faculty of Engineering.

Pereira, F. (2000). Formal Grammar and Information Theory: Together Again? *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 358(1769):1239 – 1253.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems.* Cambridge University Press.

Song, F. and Croft, W. B. (1999). A General Language Model for Information Retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321.

Sumi, K. and Nagata, M. (2006). Animated storytelling system via text. In *ACM International Conference Proceeding Series; Vol. 266.* ACM New York, NY, USA.

Tversky, B., Morrison, J., and Betrancourt, M. (2002). Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262.

Yngve, V. (1961). Random generation of English sentences. In *International Conference on Machine Translation of Languages and Applied Language Analysis*, pages 5–8.

A Case Study In Multimodal Interaction Design For Autonomous Game Characters

Michael Kriegel, Mei Yii Lim, Ruth Aylett
School of Mathematical and Computer Sciences
Heriot-Watt University
{mk95,M.Lim}@hw.ac.uk,{ruth}@macs.hw.ac.uk

Karin Leichtenstern
Lehrstuhl für Multimedia-Konzepte und Anwendungen
Universitätsstr. 6a
86159 Augsburg
leichtenstern@informatik.uni-augsburg.de

Lynne Hall
School of Computing and Technology,
University of Sunderland
lynne.hall@sunderland.ac.uk

Paola Rizzo
Interagens s.r.l.
c/o ITech, Via G. Peroni 444, 00131, Rome, Italy
p.rizzo@interagens.com

Abstract

This paper presents our experience of designing the educational collaborative role-play game ORIENT with a special focus on the multimodal interaction used in the game. The idea behind ORIENT is to try increasing a user's intercultural empathy through role-play set in a science fiction narrative in which the users have to befriend an alien race, the Sprytes. The Sprytes are virtual characters that communicate with the users through a mix of gestures and natural language. We explain how the choice and design of those output modalities was driven by choice of interaction technology. We also show how the user's perception of the Spryte's behaviour could be enhanced through the inclusion of an assistive agent, the ORACLE and report on a small scale evaluation of the system and its interaction technology.

Keywords: role-play, novel interaction devices, whole body interaction

1 Introduction

The EU FP6 project eCIRCUS[1] aimed to apply innovative technology to the context of emotional and social learning. This paper is about one of the showcases produced during the project: ORIENT. In the case of ORIENT, the application design started with a stated learning goal: to improve the integration of refugee/immigrant children in schools through technology assisted role play. This type of acculturation is a two-way process in which both the incoming group and the host group have to negotiate a common understanding. An educational application could therefore target either of those groups. In our case the more obvious choice was to focus on the host group since this is the group with less intercultural experiences and to foster intercultural

[1] http://www.e-circus.org/

sensitivity through the development of intercultural empathy. In other words we try to increase the responsibility and caring for people with different cultural backgrounds. This gave us the basic framework for a role playing application in which the users (i.e. learners) are outsiders in an unknown culture and interact with virtual characters that are members of that culture. The quests and challenges in the game are built around slowly getting accustomed to the alien culture, as theorized by Bennett's model of intercultural sensitivity (Bennett, 1993). For more information about the learning objectives within this application see (Kriegel et al., 2008)

We decided that this virtual culture should not be a replica of an existing human culture and opted instead for a completely fictional culture, which we eventually named Sprytes. By portraying a fictional culture, our application is more flexible and suitable for users from diverse backgrounds. Furthermore, it allows us to exaggerate cultural differences for dramatic and educational purposes. In the remainder of this paper we will first give an overview of ORIENT and then describe the considerations involved in designing the multimodal communication interface between the Sprytes and the users.

2 Overview of ORIENT

ORIENT was designed to be played by a team of 3 teenage users, each a member of a spaceship crew. Their mission takes them to a small planet called ORIENT, which is inhabited by an alien race, the lizard-like, humanoid and nature-loving Sprytes. The users' mission is to prevent a catastrophe - a meteorite strike on ORIENT - which the Sprytes are unaware of. The users can achieve this goal by first befriending the Sprytes and ultimately cooperating with them to save their planet. Through interaction with the Sprytes, ORIENT promotes cultural-awareness in the users, at the same time acts as a team building exercise where users play as a single entity rather than as individuals. All users have the same goal in the game although their roles and capabilities differ.

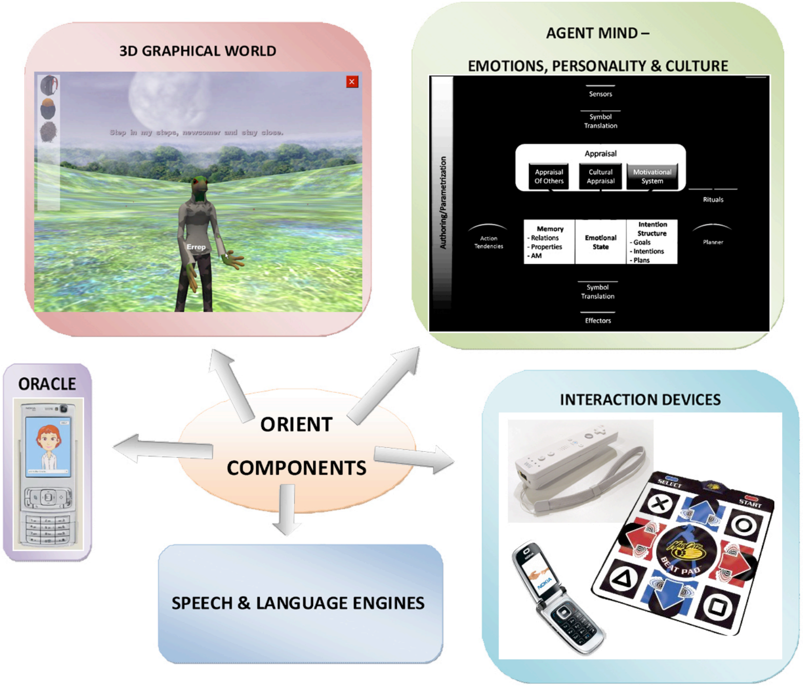
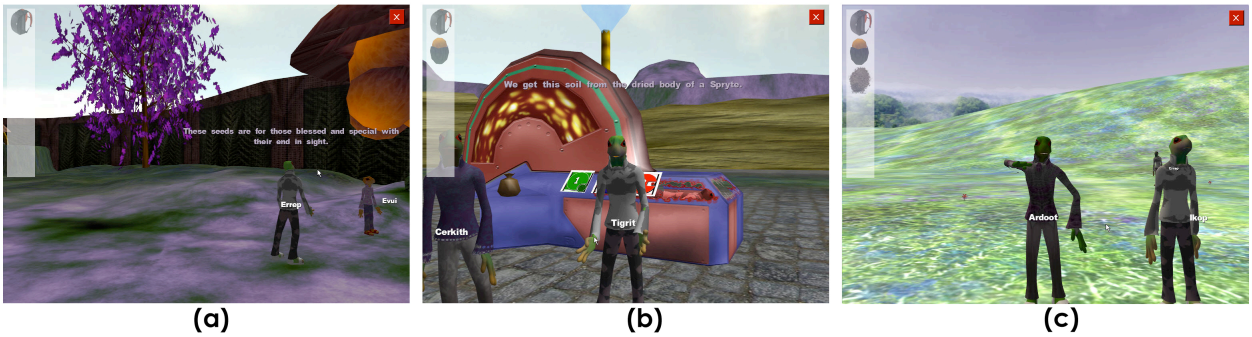


Figure 1: ORIENT system components

ORIENT consists of many components as shown in Figure 1. It has a virtual 3D graphical world where all users share a single first person perspective. In the implemented prototype version, users can explore 4 different locations of the Sprytes' world. In each of these locations, a different interaction scenario takes place. The users have an opportunity to witness the Sprytes' eating habits - eating only seedpods that dropped on the ground (Figure 2(a)), life cycles - recycling the dead (Figure 2(b)), educational styles, family formation and value system - trees are sacred (Figure 2(c)).


These seeds are for those blessed and special with their end in sight.

We get this soil from the dried body of a Spryte.

Figure 2: ORIENT scenarios

2.1 Spryte Culture

The Sprytes are autonomous affective characters driven by a mind architecture (Dias and Paiva, 2005; Lim et al., 2009b). These characters have drives, individual personalities, cultural values and are able to express emotions. The Sprytes culture has been defined based on a subset of Hofstede's dimensions (Hofstede, 1991). Hofstede defines culture in terms of dimensions such as Hierarchy, Identity, Uncertainty avoidance and Gender. The Sprytes' culture has a hierarchical organisation which depends highly on respect and age. They are also a collectivistic culture, which makes them compassionate with each other, and live in a group where the majority holds power. The Sprytes are highly traditional in their ways and view uncertainty as a threat but exceptions do exist in younger Sprytes. We designed the Sprytes to be genderless creatures, which eliminates the Gender Dimension. An extension to the mind architecture allows those cultural parameters to directly influence the agent's behaviour. A detailed description of this model can be found in (Mascarenhas et al., 2009).

3 Designing The Spryte Communication

An important distinguishing element between different cultures is communication. This includes for example factors such as gestures, facial expressions and language. During our design of ORIENT we also had to consider these factors for the Spryte culture. The fact that the Sprytes are different from us is a premise of ORIENT's narrative framework, emphasised by the way the Sprytes communicate.

3.1 Gestures

In order to make them interesting and different we made the Sprytes rely heavily on gestures in their communication. Sprytes use gestures instead of facial expressions to convey emotions. Additionally they use gestures like verbs to convey meaning. Ideally we would have liked the Sprytes to communicate only using gestures. However, the narrative framework consisted of a complex story in a world full of strange and unknown things and we found it infeasible to tell this story solely through gestures and without any use of language. Therefore we gave the Sprytes the additional ability to speak. Another reason for this decision lies within the cost and resources that would be required to build a huge gestural animation repertoire.

3.2 Speech and Natural Language

In ORIENT, dialogues are treated as symbolic speech acts by the mind architecture (Dias and Paiva, 2005; Lim et al., 2009b). When a Spryte speaks, the speech act is transformed into natural language subtitles by a language engine. The story explanation for the subtitles is the advanced language computer that our space travelling users carry with them. On the auditory channel, the subtitle is accompanied by an artificial incomprehensible gibberish language that is generated by a speech engine simultaneously. We used a customized text to speech system based on the Polish language to create an alien gibberish language for the Sprytes' speech. Whenever a Spryte speaks, the same utterance that is displayed as a subtitle is also used as input for the speech generator. The generated gibberish has no real semantics but it does contain special words for important objects and character names. Care has been taken to ensure that the language sounds varied enough. In the next section we are going to describe the user interaction in ORIENT in detail and explain the influence it had on the further refinement of the Spryte's communication style.

4 Input Modalities

4.1 Related Work

A large variety of interfaces have been proposed for role-play environments including desktop-based interfaces, mobile interfaces, augmented reality as well as novel forms of interaction based on the use of electronic toys, conversation with virtual characters or instrumented story environments. Sensor-equipped toys such as SenToy (Paiva et al., 2003) were found to provide an effective means of self expression, an essential requirement for successful role play. Another approach to engage users is the use of so-called magic objects to enhance experience through discovery (Rawat, 2005).

So far, only a few studies have been conducted directly comparing desktop-based interaction with novel forms of interaction within a physical environment. A study by Fails et al. (2005) comparing different versions of the Hazard Room Game that contains elements of role play and interactive story telling indicated that interaction in a physical interactive environment may increase the learner's interest and understanding compared to a traditional desktop-based version. Their study also revealed interesting gender-specific differences - while girls verbalized a lot, boys made more use of the tangible props. Dow et al. (2007) investigated the impact of different interaction styles on the user's sense of presence and engagement by comparing three versions - a desktop keyboard-based version, a speech-based version and an Augmented Reality version - of the story telling system Façade (Mateas and Stern, 2003). Their study revealed that interaction in Augmented Reality enhanced the sense of presence but reduced the player's engagement. A similar observation was made for the keyboard-based versus the speech-based version where the more natural form of interaction did not necessarily contribute to a more compelling experience. Overall, these studies indicate that a deeper understanding of the relationship between presence and engagement is needed to create interfaces that appropriately support interactive role play.

Another question relevant to our research is how interfaces can help to foster social interaction between learners. Inkpen et al. (1999) observed that by giving each learner an input device, a positive effect on collaboration results when solving a puzzle even if only one learner could interact at a time. Mandryk et al. (2001) investigated the use of handheld devices to foster collaboration between learners in an educational game. Their study revealed that learners preferred to play the game with friends than by themselves and that the learners spent a great deal of time interacting with each other. Stanton et al. (2001) observed in their study to support learners in stories creation that the use of multiple mice contributed to more symmetrical interactions and higher engagement. Additionally, it was observed that by assigning each user a specific role tied to an interaction device with a dedicated function, more organised interaction within a group is produced, balancing the level of interactivity and avoiding dominant users (Leichtenstern et al., 2007).

Overall, there is empirical evidence that learners seem to be more engaged and more active when playing on a computer with multiple input devices and cursors than when using a computer by themselves. These studies also indicate the great potential of tangible and embodied interaction for improved interaction experience as opposed to desktop based interaction.

4.2 Interaction Devices in ORIENT

In ORIENT, it is important for user interaction modalities to reinforce the story world and bring it into the real world to ensure a successful role-play and establishment of social believability. Taking the different studies into consideration, ORIENT's user interface was designed to be physical and tangible so that discrepancy between action and perception can be reduced. Interaction is supported through large and micro screens, physical interfaces and multi-modal interaction devices. Full body interaction and movement in the physical space, particularly important in social behaviour and culturally specific interaction are supported as shown in Figure 3. Each user is assigned a role which relates to a specific interaction device - a mobile phone, a Dance Mat or a WiiMote - that has unique functions, necessary to achieve the overall goal of the game. Bluetooth communication is utilised for both the mobile phone and the WiiMote while the Dance Mat is connected to the computer through USB.

The Nokia NFC 6131 phone supports speech recognition and RFID-based input. The recognition of 'magic words' is needed for the users as a means to grab the characters' attention in order to communicate. On the other hand, the RFID reader on the phone allows users to interact with physical objects augmented with RFID tags. These objects exist both in the real world and the virtual world and by simply touching a physical object with the phone, the same object will be selected in the story world. Thus, users can pick up objects and exchange or trade them with the Sprytes.

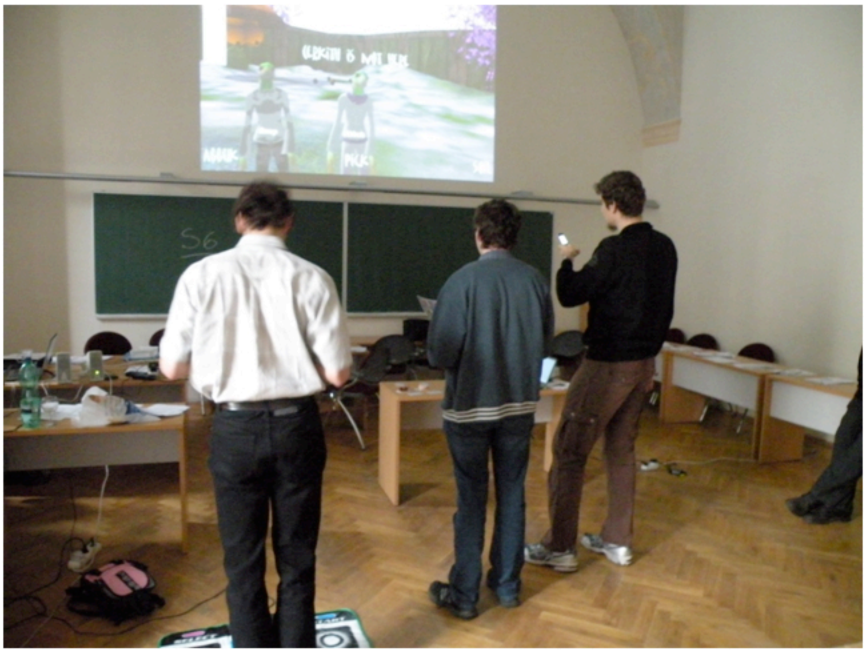


Figure 3: User interacting with ORIENT

The WiiMote uses accelerometers to sense movements in 3D space. Acceleration data is gathered from three axes (x: left/right, y: up/down, z: back/forth) and contributes to a typical signal. Features are calculated on the signal vectors and used for the classification task. In ORIENT, the WiiMote is used for expressing communicative content by gestures. It allows training of arbitrary three dimensional gestures that are closely linked to the storyline, for example, greeting by moving the WiiMote horizontally from left to right. The use of gestures for communication eliminates the need for natural language processing which is still not very realiable. Gesture recognition is realised by the WiiGLE software (Wii-based Gesture Learning Environment)[2], which allows for recording training sets, selecting feature sets, training different classifiers like Naïve Bayes or Nearest Neighbour and recognizing gestures with the WiiMote in realtime. Besides this, users can also use buttons on the WiiMote to perform selection.

Navigation in the virtual world is achieved through the Dance Mat. The users can move forward, backward and turn left or right by stepping on one of the pressure-sensitive section of the mat. This allow the exploration of the virtual world. Besides visual output, the virtual world is also enriched with audio effects, such as birds chirping, wind blowing and wave splashing to create a sense of presence in the users.

During the game, users have to work together not only to achieve a common goal but at each input phase. First, the user controlling the Dance Mat will navigate the group to their chosen destination. Then, in order to send a message or request to the Sprytes, the users having the mobile phone and the WiiMote have to cooperate to create a single input phrase to be sent to the system. Each phrase consists of an Action Target (Spryte name, that is, the magic word), an Action (gesture performed with WiiMote) and an Object (embedded with RFID tag). Object is optional.

