Available online at www.sciencedirect.com
ScienceDirect
Artificial Intelligence 172 (2008) 1219–1244
Artificial Intelligence
www.elsevier.com/locate/artint

Fully generated scripted dialogue for embodied agents

Kees van Deemter [a,*], Brigitte Krenn [b], Paul Piwek [c], Martin Klesen [d], Marc Schröder [d], Stefan Baumann [e]

[a] *Computing Science Department, University of Aberdeen, Scotland, UK*
[b] *Austrian Research Centre for Artificial Intelligence (OEFAI), University of Vienna, Austria*
[c] *Centre for Research in Computing, The Open University, UK*
[d] *German Research Centre for Artificial Intelligence (DFKI), Saarbruecken, Germany*
[e] *IfL Phonetik, University of Cologne, Germany*

Received 13 March 2007; received in revised form 29 January 2008; accepted 14 February 2008

Available online 29 February 2008

Abstract

This paper presents the NECA approach to the generation of dialogues between Embodied Conversational Agents (ECAs). This approach consist of the automated construction of an abstract script for an entire dialogue (cast in terms of dialogue acts), which is incrementally enhanced by a series of modules and finally "performed" by means of text, speech and body language, by a cast of ECAs. The approach makes it possible to automatically produce a large variety of highly expressive dialogues, some of whose essential properties are under the control of a user. The paper discusses the advantages and disadvantages of NECA's approach to Fully Generated Scripted Dialogue (FGSD), and explains the main techniques used in the two demonstrators that were built. The paper can be read as a survey of issues and techniques in the construction of ECAs, focusing on the generation of behaviour (i.e., focusing on information presentation) rather than on interpretation.

Keywords: Embodied conversational agents; Fully generated scripted dialogue; Multimodal interfaces; Emotion modelling; Affective reasoning; Natural language generation; Speech synthesis; Body language

1. Introduction

A number of scientific disciplines have started, in the last decade or so, to join forces to build Embodied Conversational Agents (ECAs): software agents with a human-like synthetic voice and a computer-animated body, who can engage in a conversation in natural language. Although many techniques in this area are shared between all ECAs, this paper focuses on one particular "family" of ECAs, whose behaviour is determined by an automatically generated *scripted* dialogue, rather than by autonomous agents that make their own decisions. Let us start by explaining what a scripted dialogue is.

* Corresponding author.
E-mail address: k.vdeemter@abdn.ac.uk (K. van Deemter).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2008.02.002

Scripted dialogues follow a master plan. Perhaps the most basic example of scripted dialogue is the stage dialogue, in which actors behave according to a script that was written not by themselves but by a playwright. Two actors playing Romeo and Juliet, for example, do what they do not because they want to, necessarily, but because someone else (Shakespeare, or someone adapting his work) wants them to. The communication between the actors is arguably fake; the 'real' flow of information goes from the script writer to the audience. The same is true for the dialogues between people in a TV commercial, where the real communication is from manufacturer to customer.

This paper describes an approach to the computational production of scripted dialogues that has arisen from the NECA[1] project, and which is henceforth called the NECA approach to scripted dialogue. In the NECA approach, the generation of dialogue behaviour is centralised: the heart of the NECA system is an automated script writing engine. This engine produces a script that can be performed by ECAs. The ECAs are comparable to actors: like their human counterparts, they are carrying out a script that was written by someone else.

ECAs appear to have entered the world of scripted dialogue in a number of systems described in [2]. Initially, scripts were mapped to words and gestures in a fairly direct manner (up to fully canned text). In this paper, however, we show how the approach can be made more powerful when combined with techniques from Natural Language Generation (NLG), which is why we speak of *fully generated* scripted dialogue (FGSD). NLG programs are able to express any well-formed input information in a language such as English or German, for example. NLG makes it possible to express one and the same content in many different ways. This makes it possible to create an endless variety of different actors, each of which acts out any role that is given to them, following a single set of rules that govern his or her manner of speaking and moving. This is especially important—and especially challenging—when different ECAs take on distinct 'personalities', and when their expressive power starts to include the expression of emotion, as is more and more often the case. Henceforth, when we speak of 'expressive' dialogues, we mean multimodal dialogues that are not only able to express factual information, but the affective state of the characters in it as well.