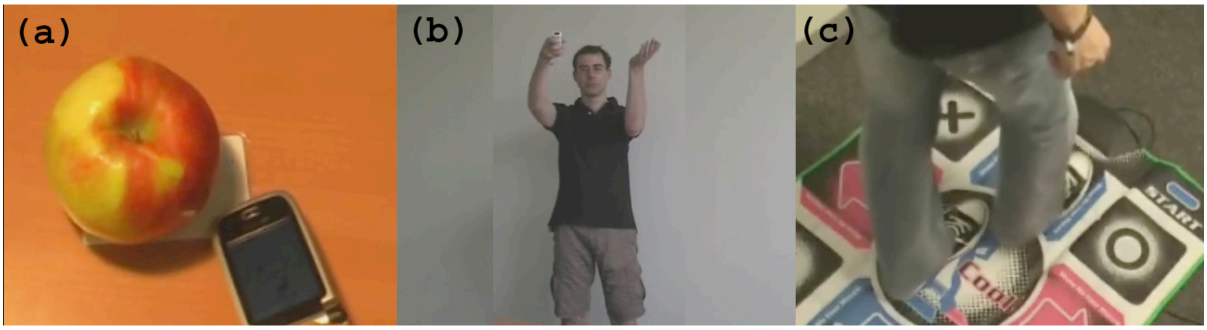


Figure 4: ORIENT interaction devices

4.3 The ORACLE As A Parallel Communication Channel

The ORACLE is a 2D Flash character animated in real time by a patent-pending software developed by Interagens[3] as shown in Figure 5. The ORACLE's mind is a production system containing "reactive" rules, that fire when the user presses the "Help!" button, and "proactive" rules, that fire according to the occurrence of specific events in ORIENT. A Java socket server connects ORIENT, Drools[4] rule engine and the Flash client on a phone. The ORACLE's main goal is to aid users in their mission and enhance their learning in the game. It is operated by the user who is controlling the dance mat.

It performs its pedagogical function by asking appropriate questions and making comments on users' actions. It also helps to keep the users motivated and engaged during the mission. In terms of the users' perception of the Sprytes, the Oracle can help by explaining the current situation (e.g. this Spryte is angry at you, you should apologize) and thus clarifying Spryte behaviour that was unclear to the users. However the rules driving the Oracle will only proactively make those suggestions if it is clear that the users have not understood the Sprytes' behaviour. In such cases, the phone rings to attract the user's attention before the ORACLE starts giving advice. Passively this information is always available through the the "Help!" button. When the user presses the "Help!" button on the user interface: the ORACLE analyzes the game situation and displays

[2] http://mm-werkstatt.informatik.uni-augsburg.de/wiigle.html
[3] http://www.interagens.com/
[4] http://www.jboss.org/drools

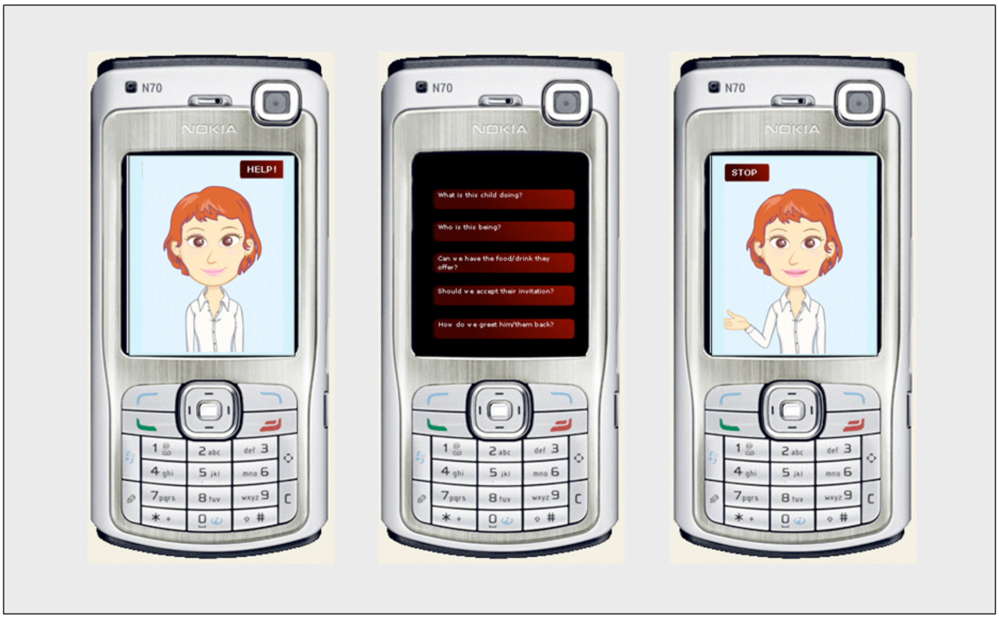

N70
NOKIA
HELP!
What is this child doing?
Who is this being?
Can we have the food/drink they offer?
Should we accept their invitation?
How do we greet him/them back?
STOP

Figure 5: The ORACLE user interface

a set of disambiguation questions for the user to choose from (second picture in Figure 5), the ORACLE then plays a predefined answer corresponding to the selected question.

4.4 Interaction Scenario

During the mission, the users will witness the Sprytes' lifestyle and values. An example scenario that is related to the Sprytes' life cycle (Figure 2(b)) is described below. The phrases in italic are the output of the system.

> The interaction starts with the Sprytes performing the *'Greet' gesture*. In response, the users return the greeting to each of the Sprytes present: Subject (calling the Spryte's name into the mobile phone) + Action (performing the 'Greet' gesture using the WiiMote). After the greeting, a Spryte will invite the users to the 'recycling of the dead' ceremony - *audio output of giberrish and translated subtitle on the screen*. The users can accept or reject this invitation by inputting the Subject (Spryte who invited them) + Action ('Accept' or 'Reject' gesture) + Object (scanning the Recycling RFID tag). Assuming the users accepted the invitation, the Spryte will ask users to follow it - *gibberish and subtitle output*. Users can move in the direction of the Spryte by stepping forward on the Dance Mat. As users arrive at the recycling machine, they will be invited to press a button on it to start the recycling process - *audio output and subtitle*. The users can ask questions about the recycling machine as well as the recycling ceremony by sending Subject (Spryte's name) + Action ('Ask' gesture) + Object (RFID tag for the topic).
>
> There are two phases in the recycling process which can be achieved through buttons on the recycling machine. First, the dead Spryte body will be dried and the machine will produce some green juice - *a cup with green juice will appear on the machine when the right button is pressed*. The second step involves crushing the dried body into soil - *a bag of soil appearing at the side of the machine when the right button is pressed*. These steps have to be performed in order. The 'Green' button on the machine (button '1' on the WiiMote) will achieve the first step while the 'Red' button (button '2' on the WiiMote) will achieve the second step. Thus, the users have to make a choice

and if they made the wrong choice, then they will break the machine and the Spryte will be angry - *audio output, subtitle and 'Angry' gesture.* The Spryte will forgive the users - *performing 'Forgive' gesture* - if they apologise: Subject (Spryte's name) + Action ('Apologise' gesture). If they select the right button, they will be invited to drink the green juice - *audio output and subtitle.* Here again, they can accept or reject the invitation by performing the Subject + Action + Object input and their response affects future relationship with the Sprytes. Let's say they rejected the offer, the Spryte will be angry - *angry gesture* - because the users are considered disrespectful by refusing the great honour presented to them. In this situation, the interaction can proceed only if the users apologise to the respective Spryte. Again, the Sprytes will accept the apology by performing the *'Forgive' gesture.* The scenario ends when the soil is produced and is being offered to the users as a gift - *audio output, subtitle and 'Give' gesture.*

4.5 User Interaction Informing Multimodal Output

The input modalities described above had a profound impact on the Spryte's communication design, in particular in relation to gestures. Because there are many more communicative actions that we wanted the Sprytes to perform than we could generate gestures, we needed some kind of measure to decide which communicative acts should be represented through a gesture instead of language. Since the users also communicate using gesture this decision became easier. We simply gesturized those communicative acts that the users also had to use, that is, those that were important verbs in the users communicative repertoire. These include greeting, offering, accepting, rejecting, apologizing, asking and giving attention to someone. The fact that any gesture that the user can perform can also be performed by the Sprytes reinforces the cultural learning component of the game. By careful observation of the Sprytes' behaviour the users can mimic their gestures through the WiiMote, enabling full body interaction in ORIENT. This symmetry of the gesture repertoire works both ways: every gesture that the Sprytes can perform can be copied by the user. This furthermore means that user interaction modalities not only had an effect on the mapping of meanings to gestures but also on the physical manifestation of the gestures. The gestures were acquired by experimenting with the WiiMote and were mainly chosen for their distinctiveness and good recognition rates using the WiiGLE software plus for the ease of learning to perform them. Videos of a user performing the gestures with a WiiMote were then sent to the graphics design team which created matching animations for the Sprytes.

5 Evaluation

The evaluation of ORIENT was designed as an in-role experience for adolescent users in UK and Germany. In total, 24 adolescents, 12 from each country participated in the evaluation. Each evaluation session took approximately 2 hours with the key aim to test the suitability of ORIENT as a tool for: (a)fostering cooperation/collaboration; and (b) fostering reflection on intercultural problems. As the focus of this paper is on the interaction modalities, only a brief discussion will be provided on the pedagogical evaluation. More information can be found in Lim et al. (2009a). Overall participants rated the prototype positively and readily engaged with it and with one another, with interactions indicating that this approach has the potential to foster cooperation among the user group. They were able to identify similarities and differences between their own and the culture of the Sprytes but found that the Sprytes are lacking individual personality. The Sprytes triggered different feelings among users in UK and Germany. German users found the Sprytes friendly while British users found the Sprytes hostile. This could either be due to different cultural backgrounds or gender differences, due to the fact that the German sample was exclusively female, while the British sample was mixed gendered. In any case this is an interesting finding that future evaluations of the system could explore further.

The technical evaluation focused on the experience of interacting with ORIENT (ORIENT Evaluation Questionnaire), the usability of the ORACLE (ORACLE Evaluation Form), and on the usability of the interaction devices (Device Questionnaire). A summary of the positive and negative feedback on the different interaction components in presented in Table 1.

Table 1: Positive and negative comments from users regarding the interaction components

Comments	**Mobile phone**	**WiiMote**	**Dance Mat**	**ORACLE**

Positive	very handy;
	very helpful;
	scanning was easy to use and interesting;
	talking was well functioning;
		good;
		was fun to play with it;
		interesting because gestures where unusual[5]
			funny and a good way of moving;
			interesting because one has to move oneself
				good and easy to use;
				useful in difficult situations;
				helped a lot - but would be better if it is a hologram

Negative	didn't work properly; hard to get it to understand things; names where hard to pronounce	complicated and too much to remember; confusing; didn't work like it should	hard to navigate; good idea, but inaccurate regarding steps and direction; sometimes goes too fast	sometimes the information were unimportant; irrelevant information; bossy

5.1 Discussion

From the users' feedback, it can be observed that they liked the idea of physical environment integration and bodily interaction because it allowed them to move more freely, hence, interact more naturally. They also liked the use of different innovative devices and novel approaches as means of input and output. They found it interesting to handle the different devices, and that all devices were needed to accomplish the interaction with the Sprytes despite the fact that it took them quite a while to be able to control the devices. Although some of the users enjoyed the challenges posed by these interaction techniques, others found the approaches too complicated. The effort and challenges frequently absorbed more of the user's time than the Sprytes and ORIENT did resulting in inappropriate focus on devices rather than the interaction.

The mobile phone worked well for RFID scanning but did not do too well in speech recognition. This might be due to the difficulty to pronounce alien names and the trainer's accent. Since speech recognition works the best when the speaker is the trainer, it is not surprising that this problem occurred in ORIENT. We tried to overcome this problem by implementing a context sensitive interface. Thus, if the users' speech is wrongly interpreted by the speech recognition system, the interface will check if the highest rating recognition refers to a character in the current scenario, if not, it will proceed to the second highest rating recognition until an appropriate character is selected.

Due to the different styles in handling the WiiMote, user-dependent recognition would be preferred. However, in order to reduce the time of evaluation, the classifiers were pre-trained and users were given a short testing session prior to the interaction to try out the different gestures. This could be the source of frustration during the interaction because the WiiGLE failed to recognise gestures performed by certain users. Additionally, there was information overload - users found it hard to remember all the 9 gestures available particularly because these gestures are uncommon to their daily life.

Navigation using Dance Mat resembles real-world movement because it required users to step on an appropriate pad in order to navigate through the virtual world. However, users found the navigation direction confusing. The main reason for this is that the pathways were predefined using way-points which might not be of the same distance or angle. Thus, users were not able to predict their movement through the scene easily.

The ORACLE was perceived as helpful assistant during the game but not very intelligent because sometimes it provided inappropriate information. For example in certain situations it would give generic answers such as "'You are in the Spryte's village"' when the users were looking for more specific information.

6 Conclusion

Applying multimodal interaction modalities in a computer game is a challenging task. This paper presents our experience on employing innovative multimodal interaction modalities in an educational collaborative role-play game, ORIENT and explains the influence this interaction technology had on the virtual character design. An interesting finding was that the same output interface triggered different responses among users from different countries which could be due to cultural differences. We hope to have provided a useful case study that shows how deeply interwoven the design of input and output modalities in pervasive games using novel interaction technology is. The design process of such an application should take this fact into account.

Acknowledgements

This work was partially supported by the European Commission (EC) and was funded from 2006 to 2009 by the eCIRCUS project IST-4-027656-STP with university partners Heriot-Watt, Hertfordshire, Sunderland, Warwick, Bamberg, Augsburg, Wuerzburg plus INESC-ID and Interagens. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein.

References

Bennett, M. J. (1993). Towards ethnorelativism: A developmental model of intercultural sensitivity. *Education for the intercultural experience*, pages 448–462.

Dias, J. and Paiva, A. (2005). Feeling and reasoning: A computational model for emotional agents. In *12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, pages 127–140. Springer.

Dow, S., Mehta, M., Harmon, E., MacIntyre, B., and Mateas, M. (2007). Presence and engagement in an interactive drama. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1487–1484.

Fails, J., Druin, A., Guha, M., Chipman, G., Simms, S., and Churaman, W. (2005). Child's play: A comparison of desktop and physical interactive environments. In *Conference on Interaction Design and Children*, pages 48–55.

Hofstede, G. (1991). *Cultures and Organisations*. McGraw-Hill, London.

Inkpen, K., Ho-Ching, W., Inkpen, K. H.-C., Scott, S., and Shoemaker, G. (1999). This is fun! we're all best friends and we're all playing: supporting children's synchronous collaboration. In *Proceedings of the 1999 conference on Computer support for collaborative learning.*

Kriegel, M., Lim, M. Y., Nazir, A., Aylett, R., Cawsey, A., Enz, S., Rizzo, P., and Hall, L. (2008). Orient: An inter-cultural role-play game. In *Proceedings of 5th International Conference on Narrative and Interactive Learning Environments (NILE), Edinburgh, UK*, pages 69–73.

Leichtenstern, K., André, E., and Vogt, T. (2007). Role assignment via physical mobile interaction techniques in mobile multi-user applications for children. In *European Conference on Ambient Intelligence, Darmstadt, Germany.*

Lim, M. Y., Aylett, R., Enz, S., Kriegel, M., Vannini, N., Hall, L., and Jones, S. (2009a). Towards intelligent computer assisted educational role-play. In *Proceedings of the 4th International Conference on E-Learning and Games*, Banff, Canada. Springer.

Lim, M. Y., Dias, J., Aylett, R., and Paiva, A. (2009b). Intelligent npcs for educational role play game. In et al., F. D., editor, *Agents for Games and Simulations*, pages 107–118, Budapest, Hungary. Springer-Verlag Berlin Heidelberg.

Mandryk, R., Inkpen, K., Bilezikjian, M., Klemmer, S., and Landay, J. (2001). Supporting children's collaboration across handheld computers. In *Proceedings of the SIGCHI conference on Human factors in computing systems, New York, USA*.

Mascarenhas, S., Dias, J., Afonso, N., Enz, S., and Paiva, A. (2009). Using rituals to express cultural differences in synthetic characters. In Decker, Sichman, Sierra, and Castelfranchi, editors, *8th International Conference on Autonomous Agents and Multiagent Systems*, pages 305–312, Budapest, Hungary. International Foundation for Autonomous Agents and Multiagent Systems.

Mateas, M. and Stern, A. (2003). Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference: Game Design Track*, San Jose, California.

Paiva, A., Prada, R., Chaves, R., Vala, M., Bullock, A., Andersson, G., and Höök, K. (2003). Towards tangibility in gameplay: building a tangible affective interface for a computer game. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 60–67, New York, NY, USA. ACM.

Rawat, T. J. (2005). Wonder objects: Magic and interactive storytelling. In *3rd International Symposium on Interactive Media Design (ISIMD)*.

Stanton, D., Bayon, V., Neale, H., Ghali, A., Benford, S., Cobb, S., Ingram, R., Wilson, J., Pridmore, T., and O'Malley, C. (2001). Classroom collaboration in the design of tangible interfaces for storytelling. In *Proceedings of the SIGCHI conference on Human factors in computing systems, New York, USA*.

Generating Verbal Assistance for Tactile-Map Explorations[1]

Kris Lohmann, Matthias Kerzel, Christopher Habel
Department of Informatics
University of Hamburg
D-22527 Hamburg, Germany
{lohmann | kerzel | habel}@informatik.uni-hamburg.de

Abstract

Tactile maps are a means to communicate spatial knowledge providing access to spatial representations of knowledge for visually impaired people. However, compared to visual maps, tactile maps have some major drawbacks concerning the integration of information due to the need of sequential exploration. Verbal descriptions providing abstract propositional knowledge have an advantageous effect on tactile map reading. They can be used to communicate knowledge that on a visual map is usually realized in the form of textual labels. Further, verbal assistance can facilitate the acquisition of global spatial knowledge such as spatial relations of streets and support the tactile-map user by assisting exploration, for example, by giving information about landmarks next to a street. This paper presents an approach towards a verbally assisting *virtual-environment tactile map (VAVETaM)*, which provides a multimodal map, computing situated verbal assistance by categorizing the user's exploration movements in semantic categories called MEPs. Three types of verbal assistances are discussed. VAVETaM is realized using a computer system and the PHANToM® desktop haptic force-feedback device, which allows haptic exploration of 3D-graphics-like haptic scenarios.

Keywords: verbal assistance, tactile map, haptic, representation, propositional, analog, spatial-analog

1 INTRODUCTION

Tactile maps provide blind and visually impaired people with useful means to acquire knowledge of their environment. As such, they can be used as substitutes for visual maps (Ungar et al., 1993). As Espinosa et al. (1998) point out, tactile maps can potentially increase the independence and autonomy of blind and visually impaired people, in particular for navigation in complex urban environments without the assistance from a sighted guide. Although different types of tactile maps are in use, neither generally agreed principles for tactile-map design nor standards of tactile-map production exist today Perkins (2002); even if Perkins' progress-report covers the phase 1993 to 2001, with respect to design principles the situation has not changed. On the other hand, the technological development has led to additional options in map production and in haptic interfaces (see below.)

Compared to visual maps, the major problem in using tactile maps is due to the restriction of the haptic sense regarding the possibility of simultaneous perception of information, for an overview see Loomis et al. (1991). In haptic perception additional effort has to be assigned to integrate information perceived over time. This leads in the case of map exploration to specific limitations for building up cognitive maps, such as *sparse density of information* and *disadvantage of survey knowledge compared to route knowledge*. Due to the restriction of the haptic sense in simultaneous perception of information, additional information given in another modality, e.g., speech, can be very useful (Wang et al., 2009). The increasing availability of haptic interfaces for human-computer interaction (HCI) offers a large variety of prospects for training and assisting blind people. In particular, by the means of such devices (e.g., the PHANToM® desktop used for VAVETaM), it is possible to realize map-like representations of physical environments that are HCI counterparts to traditional tactile maps (Kostopoulos et al., 2007; Lahav & Mioduser, 2000). *Virtual-environment (VE) tactile maps* offer the option to generate situated verbal

[1] The research reported in this paper has been partially supported by DFG (German Science Foundation) in ITRG 1247 'Cross-modal Interaction in Natural and Artificial Cognitive Systems' (CINACS). We thank the anonymous reviewers for their highly useful commentaries.

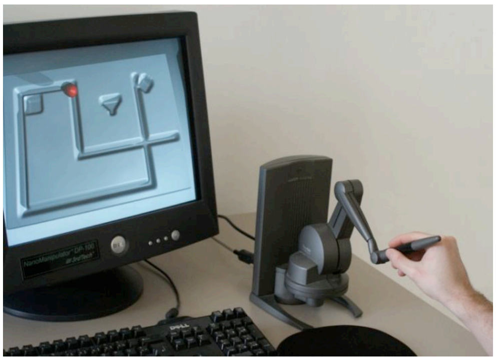
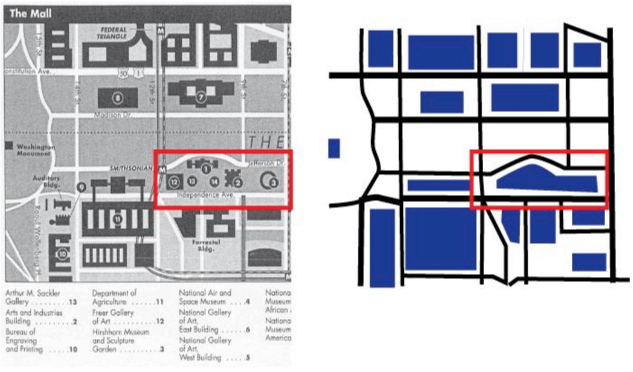


Fig 1: PHANToM® desktop and visualized VETM

descriptions (compare figure 1 for a visualized virtual-environment tactile map in use). Thus, both representational modalities, maps and language, can be used to communicate spatial information. In particular, the sequential nature of verbal descriptions supports incremental construction and updating of spatial knowledge.

The multimodal combination of *virtual-environment haptic interaction* and *assistive auditory signals* has been proved to increase speed and accuracy in exploring tactilely depictions of different types, as well as the reliability of their interpretations: this holds, inter alia, for maps (Jacobson, 2002), graphs (Wall & Brewster, 2006) and tables (Kildal & Brewster, 2007). Several approaches to augmented tactile mapping systems exist, but they do not take the generation of natural language assistance in *interaction with the user's movements* into account (Wang et al., 2009; Jacobson, 1998; De Felice et al., 2007; Parente & Bishop, 2003; Moustakas et al., 2007). The approach presented in this paper combines both types of modalities for assistance, namely virtual-environment haptic interaction and natural language assistance, by focusing on the generation of natural language assistance based on computerized understanding of the *map-exploration procedures (MEPs)* (see section 4), i.e. by exploiting the movements the user does during exploring the virtual-environment tactile map to generate discourse that helps the user in building an internal mental map. With an abstract semantic categorization of the users' movements, knowledge about what they explore can be used to compute verbal assistance in scenarios where augmenting the haptic representation provides useful hints either for further exploration procedures or for the efficiency of building up survey knowledge of the environment represented in the map. Additionally, besides the description of labels in visual maps, users demand information about locations of auditory landmarks like audio-enabled traffic lights and further information about the relations of complex entities such as long streets (Wang et al., 2009). In our approach towards *verbally assisting virtual-environment tactile maps (VAVETaM)* presented in this paper, verbal descriptions are used to communicate three kinds of verbal assistances: (a) *labeling information* such as street names, (b) *complex global spatial relations* such as parallel roads or junctions in exploration direction, and (c) *comments to instruct exploration*, for example, if a landmark that is supposed to be important has been 'overlooked' (compare section 5 for an example).

A first example for the improvement of tactile maps with verbal descriptions is shown in figure 2, taken from a tourist guide of Washington[2]. This map provides a good example for the usefulness of augmenting tactile map exploration with verbal assistance, as it includes both, a large density of labeling information such as street and building names, and a lot of salient global spatial relations, such as streets being parallel.[3] Within the virtual-environment tactile map, written textual labels cannot be used. Even though the exemplifying visual map is a relatively straightforward one, exact information about the shape of the buildings is not (re-)presented within the tactile map modality, as shown in figure 2. Instead, verbal descriptions can be used to communicate further, more detailed knowledge about a given entity. This can be information about the name of a building, another landmark (e.g., *'This is*


The Mall
FEDERAL TRIANGLE
15th St.
14th St.
12th St.
9th St.
7th St.
onstitution Ave.
50
1
Madison Dr.
THE
Washington Monument
SMITHSONIAN
M
Jefferson Dr.
Independence Ave.
Auditors Bldg.
Raoul Wallenberg Pl.
Forrestal Bldg.
1
2
3
7
8
9
10
11
12
13
14
Arthur M. Sackler Gallery 13
Arts and Industries Building 2
Bureau of Engraving and Printing 10
Department of Agriculture 11
Freer Gallery of Art 12
Hirshhorn Museum and Sculpture Garden 3
National Air and Space Museum 4
National Gallery of Art, East Building 6
National Gallery of Art, West Building 5
Nationa Museum African
Nationa Museum America

(a) Visual Map (b) Abstraction for Tactile Map

Fig 2. Example of an abstraction for a tactile map of the National Mall of Washington

[2] The depiction of the left map is derived from: Fodor's Washington, D.C. 2001, page 29. © Fodor's Travel Publications; Random House: New York.

[3] We have chosen *travel guides* as one domain of application. In particular we use 'published' map-text constellations to design tactile maps and to determine verbal comments to be adequate in assisting a haptically map-exploring user.

the Washington Monument'), or further information about a complex of buildings too intricate to be represented in the haptic modality like the one marked in figure 2. A useful output in this case could be: *'The landmark you are exploring consists of four large buildings. In the west is Freer Gallery of Art. In the east is Hishhorn Museum and Sculpture Garden. In between there is the Smithonian Institution Building and the Arts and Industries Building.'*

Another example is shown in figure 3. The red line indicates the exploratory movements along street segments of the map. At the point indicated with the arrowhead, several verbal assistances are possible. A useful assistance concerning labeling information would be to state the name of the street explored: *'You are exploring Independence Avenue'*. Further, information about the global spatial relations is useful for the integration of spatial knowledge: *'The street you are exploring is parallel to Constitution Avenue you explored before'* or *'You are heading towards a junction with 7th Street'*. A major drawback of tactile map exploration is the limited sensor field in exploring by finger movement; especially in virtual-environment haptics, it is complicated to find landmarks next to the track. Therefore, a verbal assistance such as: *'You are passing the church'* would be very useful. As exploration continues during the utterance, due to the time constraints resulting it is—in many scenarios—more efficient to say: *'You are passing three buildings'* than to mention each single building.

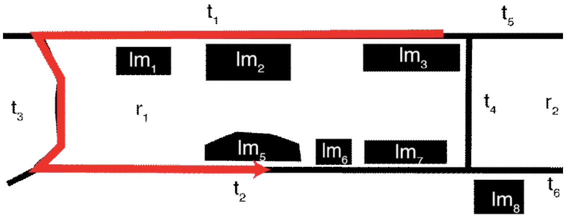


Fig 3. Example exploration of streets in a tactile map

2 An outline of VAVETaM

The system we propose has two major interaction modalities: On the one hand, a virtual-environment tactile map accessible by the haptic device, and, on the other hand, verbal descriptions providing additional assistance. The virtual-environment (VE) tactile map is based on a virtual three-dimensional haptic space, which can be explored by moving a *virtual interface point (IP)* with the handle of the device (Salisbury et al., 1995). The virtual tactile map is realized by using 3D-graphics-like shapes. A VE-tactile map can, for example, be a simulated plane area with depressed lines representing streets and depressed or raised areas representing landmarks (see figure 1, which shows a simple map for training people in the usage of our VE-tactile maps). During exploration, the user moves the device and information about these movements is accessible to the system.

The structure of VAVETaM is illustrated in figure 4: A component called *Virtual-Environment Tactile Map* component (VETM) provides a model of the tactile map including spatial-geometric specification and propositional information (such as qualitative relations between map entities and labeling information). As maps can be seen as hybrid representation systems for knowledge about the physical environment (see section 3 for more detailed discussion), the VETM consists of two representational layers, a spatial-geometric layer and a propositional layer. The spatial-geometric layer enables the generation of *spatial-analog map presentations*, in particular for tactile exploration.

While this VE-tactile map presentation is explored, the *Haptic Device* provides position and, hence, movement information. This information is processed by the so called *MEP Observer*, a component essential for the interaction between the modalities, which is discussed in more detail in section 4. The MEP Observer is the system internal counterpart to human assistants who observe a tactile-map exploring user. Based on their observation of the hand movements and their interpretation of the map, the assistants are able to give verbal comments. The stream of movement data has to be represented abstract and interpreted semantically, therefore, the movements are categorized in *map-exploration procedures (MEPs)* in the *MEP Observer*, which consists of two subcomponents, the *Haptic-Movement Observer (HMO)* and the *MEP Categorization (MEPC)*. Furthermore, the MEP Observer has access to the *MEP Specification* component providing information about the MEPs in use during the exploration movements.

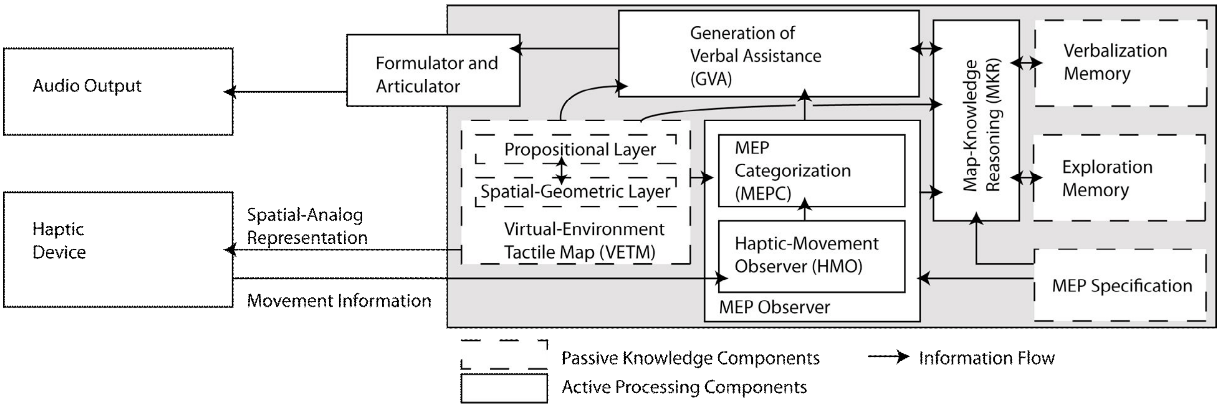

Audio Output
Formulator and Articulator
Generation of Verbal Assistance (GVA)
Map-Knowledge Reasoning (MKR)
Verbalization Memory
Exploration Memory
MEP Specification
Propositional Layer
Spatial-Geometric Layer
Virtual-Environment Tactile Map (VETM)
MEP Categorization (MEPC)
Haptic-Movement Observer (HMO)
MEP Observer
Haptic Device
Spatial-Analog Representation
Movement Information
Passive Knowledge Components
Active Processing Components
Information Flow

Fig 4. Structural model of VAVETaM

For the generation of non-redundant and efficient verbal assistance, it is essential to analyze the user's exploration, in particular, to build assumptions about what parts of the maps are known and what parts are still unexplored. This information is stored by the *Map-Knowledge Reasoning (MKR)*. This component accesses two memories: The *Verbalization Memory* and the *Exploration Memory*. The MEP Observer's output and the assumptions about the knowledge the user has gained from exploration and verbal assistance are used within the *Generation of Verbal Assistance (GVA)* component generating propositional, pre-verbal messages (Levelt, 1989) in order to fill *informational needs* of the user (Pirolli & Card, 1999). Once such a pre-verbal description is generated, it is stored in the *Verbalization Memory* and is sent to the *Formulator and Articulator* component to generate a speech.

3 HYBRID REPRESENTATION WITHIN MAPS – VIRTUAL-ENVIRONMENT TACTILE MAPS

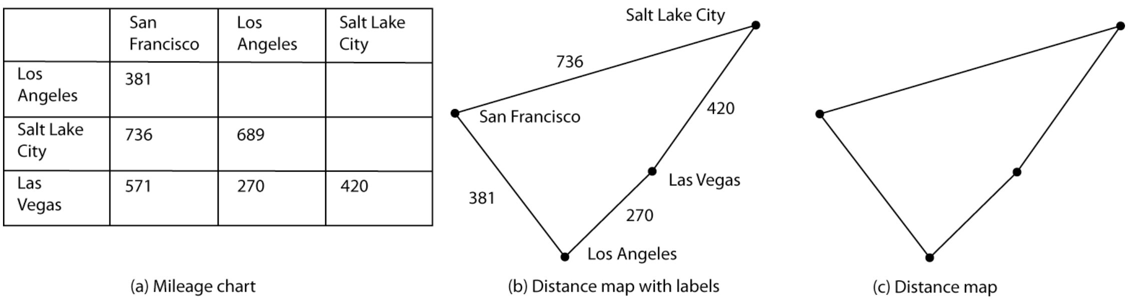
In the previous section the need for a model of spatial representations within the VETM was described. In order to construct such a model, the formats for representing knowledge in maps are discussed in this section.[4] We focus here on maps as *external* representations, in contrast to *internal* spatial representations usually called mental maps (see, e.g., Lobben, 2004).

Generally, the investigation of representation in cognitive science has led to the discussion of two representational setups: *propositional* and *analog* (Palmer, 1978). A propositional representation is, in contrast to the analog, discontinuous. This means, a propositional representation has relational entities that correspond to entities represented. Paradigm cases for propositional representations are written and spoken language, operator-operand structures or table-like representations as the mileage chart shown in figure 5(a). Typical definitions of analog representations include that the representation is organized continuously rather than discrete. Further, analog representations preserve spatial information about what they present (Palmer, 1978). As the notion 'analog representation' is deceptive in respect to VAVETaM being realized on a digital basis, this kind of representation will for the sake of a clear terminology in this paper be referred to as *spatial-analog*. A prototypical example for a spatial-analog representation is a depiction of distances as shown in figure 5(c): The spatial relations between the cities are spatially represented in this depiction. Maps represent spatial relations in a spatial-analog way. In addition, visual maps usually rely on labeling. Furthermore, we know about domain-specific concepts that are included in a representation of a map, e.g., streets are depicted as lines or water is depicted as a blue area. This conceptual knowledge is knowledge about *map concepts (MCs)* (Habel 2003). Map concepts consist of conventional knowledge about the components occurring on a map and the conventional knowledge about the usual

[4] Our use of the notion 'format' is committed to Kosslyn's discussion of propositional and depictive formats (see, e.g., Kosslyn, Thompson and Ganis, 2006, pp. 8-14.)

spatial-analog depiction of those. Maps are *hybrid representation systems* (Habel, 2003). Compare figure 5(b) for an illustration of a prototypical hybrid representation system. Maps differ from this representation in that they are not only hybrid due to their labeling, but also due to the interpretation relying on map concepts being hybrid themselves.