Although research on ECAs is different from work on *computer games*, it is instructive to compare the two endeavours. Games programmers create characters that display sophisticated behaviours and are often able to engage in a dialogue with each other. However, the creation of such games is time consuming and involves a great deal of handcrafting. Even so, the amount of variation displayed by the characters tends to be limited: the number of different dialogues is typically small and these are always performed in the same way, with only minor variations. Games could arguably become more interesting, enjoyable and useful if the characters in them displayed more richly varied behaviour (cf. [23] on ECAs). Taking the notion of a computer game as a point of departure, the goal of most work on ECAs can be viewed as: making it easier and cheaper to create a large variety of appealing and effective dialogues in a controlled way. The Holy Grail of this work—which can be applied to games and more 'serious' applications alike—is to create tools that allow the (semi-)automatic construction of dialogues between believable and highly expressive characters. NECA aims for that Holy Grail. It is for this reason that *variation* of the generated dialogues—at all levels, and involving all modalities—is such a central design constraint for NECA, which motivates many aspects of the approach, including the choice for *fully generated* dialogues.

Generating *scripted* dialogues involves a specific set of tasks, different from the ones involved in the construction of autonomous agents. In scripted dialogue, there is no need to recognise or understand verbal input, for example. The challenge is to *generate* dialogues between agents who behave *as if* they understood each other and reacted to each other in believable ways. "Believable" implies, of course, that the content and form of the dialogues has to be appropriate. ECA systems based on *autonomous* agents [13,43,72,81] interact with real people as well as with ECAs. This comes naturally to them, as it were. ECA systems based on scripted dialogue, by contrast, find interaction with people more difficult, because all possible interactions must be built into the script. However, they also have certain advantages, particularly in terms of the alignment between modalities, and in terms of their ability to ensure that the generated dialogues fulfil constraints on, for example, their total length, their style, and their internal coherence [68].

This paper presents NECA's approach to the creation of varied and expressive dialogues, with respect to all the different levels and modalities, and their synchronisation. Section 2 sketches the two different applications that

[1] 'NECA' stands for Net Environment for Embodied Emotional Conversational Agents, see www.ofai.at/research/nlu/NECA/. We speak of the NECA approach or system to refer to the ideas underlying the two demonstrators developed in the project.

Fig. 1. eShowroom: selection of actor personality.

were explored in order to test the generality of our methods. Section 3 discusses architectural issues. Section 4 describes how the initial dialogue scripts are produced. Section 5 explains how these scripts are subsequently treated by the Multimodal Natural Language Generation module. Sections 6 and 7 focus on speech and gestures respectively.

In the course of the paper, we will explain in some detail how NECA differs from alternatives proposed in the literature, thereby allowing the paper to be read as a review of the state of the art in the construction of ECAs, as well as an introduction into Fully Generated Scripted Dialogue. The wide-ranging character of the paper allows some important issues to emerge, such as the trade-off between quality and flexibility, and the advantages of an incremental system architecture. These issues are highlighted in the Conclusion (Section 8).

2. Two NECA applications

Each of the two NECA demonstrators takes an existing demonstrator as its point of departure: The **eShowroom** demonstrator was inspired by work on collaborating presentation agents [1]; **Socialite** is an extension of the Sysis NetLife platform, a community-building tool where users are represented by avatars [48]. In both cases, we have stuck with the names under which these demonstrators' predecessors were known. Both systems, however, were very substantially enhanced in terms of the generality of their architecture, and in terms of the variety and quality of the dialogues produced.

In the **eShowroom** scenario, a *car sales* dialogue between a seller and a buyer is simulated. The purpose of this application is to entertain the site visitor and to educate him or her about cars. User interaction is restricted: users can set a few parameters which will influence the dialogue (i.e., the content of the script and how it will be 'played' by animated characters). After the user has specified her/his preferences about cars (e.g., saying whether they find road safety particularly important), the personality of the acting characters, and the role (buyer or seller) played by a given agent, a dialogue is generated which takes these settings into account. Fig. 1 shows the interface for selecting the character's personality. For the virtual actor Ritchie the characteristics "good humoured" and "impolite" have been selected by the user. Fig. 2 illustrates the interface for determining the user's preferences on the value dimensions specified for the product. Fig. 3 shows a typical scene from the eShowroom with the two agents (seller and buyer) located in front of a selection of cars, and a screenshot from the Socialite system.[2]