It is plausible to assume that tactile maps work in the same way as visual maps, even though the map concepts vary due to the representational possibilities of the tactile map setup, that is, the resolution and complexity is reduced due to haptic perception and the need for the integration of sequential percepts. As in the visual scenario, within the VETM component one layer is spatial-geometric. This layer provides the information necessary to realize a spatial-analog map explorable using the haptic device. To allow this, map concepts stored in the propositional layer are linked with geometric specifications on the spatial-geometric layer (compare figure 4). Further, information about how to depict a map concept is stored for each map concept selected for depiction[5] (compare Maaß (1994) for a similar approach). The hybrid representation within the VETM component enables to compute the information to be verbalized in the Generation of Verbal Assistance component. As the representation is hybrid, both, spatial reasoning and propositional reasoning are possible (compare Habel, Kerzel, & Lohmann (2010) application scenarios of spatial reasoning done with visual routines). As the assistance has to be given situated, the exploration of the user is used as an input for the Generation of Verbal Assistance. To enable reasoning, the input is categorized into semantic categories in the MEP Observer component.



	San Francisco	Los Angeles	Salt Lake City
Los Angeles	381		
Salt Lake City	736	689	
Las Vegas	571	270	420

(a) Mileage chart (b) Distance map with labels (c) Distance map

Fig 5. A propositional (a), a hybrid (b) and an analog representation (c) (partly derived from Habel (2003))

4 THE MEP OBSERVER

A map-exploration procedure (MEP) is an abstract semantic description of the user's exploration movements linked to the desired knowledge about the map. Examples for MEPs are *track-MEP* (the term track is used as a general term for street-like structures involved in route planning) and *distance-MEP*. The first describes an exploration process for tracks and consists basically of straight movements along the track, while the *distance-MEP* is an estimation of the distance between two map entities, for example, a track and a landmark. When knowledge about a track is needed, this entity is explored using typical movements that, on an abstract level, form a *track-MEP*. The basic set includes four MEPs related to the desired knowledge: *track-MEP*, *landmark-MEP*,

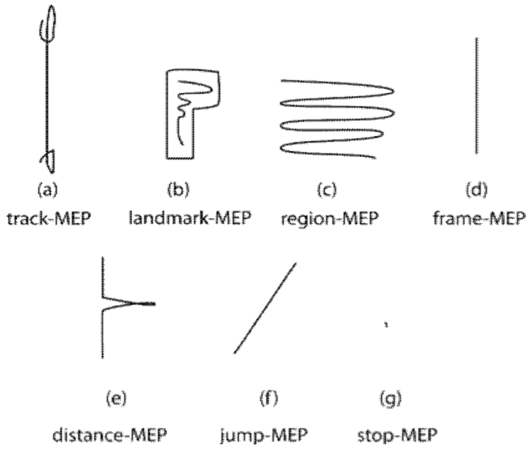


Fig 6: Movement patterns (preliminary set of MEPs)

[5] We use 'depiction' as technical term corresponding to Kosslyn's use of 'depictive representation' (see, e.g., Kosslyn, Thompson and Ganis, 2006). Thus depictions are not restricted to the visual modality, but are also fundamental for the generation of haptic representations.

region-MEP, and *frame-MEP*. An extended set includes three additional MEPs: *distance-MEP*, *jump-MEP*, and *stop-MEP* (See figure 6 for the movement patterns in exploring printed tactile maps. Habel, Kerzel and Lohmann (2010) discuss MEPs in more detail).

The aim of the MEP Observer is to recognize the users' MEPs in order to make assumptions about what information they have already gathered from their interaction with the map and what their current informational needs are. The MEP Observer is a core component in incremental conceptualization:[6] The stream of data from the haptic device is segmented into *Perceptual Units (PUs)* that represent the position of the interface point in the virtual environment with a fixed temporal and spatial resolution, abstracted from the actual hardware. These perceptual units are aggregated into conceptual representations in a hierarchical process resulting in assumptions about the MEPs executed by the user. As it is the aim of VAVETaM to provide verbal assistance accompanying the haptic exploration of the virtual-environment tactile map, it is important that the MEP Observer recognizes MEPs as early as possible, even if they are not yet finished. For example, if the user is tracing a track, the system should be able to recognize the resepctive *track-MEP* and could thus provide verbal assistance while the user is still exploring the track (compare figure 7 for a depiction of the aggregation hierarchy).

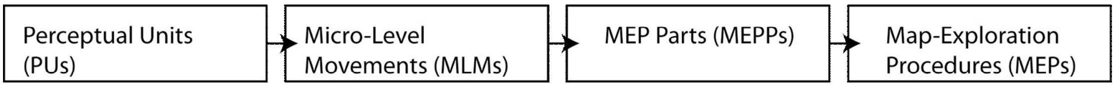

Perceptual Units (PUs)
Micro-Level Movements (MLMs)
MEP Parts (MEPPs)
Map-Exploration Procedures (MEPs)

Fig 7: Aggregation hierarchy from PUs to MEPs

This is accomplished by first segmenting the perceptual units into *Micro-Level Movements (MLMs)* utilizing both procedures for gesture recognition by the Haptic-Movement Observer and pre-segmentation depending on the position of the interface point in relation to the different objects in the virtual-environment tactile map by the MEP Categorization component, i.e., the interface point touching the surface of an object representing a track, a landmark, the empty map surface in between or the empty space above the tactile map. MLMs represent basic user movements in relation to objects of the virtual-environment tactile map, e.g., touching a track, tracing a track or leaving a track. Once an MLM is recognized subsequent perceptual units can still be associated with the same MLM, i.e., a *trace-track-MLM* is recognized while the user is still tracing the track, further tracing of the track will not create another *trace-track-MLM* but will be associated with the already recognized *trace-track-MLM*.

MLMs are stored in the Exploration Memory and are further aggregated to *MEP Parts (MEPPs)*, which represent the basic building blocks of MEPs. For example, an MEPP describing a single track being explored would be constituted by the MLMs of touching, tracing, and finally leaving the track in question. MEPPs are further combined—in a hierarchical manner—to form MEPs (kindred to Guhe et al., 2000). Figure 8 shows a visualization of an exploration, consisting of exploring a track t_1 and subsequently exploring a track t_2, with track t_2 being connected to track t_1. In processing this exploration, the MEPP for exploring a track t_1 and the subsequent MEPP for exploring a track t_2, with track t_2 being connected to track t_1, constitute the *track-MEP*(t_1,t_2) as shown in figure 9. Like MLMs, both MEPPs and MEPs need not be complete in order to be recognized. The *track-MEPP*(t_2) and the *track-MEP*(t_1, t_2) is constructed although the final *cease-touch-track-MLM(t2)*, which is depicted in grey, is still missing and more perceptual units can get associated with the *trace-track-MLM*(t_2). In other words, the snapshot depicted in figure 9 is the result of a process of building a plausible hypothesis, which possibly has to be modified or to be changed later.

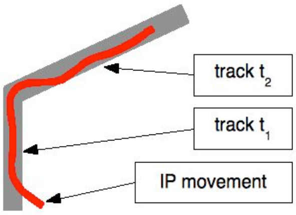


Fig 8: Subsequent exploration of two connected tracks

[6] The VAVETaM conceptualizer – a subcomponent for the Generation of Verbal Assistance (GVA), see figure 4 – will be based on the INC approach (see, Guhe et al, 2000). Guhe and Habel (2001) discuss the incremental conceptualization in the kindred domain of verbalization of 'acts of drawing line configurations'.

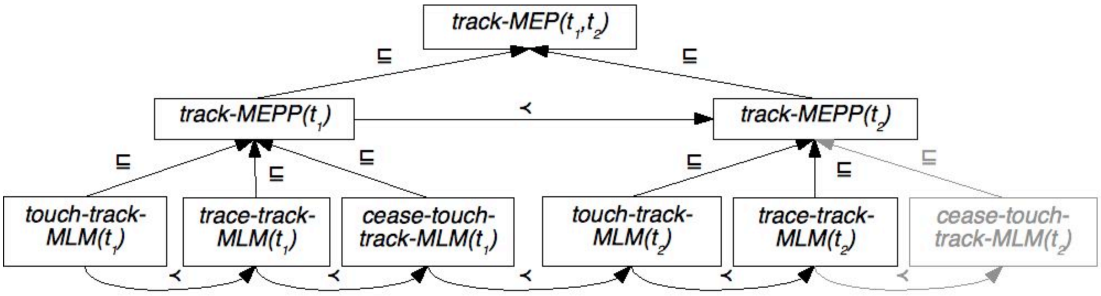


Fig 9: Hierarchical structure of a *track-MEP* (□ indicates part of relation, □ indicates subsequent MLMs or MEPPs)

5 THE GENERATION OF VERBAL ASSISTANCE

The user’s exploration is an active interaction with the VAVETaM. As the MEP Observer generates an abstract semantic representation of the user's exploration stored in the Exploration Memory, assumptions about the knowledge the map user has gained can be made in the Map Knowledge Reasoning component. This component also keeps track of the pre-verbal messages send to the Formulator and Articulator. The Generation of Verbal Assistance (GVA) basically provides output with respect to three types of assistance tasks: (a) the user explores a part of the map for the first time and has a need for labeling information, e.g., a name of a street or a building, (b) the user gets information about global spatial relations, which are difficult to detect preforming local explorations, or (c) guidance for exploration is needed, such when the user has not yet explored a salient landmark like an audio-enabled traffic light along the route which will probably be helpful for later wayfinding.[7]

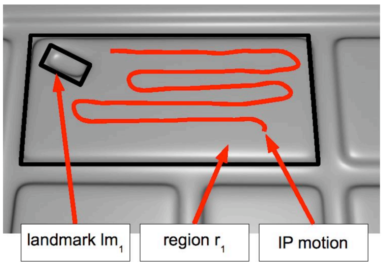


Fig 10. Assisted exploration: user overlooked lm_1 in region r_1

We use type (c) to exemplify how the Generation of Verbal Assistance interacts with the MEP Observer and the Map-Knowledge Reasoning. To produce assumptions about the current informational needs of the user, the GVA-component has to be able to reason about the map, the represented environment, and the exploration process (Habel, Kerzel, & Lohmann, 2010). Thus, MEPs are associated with informational needs. Once the MEP Observer recognizes an MEP, the associated informational need is used by the Generation of Verbal Assistance to extract information for verbalization from the VETM. This set of information is compared with the Map-Knowledge Reasoning in order to find the subset of information novel to the user.

As an example, figure 10 shows a visualized extract of a virtual-environment tactile map. The red line symbolizes the position of the interface point over time. The shape of movement is characteristic for a *region-MEP*, which is used to explore a region represented in the virtual-environment tactile map for unknown features. In this example the user has ‘overlooked’ landmark lm_1. Once the *region-MEP*(r_1) is recognized by the MEP Observer, the Generation of Verbal Assistance consults the VETM component, storing the map representation, to inspect the region r_1 for landmarks. In this case only landmark lm_1 is contained in the inspected region. Now the Map-Knowledge Reasoning is consulted. If landmark lm_1, e.g., a fountain, is neither mentioned in a preceding verbal assistance nor is the user’s haptic interaction with lm_1 recorded in the Exploration Memory, the Generation of Verbal Assistance sends this information to the Formulator and Articulator, which generates a verbal assistance such as ‘*You have missed the fountain in the upper left corner of the region you are exploring*.’

[7] Scenarios (a) and (b) are exemplified in section 1.

6 Outlook

The VAVETaM presented aims to the generation of helpful verbal descriptions that communicate street names and other information usually found as textual labels on visual maps and, added to this, verbalize information about global spatial relations and fill in knowledge gaps like unexplored entities. This is realized by analyzing the user's exploratory procedures in the MEP Observer in an abstract semantic manner using an MEP categorization related to the desired map knowledge. As two memory components keep track of the knowledge representations provided to the user by VAVETaM, assumptions about the user's map-exploration progress and the knowledge gained are made. Hence, useful verbal descriptions can be given. These verbal descriptions are not restricted to the verbalization of labels like street or building names, rather they also provide assistance with the exploration and the integration of the spatial information perceived sequentially.

The examples given in section 1 give hints to an important research question to be addressed in the future. Free map exploration to build up survey knowledge is in the focus of research. Nevertheless, a plausible usage scenario for free map exploration is to enable planning a route from a point A to a point B. During planning a route, humans make use of different levels of granularity, as shown by (Timpf & Kuhn, 2003) for the highway domain. It is very likely that granularity transformations also happen during route planning by visually impaired or blind people, and the verbal assistance should adapt to this fact. To use the example of the four buildings described above: When planning a route simply passing by the museum buildings, it may be sufficient to say that there are buildings, whereas when planning a route to the Hishhorn Museum, much more information about the location and the spatial relations towards the other buildings must be given verbally. To realize this goal, issues such as plan recognition have to be addressed.

A further scenario that is yet to be tested for its usefulness is user-triggered output, e.g., the user clicking one of the buttons of the haptic device to show the need for information. This information can be of the categories described above: Either repeating labeling information, providing exploratory guidance or global spatial knowledge.

References

De Felice, F., Renna, F., Attolico, G., and Distante, A. (2007). A haptic/acoustic application to allow blind the access to spatial information. In *World Haptics Conference* pages 310-315.

Espinosa, M.A., Ungar, S., Ochaita, E., Blades, M., and Spencer, C. (1998). Comparing methods for introducing blind and visually impaired people to unfamiliar urban environments. *Journal of Environmental Psychology*, *18:* 277-287.

Golledge, R. G., Rice, M., and Jacobson, R.D. (2005). A commentary on the use of touch for accessing on-screen spatial representations: The process of experiencing haptic maps and graphics. *The Professional Geographer*, *57*(3), 339–349.

Guhe, M., and Habel, C. (2001) The influence of resource parameters on incremental conceptualization. In *Proceedings of the Fourth International Conference on Cognitive Modeling*, George Mason University, Fairfax, Virginia, USA. pages 103–108. Mahwah, NJ: Lawrence Erlbaum.

Guhe, M., Habel, C., and Tappe, H. (2000). Incremental event conceptualization and natural language generation in monitoring environments. In *Proceedings of the first international conference on Natural Language Generation,* Vol. 14, pages 85-92.

Habel, C. (2003). Representational commitment in maps. In Duckham, M. Goodchild, M., and Worboys, M., editors, *Foundations of Geographic Information Science*, pages 69-93. London: Taylor & Francis.

Habel, C., Kerzel, M., and Lohmann, K. (2010). Verbal assistance in tactile-map explorations: A case for visual representations and reasoning. *Proceedings of AAAI workshop on Visual Representations and Reasoning*. AAAI-Conference 2010 (Atlanta, GA, USA).

Jacobson, R.D. (1998). Navigating maps with little or no sight: An audio-tactile approach. In *Proceedings of the Workshop on Content Visualization and Intermedia Representations*, pages 95-102.

Jacobson, R.D. (2002). Representing spatial information through multimodal interfaces. In *Sixth International Conference on Information Visualisation*.

Kildal, J., and Brewster, S.A. (2007). Interactive generation of overview information using speech. In *Extended Abstracts of ACM CHI 2007* (San Jose, CA, USA), ACM Press.

Kosslyn, S. M., Thompson, W.L., and Ganis, G. (2006). *The Case for Mental Imagery*. Oxford University Press, New York.

Kostopoulos, K., Moustakas, K., Tzovaras, D., Nikolakis, G., Thillou, C., and Gosselin, B. (2007). Haptic access to conventional 2D maps for the visually impaired. *Journal on Multimodal User Interfaces*, *1*(2): 13-19.

Lahav, O., and Mioduser, D. (2000). Multisensory virtual environment for supporting blind persons' acquisition of spatial cognitive mapping, orientation, and mobility skills. In *Proceedings of the Third International Conference on Disability, Virtual Reality and Associated Technologies*, pages 23–25.

Levelt, W.J.M. (1989). *Speaking: From Intention to Articulation*. The MIT Press, Cambridge, MA.

Lobben, A.K. (2004). *Tasks, strategies, and cognitive processes associated with navigational map reading: A review perspective. The Professional Geographer* 56(2): 270–281.

Loomis, J.M., Klatzky, R.L., and Lederman, S.J. (1991). Similarity of tactual and visual picture recognition with limited field of view. *Perception*, 20(2): 167-177.

Maaß, W. (1994). From vision to multimodal communication: Incremental route descriptions. *Artificial Intelligence Review*, 8(2): 159-174.

Moustakas, K., Nikolakis, G., Kostopoulos, K., Tzovaras, D., and Strintzis, M.G. (2007). Haptic rendering of visual data for the visually impaired. *IEEE Multimedia*, 14(1): 62-72.

Palmer, S. (1978). Fundamental aspects of cognitive representation. In E. Rosch and B. Lloyd (Eds.), *Cognition and Categorization,* pages 259-303. Lawrence Erlbaum, Hillsdale, NJ.

Parente, P., & Bishop, G. (2003). BATS: the Blind Audio Tactile Mapping System. In *Proceedings of the ACM Southeast Regional Conference*.

Perkins, C. (2002). Cartography: progress in tactile mapping. *Progress In Human Geography,* 26(4*)*: 521-530.

Pirolli, P., and Card, S. (1999). Information foraging. *Psychological Review,* 106(3): 643–675.

Salisbury, K., Brock, D., Massie, T., Swarup, N., and Zilles, C. (1995). Haptic rendering: Programming touch interaction with virtual objects. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 123–130.

Timpf, S., and Kuhn, W. (2003). Granularity transformations in wayfinding. In Kuhn, W., Worboys, M.F. and Timpf, S. (eds). *Spatial Information Theory 2003*, pages 77–88. Springer, Berlin.

Ungar, S., Blades, M., and Spencer, C. (1993). The role of tactile maps in mobility training. *British Journal of Visual Impairment*, *11*(2): 59-61.

Wall, S., and Brewster, S. (2006). Feeling what you hear: Tactile Feedback for navigation of audio graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1123-1132.

Wang, Z., Li, B., Hedgpeth, T., and Haven, T. (2009). Instant tactile-audio map: Enabling Access to digital maps for people with visual impairment. In *Proceeding of the eleventh international ACM SIGACCESS conference on computers and accessibility*, pages 43–50.

Use of the QuADS Architecture for Multimodal Output Generation

Ian O'Neill, Philip Hanna, Darryl Stewart,
School of Electronics, Electrical
Engineering and Computer Science,
Queen's University Belfast,
Belfast, BT7 1NN, Northern Ireland.
{i.oneill, p.hanna, dw.stewart}@qub.ac.uk

Xiwu Gu,
College of Computer Science and Technology,
Huazhong University of Science
and Technology,
Wuhan 430074, China.
guxw_wang@sina.com

Abstract

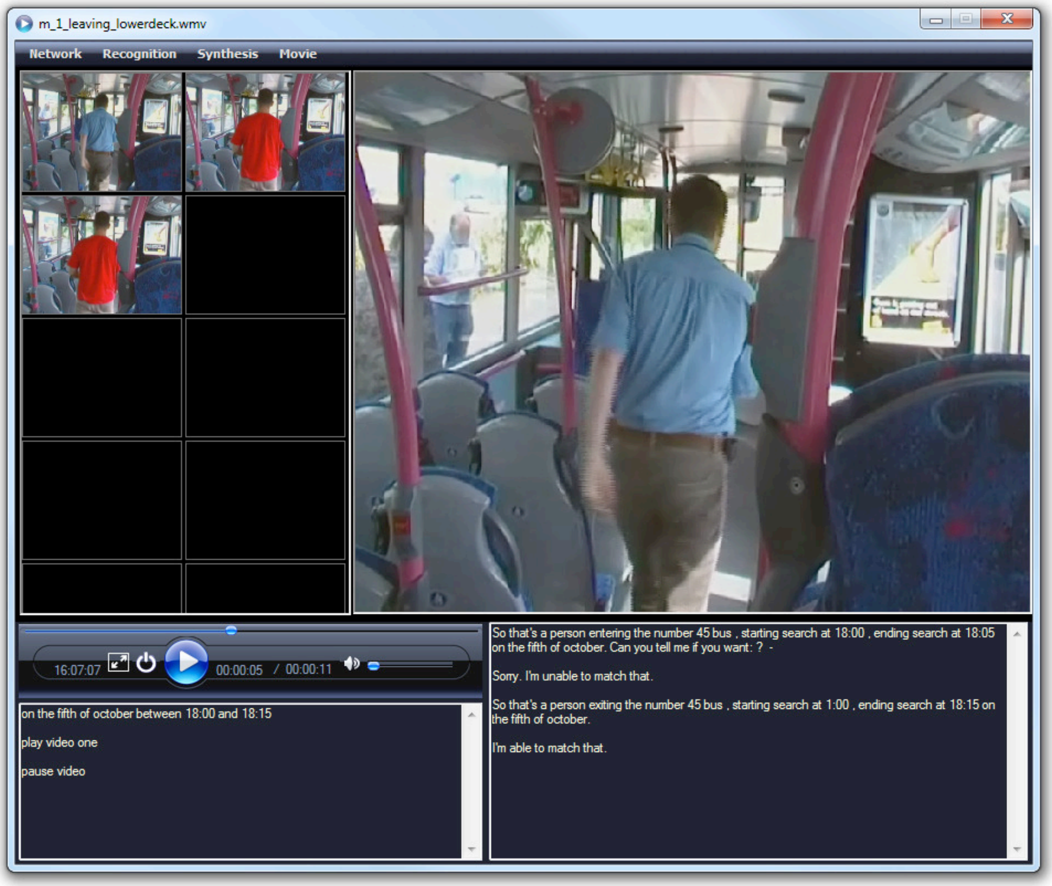
QuADS (Queen's Advanced Dialogue System) is a suite of highly generic and customizable Java classes for the development of spoken and multimodal dialogue systems. Some of the classes in QuADS represent communicative acts, such as are found in information-providing or transaction-based dialogues: in classifying these acts, whether generated by the system or understood by the system from the user's input, QuADS uses a dialogue act hierarchy based on the DIT++ taxonomy. Other classes are concerned with the underlying task of finding the information or service that the user requires, or, when the user's specific information request cannot be satisfied, of presenting reasonable alternatives. In this paper, as well as giving an overview of the QuADS architecture, we examine the means by which a system developed with QuADS selects the modalities that it will use to present information to the user, taking into account the availability of a particular modality in a given system configuration, and considering also the user's preference for particular modalities. Although we apply some obvious measures to avoid 'information overload', at present we are concerned not so much with the 'optimal' modality or combination of modalities for a particular task as with the mechanisms within a generic, domain-independent framework that make selection of modalities possible in accordance with system capabilities and user preferences.

Keywords: Multimodal Dialogue, Object-Oriented Development.

1 INTRODUCTION

ISIS is an EPSRC-sponsored Integrated Sensor Information System that analyses and stores video information gathered from public transport vehicles. Sponsored by the UK's Engineering and Physical Sciences Research Council (EPSRC, Project No. EP/E028640/1), its purpose is to detect situations that are potentially threatening to people or property. ISIS-NL is the multimodal information retrieval component of ISIS and is based on natural language dialogue. It uses a suite of re-usable dialogue components that we have developed at Queen's University Belfast: collectively these components are known as QuADS, Queen's Advanced Dialogue System (Hanna, 2008). QuADS represents a new generation of object-based natural language dialogue technology at Queen's, succeeding our Queen's Communicator architecture (O'Neill et al., 2003). Developed in Java, the QuADS toolkit adheres to the same object-oriented precepts as its predecessor, inspired by the approaches to OO development set out since the 1990's by Grady Booch and others (Booch, 2007).

ISIS-NL is intended to demonstrate how busy staff in a network operations centre can use spoken (or in some cases keyed) instructions to retrieve video footage relating to incidents of interest – for example, 'Can you tell me if a man got on a number 45 bus at Blackheath between seven-thirty p.m. and nine-thirty p.m.?' To respond to the user's request, the system potentially has available a combination of text and speech, as well as the still images and video that represent the retrieved information itself. The manner in which it presents its information and the amount of information it presents depend, amongst other things, on the number of database 'hits' or 'alternative suggestions' that the system has to convey, as well as on the user's preferences for particular modalities. The system is also limited to the output modalities that are actually available in a particular system configuration.



on the fifth of october between 18:00 and 18:15

play video one

pause video

So that's a person entering the number 45 bus , starting search at 18:00 , ending search at 18:05 on the fifth of october. Can you tell me if you want: ? -

Sorry. I'm unable to match that.

So that's a person exiting the number 45 bus , starting search at 1:00 , ending search at 18:15 on the fifth of october.

I'm able to match that.

Figure 1. A user's view of ISIS-NL.

In our latest implementation of ISIS-NL, other than introducing commonsense limitations to the amount of information the system attempts to convey in a particular modality (for example, the number of alternatives it tries to tell the user about though speech alone), we have not attempted rigorously to optimise the system's use of modalities. We have not, for instance, attempted to ensure that presentation of information is optimised to a particular individual's ability to assimilate it. However, we recognise that for some years the interplay between the modalities as channels of communication, and the effect of different modalities on the individual user in particular situations, has been a lively area of research. How and when, for example, might one modality be used to reference material that is being used in another modality? Almost two decades ago Maybury pointed out the challenges that these issues would pose for developers of multimodal interfaces (Maybury, 1992). Since then he and others have gone on to explore how information that is available in different modalities might be selected and presented in accordance with the user's preferences (Bernsen & Dybkjær, 1999; Light & Maybury, 2002; Oviatt et al. 2004). Ideally the computer-based system will replicate in its use of modalities the most natural behaviours of human conversation partners: it should be able to decide when to use speech alone, and when the communicative task best served by speech in combination with a visual indicator that is equivalent to a human conversation-partner pointing (Van der Sluis et al., 2008).

We foresee that, in many QuADS-based applications, the user himself or herself might simply decide (via a customisation GUI or even a spoken customisation sub-dialogue) which combination of outputs, which customisable configuration, is generally most effective for him or her in a typical working environment. However, our dialogue architecture is designed deliberately to provide a very high-level framework that can accommodate quite specialised behaviour at any stage of the dialogue-handling process. In Section 2.3.1 we discuss the manner in which 'forums'

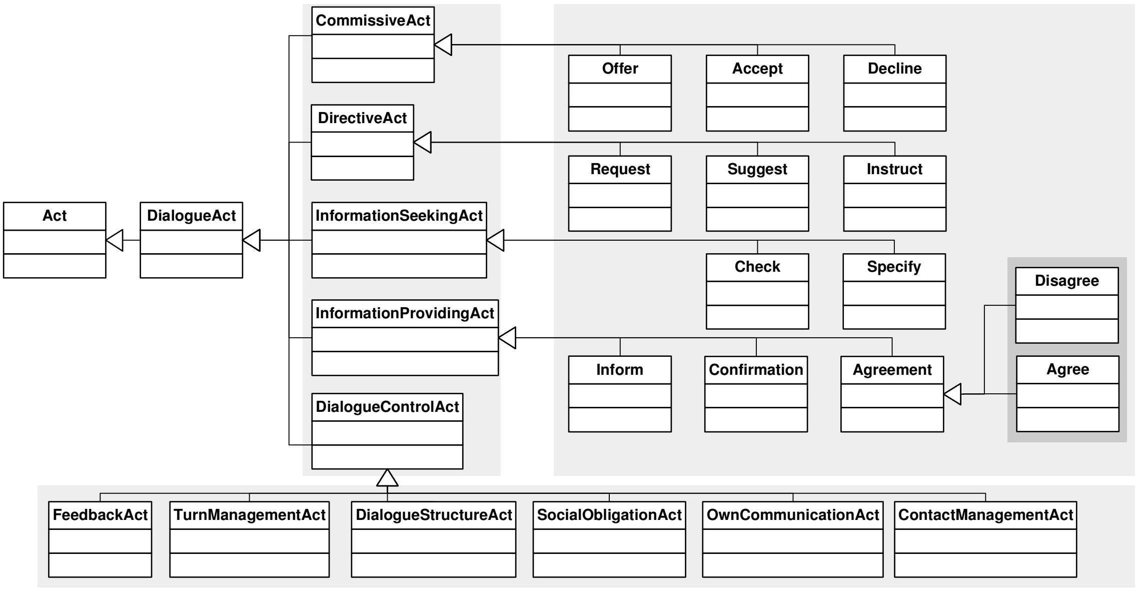


Figure 2. A representative selection of acts from the Dialogue Act hierarchy in QuADS.

are used to facilitate communication between specialised processing objects, which we refer to as *Agents*, each of which can make decisions about, and advise a co-ordinating *Manager* about, its own specialised area of dialogue-handling: some Agents might specialise in processing the transaction underlying the dialogue, others might deal with database queries and help reformulate failed information requests, and so on. While the prototype dialogue system currently implemented in QuADS has static, and rather simple hand-crafted rules to control slot-based, transaction-handling behaviour, our architecture is flexible enough to accommodate more active and subtle dialogue strategies, including those that would allow a system to learn optimal multimodal strategies from interaction with the user (Rieser & Lemon, 2010). Dialogue-handling expertise, whatever its theoretical motivation or particular implementation, can be incorporated into the system in the form of Agents that advise appropriate dialogue- and task-handling Managers. In this respect the QuADS architecture has been ‘future-proofed’ to meet eventual research demands at Queen’s, and accommodate the aspiration to adaptive, naturalistic, multimodal dialogue that is being expressed by researchers more generally (Geertzen et al., 2004).

For demonstration purposes we have included with the latest version of ISIS-NL a number of static user- and system profiles. For each user-profile, the system, according to its capabilities, exhibits different behaviours as it attempts to satisfy the user’s inquiry: these variations affect the number of choices that the system offers the user, and the modality in which it announces these choices (e.g. “I couldn’t find an exact match for your request. Here is the first option I have to suggest. Can you tell me if you want…?” etc.). We will examine the policy for composing these system turns and choosing their modality in Section 2.3.4. Whenever a match that satisfies or substantially satisfies the user’s information request is possible, the system presents video segments for the user to examine. Figure 1 gives a user’s view of the system in action as it reaches this stage: the system displays a selection of ‘thumbnail’ still images that represent the best matches for the inquiry; each time the user selects a thumbnail, he or she then uses the large window to review the corresponding video footage (‘Play video one.’ ‘Pause video.’ ‘Go forward five seconds.’).

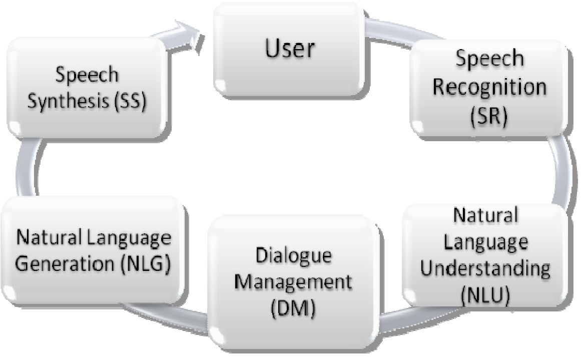


Figure 3. Main components of a typical spoken dialogue system.

2. THE QUADS ARCHITECTURE

2.1 THE BACKGROUND TO QUADS

The QuADS architecture draws on Dynamic Interpretation Theory, and in particular the DIT++ taxonomy of dialogue acts, proposed by Bunt (2000, 2007, 2009a). Dynamic Interpretation Theory is motivated by the insight that dialogue is rarely conducted for its own sake: rather, there is often an underlying goal (a need to accomplish something in the real world – like booking a ticket or finding a person). According to DIT, Dialogue acts are instrumental in achieving such goals: alongside dialogue acts whose role is to maintain the smooth interaction between the dialogue partners (giving feedback, managing turn-taking, dealing with matters of etiquette, etc.), there are also the core, task-related, information transfer acts that are used either to seek or to provide information, as well as the acts that are used to make offers and promises and to give instructions.

QuADS comprises an extensive collection of highly generic dialogue classes written in Java. These classes represent not just typical dialogue acts, of the type referred to above, but additionally provide a framework of dialogue management and task management classes within which developers may implement the mechanisms for maintaining a human-computer 'conversation' and progressing an underlying task. While the QuADS framework is not in itself prescriptive of particular dialogue- and task-management functionality, it provides outlines for some key components, whose interaction makes dialogue- and task-management possible. A representative selection of acts from the Dialogue Act hierarchy is shown in Figure 2.

2.2 THE MAIN COMPONENTS

The QuADS architecture encompasses the full cycle of processing in a spoken dialogue system, as shown in Figure 3. The key steps that are required to support naturalistic spoken exchanges between system and user are outlined in the sections that follow. Particular reference is made to the ISIS-NL implementation.

2.2.1 Speech recognition

The speech recognition component of the system converts the user-utterance to a text string. ISIS-NL uses the Automatic Speech Recogniser (ASR) supplied with Microsoft's Speech Application Program Interface (SAPI). Since the recogniser (once trained) works in the manner of a general, large-vocabulary dictation engine, very free spoken input is possible and the problem of out-of-vocabulary or ill-formed words at the recognition stage is greatly reduced. The possibility of misrecognition does, however, remain, and for this reason the system's Dialogue Manager (described in more detail below) has a range of confirmation and grounding tactics – for example, the spoken and textual content synopses that the system provides when it displays retrieved image/video content. The textual representation of the user's input, whether the input was originally spoken or keyed, is passed to the natural language understanding component.

2.2.2 Natural language understanding

To understand the significance of the user's input, ISIS-NL spots key words and phrases that are commonly used in the application domain, and that in context can be assumed to have typical pragmatic and semantic weight. (In a well-defined application area, where the possibilities of what the user will say are restricted and the intent behind them is easily understood, the natural language understanding task (NLU) of the overall dialogue system can be greatly simplified.) In QuADS-based applications, Dialogue Act objects represent both the user's speech act and its semantic content. They are objects of a particular type (an *Inform* act, a *Disambiguate* act, a *Specify* act and so on), and depending on the kind of act they represent, may contain *Elements* (small chunks of information, often sub-classed as problem-solving *Resources*) that express their full meaning – for instance, an *Inform* act, generated as a result of a user's utterance, may be used to tell the system that the user is looking for a *Bus Route* (a contained *Element* of sub-class *Resource* within the *Inform* object) and that the number of the *Bus Route* is *57*. We shall return to this example in Section 2.3.2. Currently in QuADS the input string is read by a series of word- and phrase-spotters, each of which can recognise concepts relevant to the domain and match these with the most likely dialogue act in the application context.

2.2.3 Dialogue and task management

Dialogue management determines how the system responds to information from the user, and sometimes to information emerging from its own sub-components. As it advances the dialogue towards task completion, the system introduces new dialogue acts and new sub-goals. In ISIS-NL, dialogue management involves filling task-related 'slots' of information and monitoring the confirmation status of that information (new, confirmed, modified, negated, etc.): in this sense the process has much in common with the frame-based approaches to dialogue management described by McTear (2002), and exemplified by Heisterkamp and McGlashan (1996). In ISIS-NL, the dialogue management task is broadened to include the choice of modality that the system uses when it communicates with the user: for this turn in the dialogue, will output be speech-based or text-based and will the modalities be used individually or in combination?

In ISIS-NL, responsibility for completing the underlying information retrieval task is shared between a closely collaborating 'team' of objects:-

- **The Dialogue Manager** itself, which is responsible for the system's task-independent dialogue progression. (When should the system confirm user-provided information? How should it confirm it? What happens when there is information to show or describe to the user, or a question to ask? ...And so on.) – In order to decide which dialogue management action it will perform next and in what manner, the Dialogue Manager has, respectively, a number of *advancement policies* (what to do next) and *realisation policies* (how to do it). In particular it uses its *realisation policies* to determine the modalities that will be used to realise a particular dialogue act, given a particular system configuration and user preferences.

- **The Task Manager**, which, as a class of problem solver, has the job of working out whether it has enough information from the user to attempt to answer his or her inquiry, or whether more information should be requested, and what that information should be. – Sitting alongside the broad sweep of the dialogue cycle shown in Figure 1, the Task Manager is invoked when its input is required by the Dialogue Manager, the latter having reached a point in its processing when it is able to turn to the 'real-world' task, as opposed to managing the corrections and confirmations of the 'communicative' task.