Socialite was designed as part of a multi-user Web community (derSpittelberg.at) where the users create their personal avatar, endow it with personality traits and preferences and send it to the virtual environment in order to meet other avatars. The overall goal of an avatar is to be accepted in the community, and to reach a certain degree of popularity. The community metaphor involves flat-sharing students who live in an area of Vienna named Spittelberg, hence the name of the community: derSpittelberg. Socialite scenes are strongly influenced by the evolving

[2] The screenshot is taken from a demonstrator for an international audience, which is why the text below the animation window is an English translation of the German spoken dialogue. In the online version, the German text is displayed.

Fig. 2. eShowroom: selection of value dimensions.

Fig. 3. eShowroom: typical scene.

Fig. 4. Screenshot: Socialite.

social relations that a user is involved in. When the user is not logged on, she is represented by her avatar in the ongoing (electronic) life of the community. The avatar/agent reports back every time the user logs on. Animated dialogues simulate encounters that the user's avatar has had with other avatars (Fig. 4). To diminish the likelihood of problems stemming from limited speech quality, the text of the dialogue is displayed below the animation. The frame on the left-hand side of the screen depicts the calendar functionalities including an overview of previous encounters.

Dialogues in eShowroom are based on a straightforward model of the world of cars and customers, with a focus on conveying information that is correct and relevant to the customer. Socialite, by contrast, had to accommodate a more colloquial conversational style, emphasising the personality and social background of the speaker. It was an important challenge for the project to tackle both kinds of dialogues using essentially one and the same approach to Scripted Dialogue. The fact that eShowroom (English) and Socialite (German) used two different languages was an added complication.

Evaluation. Several specific aspects of the NECA approach are evaluated in later sections, using whatever methods seemed most suitable for the technology under discussion. Even though *system-level* evaluation is not the focus of this paper, it is worth summarising the main findings from a pair of field studies that were done with the two demonstrators [33]. Beta versions of each of the two were made available to the general public for three months, accompanied by only a minimum of advertisement. In the case of Socialite, this led to 1488 logins by 66 different users, showing an encouraging return behaviour. Approximately half of the 66 participants visited their avatars at least 5 times, while 20 of them did this at least 50 times. In eShowroom, where there is no user registration and each animated presentation is self contained, we logged all those 241 presentations played during the evaluation period.

Each user, of each of the two systems, was asked to complete a questionnaire assessing her impression of the animated dialogues. In the most crucial questions, subjects were asked to express their agreement or disagreement on a five-point scale. As usual in questionnaires associated with field studies, only a fraction of participants completed their questionnaires, resulting in 17 completed questionnaires from Socialite and 11 from eShowroom. In both cases, a clear majority of subjects classified themselves as having considerable expertise in information technology. (As many as 64% of eShowroom users and 88% of Socialite users characterised themselves as using animated characters on a regular basis.) The results indicate that both demonstrators were seen as quite enjoyable. In Socialite, for example, 47% of subjects found the application enjoyable (ticking a 4 or a 5 on the agreement scale following the statement "I found the dialogue enjoyable"), 24% gave a negative opinion (a 1 or 2 on the scale), while 29% were neutral (the mid-point 3 on the scale). Participants in the questionnaire judged body movements and facial expressions to match the spoken words very well (Socialite: 48% positive, 40% neutral, 12% negative), but the quality of the speech was rated much lower (Socialite: 82% negative, 12% neutral, 5% positive; see Section 6 for discussion). In eShowroom, where this issue is of particular importance, the two characters in the dialogue were judged as matching the parameters that the user had set for them very well (cf. Figs. 1 and 2). A puzzling finding is that female participants were much more critical of just about all aspects of both demonstrators. (Similar findings were reported in [21].) All these figures need to be taken with a grain of salt given paucity of respondents (both in absolute terms and as a percentage of users), whose familiarity with animated characters was, moreover, unusual.