- **The Information Management Agent**, which interfaces with the system's database in order to retrieve the information that the Task Manager has requested, or, when a specific request fails, to examine the database more closely to see what information might be retrieved if certain inquiry constraints were relaxed. – The Information Management Agent is a domain specialist, encapsulating a combination of real-world expertise (How do I relax the constraints of this inquiry if I am not getting any hits?) and technical know-how (How do I formulate the request for this type of database?).

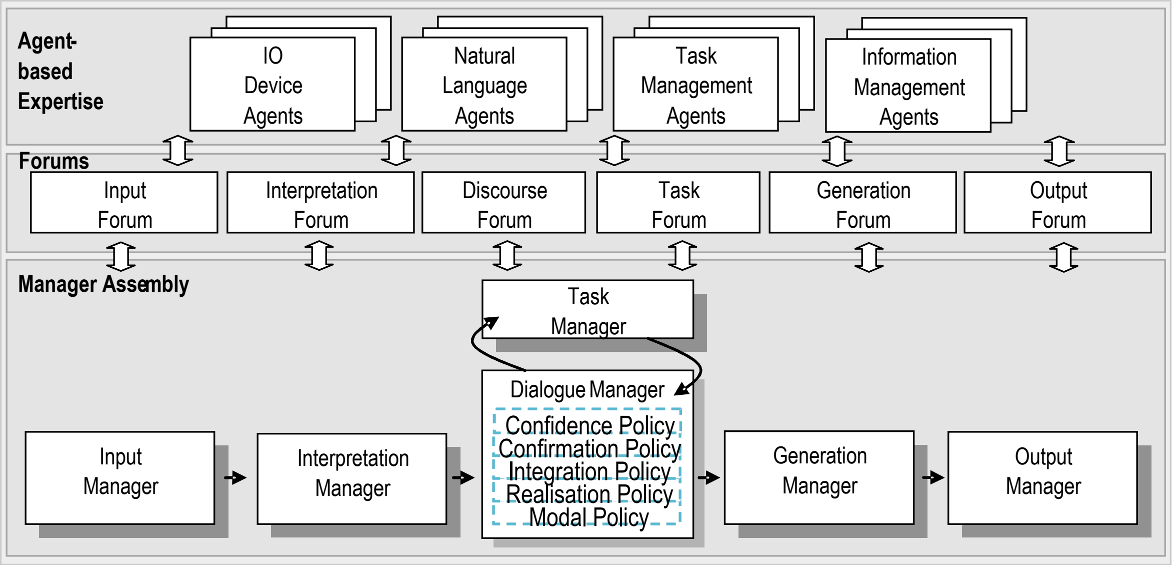


Figure 4. An overview of the QuADS system architecture.

2.2.4 Natural language generation and speech synthesis

Traditionally, any 'concepts' or dialogue acts that are generated into well-formed natural language strings (the role of the NLG component) are also spoken by the speech synthesiser. However, in the multimodal QuADS environment, a generated string may be spoken, or it may simply remain as text and be displayed as such, or a combination of both text and speech may be required. QuADS first 'internally' generates the text representing a potential dialogue act, and then assesses how it should be realised, according to system capabilities and user preferences. Indeed, it may reduce the amount of information that is generated, in order to accommodate the output capabilities of the system. In Sections 2.3.3 and 2.3.4 we will further consider this process of realising output.

2.3 Using QuADS to maintain a basic multimodal dialogue

2.3.1 The QuADS architecture in context

In QuADS the dialogue cycle of Figure 3 is maintained by a group of *Manager* components operating within an *Assembly*, where each manager is called on in sequence – the *Assembly Sequence* – to provide its contribution to the overall dialogue-handling task. In addition, each *Manager* is associated with a particular *Forum*, where it has at its disposal one or more specialised *Agents* (implemented as software objects) to help it with its dialogue-related task. We have previously discussed our interpretation of 'agents' as inheriting, collaborating dialogue-handling experts, implemented as software objects, instances of classes from an object-oriented hierarchy (O'Neill et al., 2004). The forum-based architecture has much in common with the blackboard model, which has been successfully used elsewhere (in terms of dialogue systems perhaps most notably in SmartKom (Wahlster, 2002)). However, whereas the 'blackboard' represents a shared information repository that various agents periodically consult, the forum facilitates more direct interaction between the system's object-components: members of the same forum can interact with one another. For example, the *Generation Manager*, which is responsible for processing any acts that require to be formed or generated before they are output, is associated with the *Generation* forum, as is an *NLG Agent*, which, via the shared forum, is able to provide the *Generation Manager* with well-formed phrases and sentences whenever they are required. The main features of the QuADS architecture are outlined in Figure 4.

From a historical perspective, the assembly-and-forum architecture of QuADS supersedes the hub-and-spoke architecture used by the Queen's Communicator, which, though not a DARPA project itself, used the same Galaxy

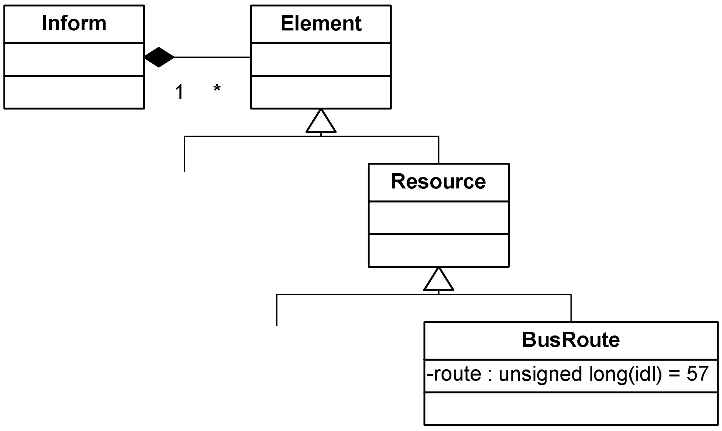


Figure 5. An example of an Inform dialogue act and its components.

Communicator hub as was used by the dialogue systems participating in the DARPA Communicator project[1]. (The Galaxy Communicator hub was developed by the Spoken Language Systems group at MIT and subsequently extended and released as an open source package by the MITRE Corporation (Bayer et al., 2001).) In the 'Communicator' configuration, system components (similar to the system components shown in Figure 3) interact with each other using information that they package and send through the hub as 'hubframes'. A 'hub-script' maintains correct sequences of interaction. Other applications have made use of similar configurations, sometimes referring to the single or multiple hub-like co-ordinating entities in their architectures as 'facilitators', 'facilitator servers' or even 'facilitator agents'. Cheyer and Julia (1995) describe their use of the then relatively new Open Agent Architecture (OAA) and its Interagent Communication Language (ICL) (Cohen et al., 1994) to enable task-specific agents to communicate with each other via facilitator agents in the context of a multimodal travel planning application. From a multimodal perspective Cheyer and Julia's work was particularly interesting in that it tackled the problem (an on-going challenge!) of resolving references that were supplied in different modalities (e.g. speech and gesture) and handled by different modality agents, but that referred to the same real-world object. MATCH (Johnson et al., 2002), is a further example of a system built around a central software 'facilitator' (in this case a message-passing component known as 'MCUBE'). Developed at AT&T, MATCH (Multimodal Access to City Help) also set out to address the problems of appropriately generating, receiving and fusing (for better understanding) information in different modalities, on this occasion taking restaurant and subway information as its application domain. (The developers of MATCH (Johnson et al., ibid.) liken the functionality of their MCUBE facilitator specifically to the functionality provided by the OAA and the Communicator hub.)

Architectures involving a central hub or one or more facilitators have much to commend them as a means of coordinating a range of dialogue system components, each with its own task to perform in its own particular modality. However, in moving away from a hub- or router-based configuration, we are exploiting instead the very free, method-calling interaction that characterises object-oriented systems. The use of forums in association with an assembly, as a means of bringing together collaborating system components, opens up a number of interesting possibilities, creating more flexible and richer interactions than the *DialogueManager - Problem Solving Manager* interaction that we proposed in Chu et al. (2005), an earlier exploration of an extension to the Queen's Communicator object hierarchy. In the new QuADS architecture, different Managers might, for example, have different Agents at their disposal, so that tasks may be completed in a number of ways. For instance, requests for output might be reinterpreted by Agents that represent human-computer interfaces with very different capabilities, some biased towards use of text, others towards use of audio or video, and so on. Moreover, the forum-based architecture means that information exchange in the dialogue cycle is not necessarily swept all in one direction, from input to output: by making use of Agents in a forum other than the one with which they might most immediately be associated, Managers can incorporate information from any stage of the dialogue cycle into their own decision-making. If, for example, the Dialogue Manager was informed that audio output was being heavily used, it might start

[1] http://groups.csail.mit.edu/sls//technologies/galaxy.shtml

to gear its dialogue acts towards simple visual output, and so on. And while the QuADS architecture supports predominantly server-side functionality, that receives, processes and outputs information coming from and going to a range of front-end devices, these front-end clients are easily interfaced to the server via IP and port addresses, in the manner typical of current client-server configurations.

In the present paper we have already mentioned use of agents in multimodal systems. On previous occasions we have alluded to examples of their use, and pointed out that, in contrast to the approach we adopted in the Queen's Communicator and now in ISIS-NL (where a single agent-object might embody expertise for a substantial real-world, user-system interaction – e.g. ticket-booking), agents in other dialogue systems sometimes perform very simple tasks. Turunen and Hakulinen (2001), for example, describe simple generic error-handling agents that ask the user to repeat misunderstood input; Cheyer and Julia (1995) describe collaborating 'macro agents and 'micro agents', the former having more substantial knowledge and ability for domain-specific reasoning, the latter typically handling fine-grained input in a particular modality. Elsewhere we have described how the domain experts or agents in the Queen's Communicator – each of which was, via inheritance, a 'dialogue manager' in its own right – used their sequences of 'expert rules' to ask the user for the information needed to complete routine, frame-based transactional inquiries (O'Neill et al. 2003, 2004, 2005).

In the Queen's Communicator our concern to support maintainability and extensibility, through the use of inheriting and collaborating agent-objects, was characteristic of the software engineer (Hanna et al., 2007). In QuADS, our attempt to capture dialogue acts as families of objects that can be interpreted and acted upon by assemblies of Managers and their Agents, is similarly motivated by an aspiration to good software design (Hanna et al., 2009). In terms of functionality, our main concern in both the Queen's Communicator and ISIS-NL is with the successful completion of the underlying real-world task: the system performs successfully if its information- and confirmation-requests can be understood and acted on by the user, and if the user is presented with information (details of a completed hotel- or theatre-booking in the case of the Queen's Communicator, relevant video footage in the case of ISIS-NL), that matches the supplied or inferred constraints of his or her inquiry.

However, other researchers have been more closely concerned with refining communicative efficiency, and in so doing increasing the naturalness, of the agent-based dialogue system's utterances. One notable example in this regard is PARADIME (Parallel Agent-based Dialogue Management Engine), funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) as part of the IMIX information extraction multi-project[2]. Bunt's DIT taxonomy captures the many dimensions into which dialogue acts may fall: task-related questions, instructions, requests; auto-feedback; allo-feedback; and so on. In the course of dialogue management in PARADIME, specialist *Dialogue Act Agents* each have the opportunity to generate a candidate dialogue act for the next system turn in their own area of competence: an *Evaluation Agent* then determines how the acts should be scheduled, perhaps over several turns, or whether some acts should be dropped from the candidate list (since they are implied by other candidate acts), or indeed whether several acts can be combined into what becomes a single multidimensional system utterance (Keizer & Bunt, 2006, 2007). The developers also take into consideration the possibility that an act might be realised non-verbally – by some event on a graphical user interface, for example.

While ISIS-NL does not yet embody PARADIME's sensitivity to dialogue act realisation, its architecture provides considerable scope for the incorporation of such expertise. The assembly-forum architecture of QuADS envisages a rich collaborative interaction between components. Thus, for example, a Generation Manager (possibly one that learns and adapts) might resolve questions concerning choice of modality by assessing the options proposed in the Generation Forum by a team of information- and context-sensitive Generation Agents.

2.3.2 Reacting to input

In our current QuADS-based implementation, ISIS-NL, for each concept that is recognised, the NLU Manager creates a new dialogue act, from within the taxonomy of act classes available to the system. Let us consider the case where a bus route number is spotted in the input phrase

"... A person who got on the number 57 bus..." (1)

First a problem-solving *Resource* object is created: a *BusRoute*, that in this example takes as an attribute the spotted *route* (*57*). Again, in a very basic dialogue scenario, where the universe of discourse is closely bounded and a limited range of dialogue acts is expected from the user, simple word- and phrase-spotting is sufficient to identify the

[2] http://wwwhome.ewi.utwente.nl/~hofs/imix/index.html

dialogic force behind the user's utterance: the ISIS-NL system is intended to retrieve footage that satisfies certain constraints, and so phrase (1) may be safely interpreted as the user *informing* the system about the kind of footage it should retrieve. The *BusRoute* (with its *route* attribute) itself becomes an attribute of a Dialogue Act object of type *Inform*. Figure 5 outlines this relationship. Our use of resources for problem solving in dialogue is influenced by the work of Blaylock et al. (2003, 2005). The following pseudocode represents the process of creating an object-based *Inform* act, from information 'spotted' in the user utterance.

```
Create new BusRoute instance called busRouteSpot;
Set Route number in busRouteSpot to the busNumber spotted;
Create new Inform instance called busRouteInform;
Set data Element in busRouteInform to busRouteSpot;
Add busRouteInform to the Acts created for this user-turn;
```

How does the Dialogue Manager respond when it receives such an *Inform* act? The system has to be able to cope with quite a fluid evolution of the dialogue, where, for example, the task-supporting elements that the user supplies in his or her *Inform* utterances may contradict or negate what went before ("I meant the number *39* bus." "*Not* the number *57*.") To deal with these situations the system, along with its *advancement* and *realisation* policies, has an *integration policy* that enables it first to check whether value types provided by the user can be mapped neatly on to slots relevant to the current tasks, or whether the significance of the values provided must be disambiguated – the latter requiring further interaction with the user. The rules of the *integration policy* further enable the dialogue manager to deal with those questions of user-supplied values that change from turn to turn: in some cases the system has to make the user aware that it has noticed the change, while in other cases the Dialogue Manager might simply assimilate the change and allow the dialogue to proceed.

From the point of view of multimodal output generation, the system's behaviour is perhaps at its most interesting when it has acquired a "full set" of information, enough to attempt the information retrieval task and let the user know, in whatever modalities are appropriate and possible, what it has found or, indeed, what alternatives it has to suggest when a user requirement cannot be satisfied.

2.3.3 Realising output

Assuming that the system has found information that it wants to show to the user, it must decide, in the current configuration, whether to output its commentary on that information as text or speech, or some combination of these. Though the development system is PC-based, we consider the possibility that it may eventually run on or be interfaced to a variety of devices, including mobile or in-vehicle devices, which, because of a very small or basic screen (an LCD for instance), may be able to display only a limited number of words per turn. The system therefore takes into account the capabilities (modalities) and capacities (maximum output per turn) of the device, before it realises output for a particular turn. It also takes into account any preferences for a particular modality that the user may have expressed, or that may be indicated by the user's profile. In our development system, both user- and device-profiles are passed to the QuADS-based Dialogue Manager by the front-end client.

In the current system, for a particular dialogue act (for example a *Specify* act, where the system wishes to present the user with a number of options to select from), all the *Elements* (options, in this instance) that the Dialogue Manager has included in the act are generated and *prepared* for output. However, before the act is output or 'realised', the Dialogue Manager applies its *modal policies* to the act. These *modal policies* take into account device capabilities and user preferences, and enable the Dialogue Manager to determine which modality will be used and how much information will be conveyed to the user on that turn. A single system turn may comprise several dialogue acts, each of which will be generated in the most appropriate modality for user and system. Currently, if different acts are to be generated in the same turn and require the same modality, those acts are generated sequentially in the particular modality: we do not yet attempt to rationalise utterances by identifying opportunities for 'simultaneous multifunctionality' – where, for example, a phrase like 'let me see' might realise a turn-taking act and a request for some thinking time (Bunt, 2009b).

In an alternative configuration, it is true, the Dialogue Manager might at the very start of each turn restrict the potential complexity of the turn; or it might give the NLG Manager a dialogue act that comprises multiple elements, but instruct the NLG Manager to generate the act in chunks that can be output over several system turns. While we recognise these as valuable options, which are likely to be the subject of experimentation in the future, our present approach is sufficient to illustrate the concept of adaptation to user and system: the system (internally) generates one complete dialogue output for the particular turn and then uses this output – in particular its size – to work out how the information will be presented to the user.

2.3.4 Selecting the output modality

At present our research and development concerning modality selection has concentrated on one particular facet of the dialogue scenario: the manner in which the system presents alternatives to the user if an initial user-inquiry has failed. The fact that the system has to present to the user one or more alternatives (or potentially apologise and offer none) makes this a useful and manageable context in which to experiment with chunking information over several turns and across different modalities. In formulating an initial, *potential* response to the user, the Dialogue Manager internally generates an array of dialogue acts. These are subjected to the Dialogue Manager's *modal policies*, so as to determine the manner in which each act will be realised and the amount of information that will be presented to the user by each act. Again, it may be that several dialogue acts are realised in a single system turn. In practice dialogue acts are relatively short – typically informing the user of decisions that the system has made and asking the user to specify additional information. Enactment of our *modal policy* includes the following steps – and here we will focus on the modality selection process for a *Specify* act that potentially entails a number of options:

1. The system determines the modalities that can be used to realise the act. Each type of *Resource* that can be associated with an act (e.g. a *BusRoute* that the system prompts the user to *Specify*) has, as an attribute, one or more modalities that may be used to realise it (unless stated otherwise, text and speech are the default modalities for realisation; a non-verbal *Resource*, like video, will require the corresponding non-verbal modality for realisation). Thus the *modal policy* considers the *Resource*s associated with a particular act with a view to identifying modalities suitable for realising the act. At this stage appropriate modalities are identified, regardless of their availability for a particular device, and regardless of suitability for a particular user.

2. The system confirms which modalities are available for the device that it is running on or interfaced to. This may mean that a modality that, in theory, *could* be used for a Dialogue Act, is now deemed unavailable in a particular system environment.

3. The system works out how much information can be handled by each modality on this device. In our initial implementation this is quite a simple metric – representing on a scale of 0 to 1 the degree to which the internally generated act (the potential output for the act) can be accommodated by the particular modality. Let us consider the case of a string that may be realised as text or speech. Since the internally generated string has a length measured in characters, and the modality has an *available length* that it can accommodate (set as a number of characters by the developer), it is easy to calculate the degree of *information load* according to the following sigmoid function (in the case of still images to be displayed as a bank of 'thumbnails', the number of images may be used as units of length; likewise, if it is possible to stream videos simultaneously, the number of videos may be used):

$$\textit{information load} = 1.0 \,/\, (1.0 + e^{\,(-\,5.0\,*\,(1.0\,-\,\textit{internally generated length}\,/\,\textit{available length})}\,)$$

4. The system next considers the user's preferences for each of the available modalities (represented in each case as *strongly like*, *like*, *neutral*, *dislike*, *strongly dislike* and *never use*.) Each of these preference types is associated with a scaling factor, a simple multiplier, that is applied to the *information load* value to create a *load preference* value: less-liked modalities receive proportionately lower *load preference* values. The *load preference* values for each modality are compared, and the winning modality – the primary modality in which the act will be realised – is the one with the highest score. (The system may be configured to allow secondary modalities to be used alongside this primary modality, so that, for example a textual output may also be spoken.)

5. In a case where the *information load* (from step 3 above) is less than 0.5 (which we take as an indication that there is too much information to convey to the user in a single turn in this modality), the system will reduce the information content of the act for this turn – setting the remaining original content temporarily

aside with a view to making it available, if the user wishes to see or hear it, over what might become several turns.

For example, if the system was planning to ask the user to choose between several options in this turn – each option being represented by a *Resource* within the *Specify* act – it may shorten the content of the act by progressively removing *Resource*s from the act and generating and re-examining the act until eventually it may decide to ask the user to choose or reject just one option on the turn. An *Inform* act accompanying the newly shortened *Specify* act will make it clear to the user that the system is presenting just the first of several options, so affording the user the opportunity to ask for further options, i.e. the *Resources* that have been (temporarily) removed on this system turn. Only at this point is the realisation process set in train, using the selected modality, with an appropriate amount of content, for the particular user. If, because of the influence of the user's preferences, a modality is selected that will require the dialogue act to be split over several turns, that is an acceptable outcome: the user's preferences are accommodated, even if a more complex sequence of dialogue interaction ensues.

The effect of the modality selection algorithm and the *information load* calculation, will, depending on the user's preferences and the system's capabilities, give rise to outputs like *Large Capability – Sequence A* and *Limited Capability – Sequence A* (shown below), which are taken unedited from the current implementation. This example is for one modality (text) and shows the effect that large and limited output capability in the selected modality has on the system's realisation of the dialogue acts. The system is announcing that it has been unable to find an exact match for the user's request (which it echoes for grounding purposes) but, by relaxing some of the user-supplied constraints, has been able to find other matches that the user may wish to consider. In each of the Sequence A transcriptions, turn *System 1b* (*Inform)* is the realisation of a system-internal *Report* that is intended to provide the user with a commentary on the system's actions: in the *Large Capability* sequence, this *Report* corresponds to the concept 'Specify Choice Provided' (i.e. all options are being presented); while in the *Limited Capability* sequence the *Report* corresponds to the concept 'Specify Choice Partial Initial' (i.e. the first of a range of options are being presented).

Large Capability – Sequence A

System 1a (*Inform*): So that's entering the number 45 bus , starting search at 1:00 , ending search at 18:15 on the fifth of October.

System 1b (*Inform*): - Sorry. I'm unable to match that, but I have 2 alternatives.

System 1c (*Specify*): Can you tell me if you want: option 1, an alternative bus event; or option 2, an alternative bus route?

Limited Capability – Sequence A

System 1a (*Inform*): So that's entering the number 45 bus , starting search at 1:00 , ending search at 18:15 on the fifth of October.

System 1b (*Inform*): I couldn't find an exact match for your request. Here is the first option I have to suggest.

System 1c (*Specify*): Can you tell me if you want: option 1, an alternative bus event?

The system can be adjusted to produce even sparser output for a device of limited capability. Omitting the opening confirmation turn from a limited capability sequence, and generating 'Specify Choice Partial Initial' still more tersely (in turn *System 1a*), gives the following output (*Sequence B*):-

Limited Capability – Sequence B

System 1a (*Inform*): No exact match found. Options follow.

System 1b (*Specify*): Can you tell me if you want: option 1, an alternative bus event?

3 LOOKING AHEAD

This research marks an early stage in the process of assessing the ways in which the QuADS architecture might be exploited and extended, and although our implementation for ISIS-NL answers user-inquiries through a combination of speech, text, still image and video, we have yet to explore the most effective balance of these modalities for different user-types in different contexts. At present, for example, it is assumed that the user will always want to see an overview (in the form of small video stills or 'thumbnails') of the video content that matches his or her inquiry, before choosing a portion of video to play. However, even this process of video selection may be subject to a number of multimodal permutations: the user might want to read or hear a synopsis of what has been found, before examining thumbnails or videos; or the user might want to read a synopsis alongside each thumbnail; or hear a synopsis for each thumbnail on request.

The most appropriate, indeed the most satisfactory functionality will differ between applications, system configurations, deployment contexts and of course the possibly idiosyncratic preferences of individual users. Part of our ongoing work will be to put in place the mechanisms that allow multimodal output to be realised as flexibly as possible, and then to give the user the opportunity to tailor the system to their particular requirements. While a GUI-based options panel would give a user an immediate means of communicating presentation preferences, in the longer term it is likely that the system's understanding of user preferences will become more closely tied to the user-system dialogue itself, whether the system asks for guidance directly ('How am I doing?'), or whether it 'intelligently' infers best practice from an analysis of user profiles and user preferences observed from live transactions. Already in the QuADS architecture many important building blocks are in place to help realise these more advanced system behaviours.

REFERENCES

Bayer, S., Doran, C., George, B. (2001). Dialogue Interaction with the DARPA Communicator Infrastructure: The Development of Useful Software. In *Proceedings of the first international conference on human language technology research*, pages 1-3, San Diego, 2001.

Bernsen, N.O. and Dybkjær, L. (1999). A theory of speech in multimodal systems. In *IDS-99*, pages 105-108, ISCA, 1999.

Blaylock, N., Allen, J., and Ferguson, G. (2003). Managing Communicative Intentions with Collaborative Problem Solving. In *Current and New Directions in Discourse and Dialogue*, Kluwer Academic Publishers, Dordrecht, 2003

Blaylock, N. and Allen, J. (2005). A Collaborative Problem-Solving Model of Dialogue. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue,* pages 200-211, Lisbon, 2005.

Booch, G., Maksimchuk, R. A., Engel, M.W., Young, B. J., Conallen, J., Houston, K. A. (2007). Object Oriented Analysis and Design with Applications (3rd Edition), Addison-Wesley, 2007.

Bunt, H. (2000). Dynamic Interpretation and Dialogue Theory. In: M.M. Taylor, D.G. Bouwhuis & F. Neel (eds.) *The Structure of Multimodal Dialogue*, Vol 2., pages 139-166, John Benjamins, Amsterdam, 2000.

Bunt, H. (2007). DIT++ Taxonomy of Dialogue Acts. http://dit.uvt.nl/.

Bunt, H. (2009a). The DIT. In D. Heylen, C. Pelachaud, R. Catizone and D. Traum (eds*.) Proceedings of EDAML@AAMAS Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts"*, pages 13-24, Budapest, May 2009.

Bunt, H. (2009b). Multifunctionality and multidimensional dialogue semantics. In *Proceedings of the DiaHolmia 2009 Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm, June 2009.

Cheyer, A, and Julia, L. (1995). Multimodal Maps: An Agent-based Approach. In *Proceedings of the International Conference on Cooperative Multimodal Communication (CMC/95),* Eindhoven, May 1995.

Chu, S.-W., O'Neill, I., Hanna, P., and McTear, M. (2005). A Multi-Strategy Approach to Dialogue Management. In *Proceedings of Interspeech 2005*, pages 865-868, Lisbon, 2005.

Cohen, P., Cheyer, A., Wang, M., and Baeg, S. (1994). An Open Agent Architecture. In *Proceedings of AAAI Spring Symposium,* pages 1-8, Stanford, March 1994.

Geertzen, J., Girard, Y., Morante, R., Van der Sluis, I., Van Dam, H., Suijkerbuijk, B., Van der Werf, R. and Bunt H. (2004). The Diamond Project. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue, Catalog '04*, Barcelona, July 2004.

Hanna, P. (2008). QuADS Developer's Guide, Version 0.05, April 25th 2008. Queen's University, Internal Research Documentation – available on request, subject to agreement.

Hanna, P., O'Neill, I., Stewart, D., Qasemizadeh, B. (2009). Development of a Java-based unified and flexible natural language discourse system. In *Proceedings of the 7th International Conference on Principles and Practice of Programming in Java*, Calgary, Alberta, 2009.

Hanna, P., O'Neill, I., Wootton, C., McTear, M. (2007). Promoting extension and reuse in a spoken dialog manager: An evaluation of the queen's communicator. In *Transactions on Speech and Language Processing (TSLP)*, Volume 4 Issue 3, July 2007

Heisterkamp, P. and McGlashan, S. (1996). Units of dialogue management: an example. In *ICSLP-1996,* pages 200-203.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., Maloor, P. (2002). MATCH: An Architecture for Multimodal Dialogue Systems. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 376-383, Philadelphia, July 2002.

Keizer, S. and Bunt, H. (2006). Multidimensional Dialogue Management. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 37-45, Sydney, July 2006.

Keizer, S. and Bunt, H. (2007). Evaluating combinations of dialogue acts for generation. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 158-165, Antwerp, September 2007.

Light, M. and Maybury, M. (2002). Personalised Multimedia Information Access. In *Communications of the ACM*, Volume 45, Number 5, pages 54-59, May 2002.

Maybury, M. (1992). Intelligent Multimedia Interfaces. In *AI Magazine*, Volume 13, Number 2, AAAI, 1992.

McTear, M. (2002). Spoken Dialogue Technology: Enabling the Conversational User Interface. In *ACM Computing Surveys*, 34(1):90-169, 2002

O'Neill, I., Hanna, P., Liu, X. and McTear, M. (2003). The Queen's Communicator: An Object-Oriented Dialogue Manager. In *Proceedings of EUROSPEECH-2003*, pages 593-596, Geneva, 2003.

O'Neill, I., Hanna, P., Liu, X. and McTear, M. (2004). The Queen's Agents: Using Collaborating Object-Based Dialogue Agents in the Queen's Communicator. In *Proceedings of Coling 2004*, pages 127-133, Geneva, August 2004.

O'Neill, I., Hanna, P., Liu, X., Greer, D. and McTear, M. (2005). Implementing advanced spoken dialogue management in Java. In *Science of Computer Programming*, pages 99-124 ,Volume 54, Issue 1, 2005.

Oviatt, S., Coulston, R., and Lunsford, R. (2004). When do we interact Multimodally? Cognitive load and multimodal communication patterns. In *Proceedings of the 6th ICMI-04.*

Rieser, V. and Lemon, O. (2010). Learning human multimodal dialogue strategies. In *Natural Language Engineering* 16:3-23, Cambridge University Press, 2010.

Van der Sluis, I., P. Piwek, A. Gatt and A. Bangerter (2008). Multimodal Referring Expressions in Dialogue. In *Proceedings of the Symposium on Multimodal Output Generation (MOG 2008)*, Aberdeen, April 2008.

Wahlster, W. (2002). SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions.
In: *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, pages 213-225
Kyoto (Japan), November 2002.

A Demonstration of Continuous Interaction with Elckerlyc

Herwin van Welbergen, Dennis Reidsma, Job Zwiers
Human Media Interaction
University of Twente, the Netherlands
{welberge|dennisr|zwiers}@ewi.utwente.nl

Abstract

We discuss behavior planning in the style of the SAIBA framework for continuous (as opposed to turn-based) interaction. Such interaction requires the real-time application of minor shape or timing modifications of running behavior and anticipation of behavior of a (human) interaction partner. We discuss how behavior (re)planning and on-the-fly parameter modification fit into the current SAIBA framework, and what type of language or architecture extensions might be necessary. Our BML realizer Elckerlyc provides flexible mechanisms for both the specification and the execution of modifications to running behavior. We show how these mechanisms are used in a virtual trainer and two turn taking scenarios.

Keywords: Continuous interaction, SAIBA, BML realization, virtual humans

1 Introduction

Virtual humans often interact with users using a combination of speech with gestures in a conversational setting. They tended to be developed using a turn-based interaction paradigm, but this is changing towards a *continuous* interaction paradigm, where actors perceive acts and speech of others continuously, and where actors can act continuously, simultaneously and therefore overlapping in time (Nijholt et al., 2008). This raises the question how acting has to be planned; clearly the traditional "perceive-plan-act" cycle from agent theories does not apply here. Still, it is clear that (human as well as virtual) actors *do* perform some form of "action planning", both for the long term as well as for the short term. Behavior of interaction partners (real or virtual) is dealt with by, on the one hand, *predicting* such behavior and, on the other hand, *re-planning* and *modifying* the virtual human's own behavior. There are various forms of this revision process, some more disruptive than others. As an example, consider a fitness trainer that has just identified a problem in that the group he is coaching starts "lagging" and does no longer follow the correct tempo of their exercise. A disruptive revision would be to stop the exercise, explain what went wrong, and to start over again. Although this is certainly a possible solution, we propose a more subtle, preferred, approach, where the trainer starts moving a little faster, and a little ahead of the group in order to try to speed up the tempo of the group. Within the SAIBA framework for behavior planning, shown in Figure 2, the first approach requires a revision of intents, followed by re-planning speech and bodily behavior.

Our alternative approach circumvents this and applies a more direct revision of bodily behavior, based upon (short term) prediction by means of so called *Anticipators*, combined with corrective adjustments of already ongoing behavior. This leads to a flexible planning approach in which part of the planning can be done beforehand, and part has to be done "on the fly". In the latter case, parts of the behavior have been executed already, and other parts can still be modified. We focus on the specification (both of the plan itself and of changes to the plan) and execution of such flexible plans. We provide abstractions for the prediction of sensor input and show how we can synchronize our output to these predictions in a flexible manner. To demonstrate the feasibility of the multimodal output generation part of our system without having to invest a lot of work in the sensing part, we have implemented placeholders for the predictors.

In this paper, we present several scenarios in which continuous interaction is achieved using small adjustments in the timing and shape of behaviors (for example: gesture or speech) while it is being executed by a virtual human. We show how such small adjustments can be specified and how we implemented these behaviors in our behavior realizer Elckerlyc. We intend to demonstrate our implementation in the demo session of the workshop.

2 Elckerlyc's Architecture

We base our architecture (see Figure 2) on the SAIBA Framework (Kopp et al., 2006), which contains a three-stage process: *communicative intent planning*, multimodal *behavior planning*, resulting in a BML stream, and *behavior realization* of this stream. Elckerlyc encompasses the realization stage. It takes a specification of the intended behavior of a virtual human written in the Behavior Markup Language (BML) (Kopp et al., 2006) and executes this behavior through the virtual human. The BML stream contains behaviors (such as speech, gesture, head movement etc.) and specifies how these behaviors are synchronized. Synchronization of the behaviors to each other is done through BML *constraints* that link synchronization points in one behavior (start, end, stroke, etc; see also Figure 1) to synchronization points in another behavior. BML can be used to add new behaviors or remove running behaviors, but does not contain mechanisms to slightly modify behavior that is already running. However, we argue that some desired changes to planned behavior are only on their timing or parameter values (speak louder, increase gesture amplitude) and should not lead to completely rebuilding the animation or speech plan. Such small adaptations of the timing of or shape of planned behavior occur in conversations and other interactions (Nijholt et al., 2008).

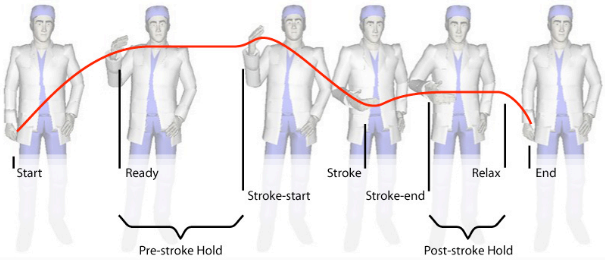


Figure 1: Standard BML synchronization points (picture from `http://wiki.mindmakers.org/projects:bml:main`)

There is typically a planning delay between sending a BML stream to the Behavior Realizer (Elckerlyc in our case) and the realization of this stream. By fine-tuning running or planned behaviors rather than re-planning the complete behavior plan when only small changes are required, we avoid this planning delay and allow fluent and timely behavior execution. Others have used similar mechanisms for incremental planning in gesture/speech synthesis: Kopp and Wachsmuth (2004) make use of an incremental planning mechanisms that allows the late planning of transitions between segments of gesture and speech, which are highly context dependent (depending on current gesture and the next gesture), but for which some parts can be pre-planned (e.g. the speech synthesis). In human-human behavior, there is some evidence of similar pre-planning mechanisms (Nijholt et al., 2008), for example to allow rapid overlaps between turns in a dialog. We extend BML to allow the specification of synchronization to anticipated timing of external events (from the environment, or other (virtual) humans). Elckerlyc allows partial pre-planning of behavior that is timed to such events. The timing of such behavior is refined and completed continuously, while keeping inter-behavior constraints consistent.

To achieve this incremental temporal control, we introduced *Time Pegs* and *Anticipators*. BML specifies constraints between behaviors, indicating that their synchronization points should occur at the same time. We maintain a list of Time Pegs – symbolically linked to those synchronization points that are constrained to be on the same time – on the *Peg Board*, together with the current

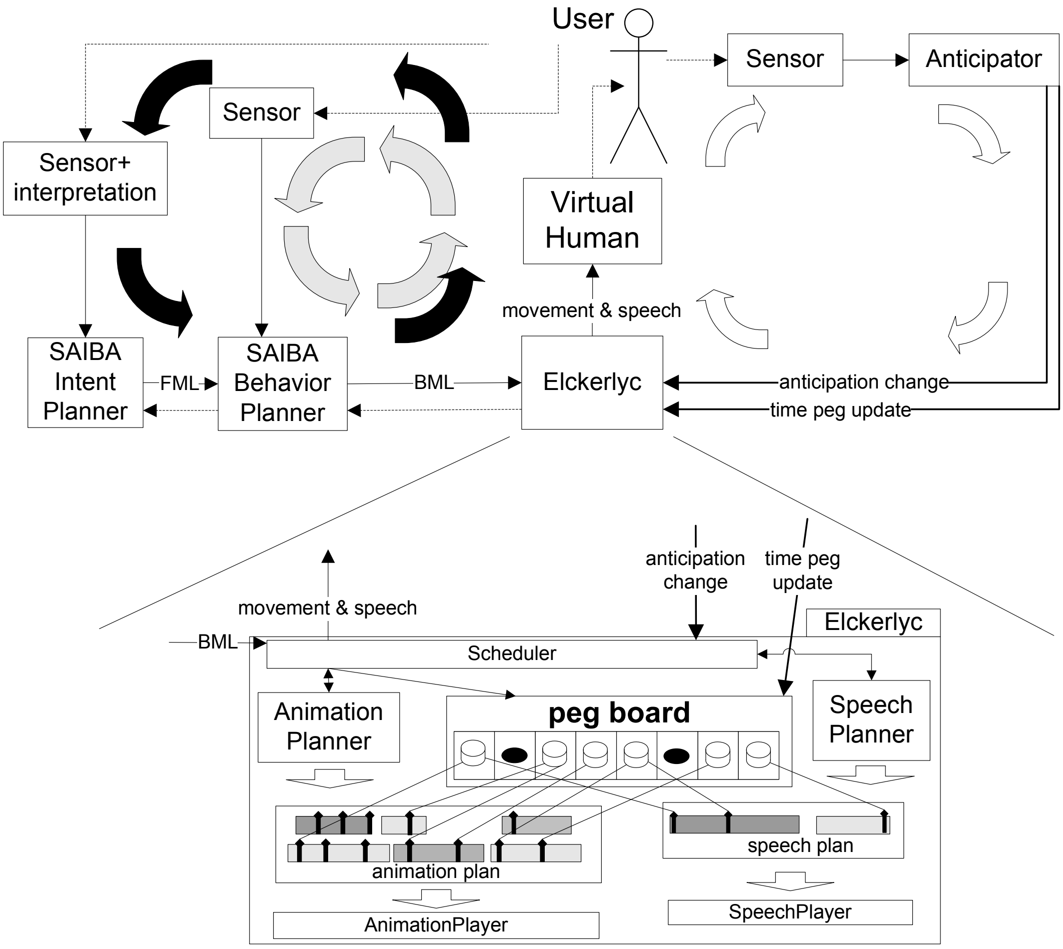
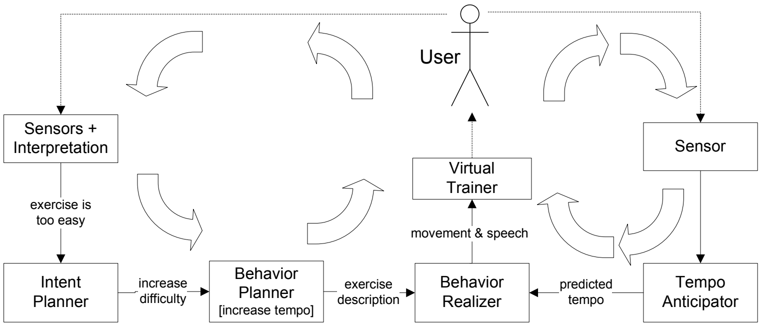


Figure 2: Elckerlyc architecture and its location within the SAIBA framework

expectation of their actual execution time (which may change at a later time and can be unknown).

Interaction with the world – and conversation partner – is achieved through Anticipators. An Anticipator instantiates Time Pegs that can be used in the BML stream to constrain the timing of behaviors. This is specified in a similar manner as BML constraints: a synchronization point of a behavior is linked to the synchronization point of an Anticipator (as identified by the Anticipator id and its TimePeg id, see Figure 3, 4 for some examples). The Anticipator uses sensors that perceive events in the real world to continuously update the Time Pegs, by extrapolating the perceptions into predictions of the timing of future events.

Several feedback loops between user and agent behavior can exist in the SAIBA framework. The SAIBA Intent Planner makes use of *interpreted* user behavior to decide on the Intent of actions that are to be executed by the virtual human (indicated with the black arrows in Figure 2). Bevacqua et al. (2009) argue for another feedback loop (indicated with the gray arrows), using sensor-activated unconscious and unintentional (so not originating from the Intent Planner) behavior in the Behavior Planner. One example of such behavior is mimicry, which they propose to implement by submitting new BML to the Realizer, which then has to be re-planned. In this paper we demonstrate the need for an even tighter feedback loop (indicated with the white arrows) which allows small modifications based on user observations to be made to running behaviors directly, without the need for re-planning behavior. Similar layered feedback loops between a user and a virtual human occur in the Ymir system (Thórisson, 2002).



```
<bml id="bml1">
  <bmlt:procanimation id="exercise1" name="squat-jump"/>
  <constraint id="c1">
    <synchronize ref="exercise1:squatdown">
      <sync ref="exerciseAnticipator:squatdown-0.5"/>
    </synchronize>
    <synchronize ref="exercise1:jumpstart">
      <sync ref="exerciseAnticipator:jumpstart-0.5"/>
    </synchronize>
    ...
  </constraint>
</bml>
```

Figure 3: Exercise scenario. `exercise1:squatdown` and `exercise1:jumpstart` refer to the squated down position and the start of the jump in the squat-jump exercise animation respectively. `exerciseAnticipator:squatdown` and `exerciseAnticipator:jumpstart` refer to the anticipated timing of squatdown and jumpstart as predicted by movement of the user.

Elckerlyc can be used as a black box that converts BML into multi-modal behavior for a VH.[1] If required however, direct access to the Scheduler, Planners, Plans and Players is also available. Some of this functionality is used in the demo scenarios described in this paper to adapt the parameter values of ongoing behavior (e.g. speak louder). We refer the reader to (van Welbergen et al., 2010) for a extensive explanation of Elckerlyc's architecture.

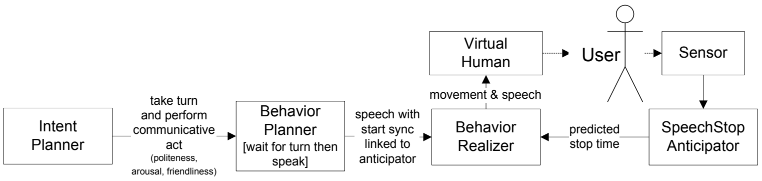
3 Scenarios

3.1 Guiding Exercise Tempo

A virtual (fitness) trainer executes an exercise together with a human user in a certain tempo. The trainer would like to increase the tempo that the user is moving in. A subtle technique to achieve this is to move in the same tempo as the user but slightly ahead of him, so he constantly has the feeling of being 'too late' in his movements (a similar technique is used by our virtual conductor to guide the tempo of a real orchestra (Reidsma et al., 2008)). We assume that an Anticipator can be designed that can perceive the tempo a user is exercising in and from this information extrapolates future exercise time events [2]. By making use of the time predictions of this Anticipator, we can specify the trainer's movement to be slightly ahead of them. Note how the availability of a specific Anticipator, and its exact implementation, are application dependent.

[1] This functionality will be shown in our demo and can be tested from the Elckerlyc webstart at `http://hmi.ewi.utwente.nl/showcase/Elckerlyc`.

[2] In our demo we fake these perceptions by using space bar presses instead. For simple fitness exercises one could use, e.g., accelerometers attached to the wrists and ankles of the user, detecting the tempo from the peak structure in the accelerations. Future peak points are then predicted by extrapolating the average tempo of the last few peak points in the exercise performed by the user.


Virtual Human
User
Sensor
movement & speech
Intent Planner
take turn and perform communicative act
(politeness, arousal, friendliness)
Behavior Planner
[wait for turn then speak]
speech with start sync linked to anticipator
Behavior Realizer
predicted stop time
SpeechStop Anticipator

```
<bml id="bml1">
  <speech id="speech1" start="speechStopAnticipator:stop+x">
    <text>Bla bla</text>
  </speech>
</bml>
```

Figure 4: Taking the turn

Figure 3 illustrates this scenario. From interpretation of sensor values (for example: heart rate), the Intent Planner is informed that the current exercise is too easy. It decides to increase the exercise difficulty. The Behavior Planner selects a strategy to achieve this: it decides to gradually increase the tempo of the exercise. This is realized using the strategy described above. This strategy is encoded in the BML block shown in Figure 3. This block describes how synchronization points of a procedural exercise animation (`exercise1`) are synchronized to be slightly ahead (0.5 seconds) of the anticipated synchronization points in the exercise as executed by the user. Each of these synchronization points is linked to a Time Peg. The timing of these Time Pegs is continuously updated using the perceived tempo of the user in the feedback loop on the right, so that the trainer keeps on moving ahead of the user, even when the tempo of the user changes. Of course, if the tempo of the user deviates really too much from the desired tempo, the Intent Planner might still decide on a different exercise strategy, such as choosing a completely different exercise.

3.2 Turn Taking in Speech

3.2.1 Taking the Turn

Humans can take the turn at different moments, for example, slightly before their interaction partner stops speaking, at exactly the moment their interaction partner stops speaking, or slightly after their interaction partner stops speaking. The turn taking strategy used can modulate the impression of politeness, friendliness and arousal of the virtual human (ter Maat and Heylen, 2009). We assume that we can design an anticipator that can predict the end of speech of a user[3], called the `speechStopAnticipator`. Figure 4 illustrates a turn taking scenario. The Intent Planner decides to take the turn and perform a communicative act. The Behavior Planner selects a turn taking strategy, based on the politeness, arousal and friendliness of the virtual human. In the illustrated case, it waits for the user to stop speaking and starts speaking after a certain delay x (could be negative to start speaking slightly before the user stops speaking). To allow an immediate response of the virtual human to the (anticipated) speech stop of a user, the behavior is pre-planned, and its start time is synchronized to this (can be currently unknown) anticipated speech stop. The Behavior Planner thus only specifies that the virtual human starts speaking after the user stops speaking, and the exact and precisely timed execution of this behavior is handled by the Behavior Realizer, using the speech stop anticipator.

3.2.2 Keeping the Turn

To keep the turn, one can simply ignore the interruption request of the interaction partner. Alternatively, one can raise the volume of the voice at the moment of the interrupting speech. The turn keeping strategy used can modulate the impression of friendliness and arousal of the virtual human (ter Maat and Heylen, 2009). Raising the voice requires a real-time change in parameter

[3] We currently fake the detection of speech endings by pressing the space bar.

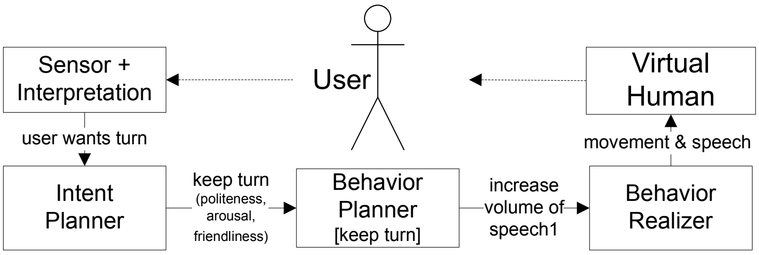


Figure 5: Keeping the turn

```
<bml id="bml2" scheduling="tight-merge"/>
  <bmlt:setparameter id="reparam1" start="10" end="speech1:end"
    target="speech1" parameter="volume" curve="linear" startvalue="25" endvalue="100"/>
</bml>
```

Figure 6: Change the volume of `speech1`, starting at absolute time 10, until `speech1:end`. The volume ranges from 25 to 100.

values (volume in this case) of the virtual human's speech. Elckerlyc currently allows this by providing direct access to the parameter values of each behavior in an animation or speech plan.

Figure 5 illustrates this scenario. The Intent Planner is informed by an interpretation of sensor values that the user would like to get the turn [3]. The Intent Planner decides that the virtual human would like to keep the turn. Based on the provided politeness, arousal and friendliness values, the Behavior Planner decides to realize this in intent by increasing the volume of behavior `speech1`. Currently this is achieved by an ad hoc function call in the Behavior Realizer.

4 Discussion

We have shown that Elckerlyc's Anticipators and Time Pegs provide a flexible formalism for both the specification and the execution of behavior that requires anticipation of the behavior of a (human) interaction partner. They also provide a flexible pre-planning mechanism for behaviors that have to be executed at a (to be determined) later time moment.

We have discussed a scenario in which a parameter value change of running behavior is desired in Section 3.2.2. Currently we apply such parameter value changes in a ad hoc manner. We change the parameter values of a motion or speech fragment by accessing the animation plan or speech plan directly and adapting the parameters of the (possibly running) behavior. We then need to take care of the parameter value curve and the duration of the parameter value change as well. We are currently exploring more formal methods of parameter change specification and execution. An interesting method to achieve this is implemented in the Multimodal Presentation Markup Language (Brügmann et al., 2008): parameter value changes are implemented as an Action (a concept similar to a BML behavior). This allows one to easily define parameter value changes outside the Realizer and to specify the synchronization of the change to other behaviors in a conceptually similar manner as behavior synchronization. Additionally, such a script based specification of parameter value changes allows easy experimentation with parameter values and curves. Figure 6 shows how such a parameter value change could be expressed using BML [4].

However, parameter value change as a BML behavior does not match very well with the other behavior types (gaze, locomotion, speech, etc.) and requires specialized planning mechanisms to be able to refer to BML elements from previously planned BML blocks. Furthermore, we probably do not want parameter values to modify synchronization constraints like other behaviors can do, they simply need to adhere to the timing prescribed by other behaviors. So conceptually it might

[4] Note that this BML block requires a special scheduling mechanism (tight-merge) to allow it to refer to a behavior in a previous BML block, see `http://wiki.mindmakers.org/projects:bml:multipleblockissue`

be nicer to provide a separate (non BML) channel in the Realizer through which a specification of parameter value changes (the timing of which can depend on timing of BML behaviors and that can target a specific BML behavior) can be sent.

Acknowledgements

Many of the ideas and issues discussed here emerged during our very interactive talk and visit of Stefan Kopp's Sociable Agents Group in Bielefeld, we would like thank the participants for a fruitful discussion on continuous interaction. We thank Bart van Straalen for his help in refining the Intent/Behavior planning terminology and Mark ter Maat for explaining his insights on turn taking strategies. This research has been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO) and the Dutch ICT Research and Innovation Authority (ICT Regie).

References

Bevacqua, E., Prepin, E., de Sevin, R., Niewiadomski, R., and Pelachaud, C. (2009). Reactive behaviors in SAIBA architecture. In *Towards a Standars Markup Language for Embodied Dialogue Acts Workshop at Autonomous Agents and Multi-Agent Systems*.

Brügmann, K., Dohrn, H., Prendinger, H., Stamminger, M., and Ishizuka, M. (2008). Phase-based gesture motion parametrization and transitions for conversational agents with MPML3D. In *INtelligent TEchnologies for interactive enterTAINment*, pages 1–6. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., Thórisson, K. R., and Vilhjálmsson, H. H. (2006). Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents*, volume 4133 of *LNCS*, pages 205–217. Springer.

Kopp, S. and Wachsmuth, I. (2004). Synthesizing multimodal utterances for conversational agents. *Comput. Animat. Virtual Worlds*, 15(1):39–52.

Nijholt, A., Reidsma, D., van Welbergen, H., op den Akker, H. J. A., and Ruttkay, Z. M. (2008). Mutually coordinated anticipatory multimodal interaction. In *Nonverbal Features of Human-Human and Human-Machine Interaction*, volume 5042 of *LNCS*, pages 70–89. Springer.

Reidsma, D., Nijholt, A., and Bos, P. (2008). Temporal interaction between an artificial orchestra conductor and human musicians. *Computers in Entertainment*, 6(4):1–22.

ter Maat, M. and Heylen, D. (2009). Turn management or impression management? In *Intelligent Virtual Agents*, volume 5773 of *LNCS*, pages 467–473. Springer Verlag.

Thórisson, K. R. (2002). Natural turn-taking needs no manual: Computational theory and model, from perception to action. In *Multimodality in Language and Speech Systems*, pages 173–207. Kluwer Academic Publishers, Dordrecht, The Netherlands.

van Welbergen, H., Reidsma, D., Ruttkay, Z. M., and Zwiers, J. (2010). Elckerlyc: A BML realizer for continuous, multimodal interaction with a virtual human. *To Appear in Journal on Multimodal User Interfaces*.

List of authors