3. Architecture and representation language for scripted multimodal dialogue

Each of the different modalities (text, speech and body language) that are employed in a dialogue involve expressive choices, for example, concerning the words, gestures and intonation patterns that are used. All these choices must be properly synchronised. For example, if a particular concept is new or important, a pitch accent must appear on the words that convey this concept; additionally, the mouth and eyebrows should move at the right moment. In order to meet these challenges, NECA uses a specially designed architecture, representation language, and processing model. These key aspects of the NECA approach will be introduced in this chapter. We start by focusing on the architecture and the processing model before discussing the representation language.

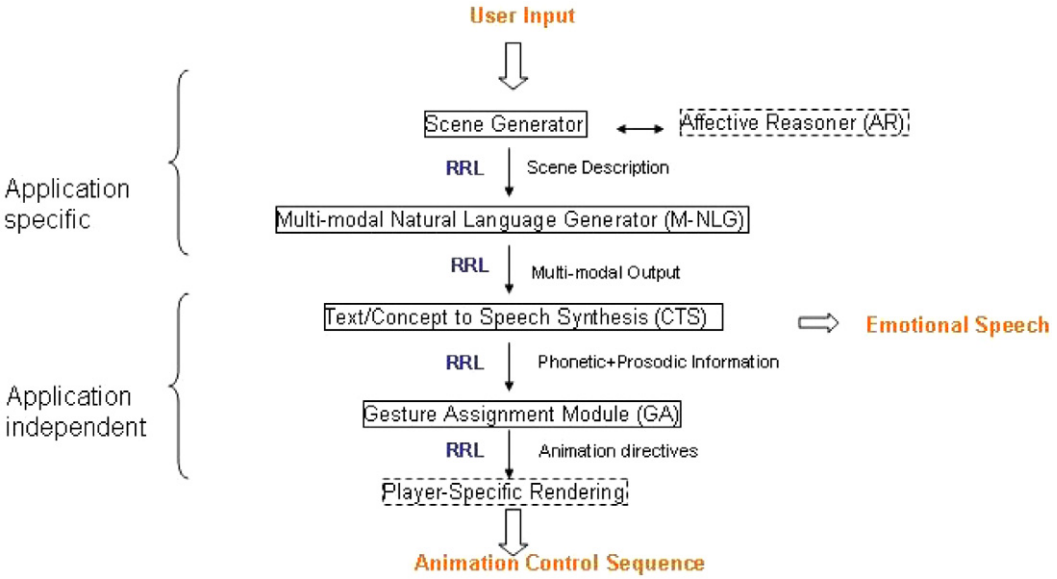

Fig. 5. Architecture realising the scripted multimodal dialogue component.

3.1. An architecture for generating multimodal dialogue

Fig. 5 offers an overview of the NECA architecture. The *Scene Generator*, which uses an "Affective Reasoner" (also called Emotion Engine) to produce a Scene Description, takes the role of a playwright, planning the dialogue and generating a script. In the Scene Description, dialogue and presentation acts are specified as well as their rough temporal coordination. The dialogue is not generated from left to right (e.g., one turn at a time), as in a conventional interactive system, but from the top down. The Scene Description specifies the semantic content, type, temporal order, and associated emotion of the communicative acts that the characters will perform. All this information is encoded in an XML document which is incrementally refined in the course of processing. Since the Scene Generator constructs (outlines of) dialogues, this module is specific to Scripted Dialogue. All later modules, however, use techniques that could equally be used to produce dialogues between *autonomous* agents, once each of these agents' behaviour is specified in the right format.

The Scene Description is then handed over to the Multimodal Natural Language Generator (Section 5), which transforms the formal specification of the communicative acts into text. This component is also partially responsible for the selection of gestures. The Multimodal Output is an XML-based script specifying a set of sentences and gestures with their temporal ordering. The task of Speech Synthesis (Section 6) is then to convey, through adequate speech, the intended meaning of the text as well as the emotion with which it is uttered.[3] It also provides information on the exact timing of utterances, syllables and phonemes, which is indispensable for the Gesture Assignment Module (Section 7). The latter module is responsible for exact timing of gestures relative to speech. Its output is a script of "animation directives", that is, a control sequence comprising the synchronised verbal and non-verbal behaviour of all the characters in the scene. In a last step this control sequence is converted into a data stream that is processed by an animation player. While scene generation, dialogue planning and textual surface realisation are largely application specific (though important parts of their mechanisms can be reused), later components are almost entirely domain independent.

The key feature of the NECA processing model is its incrementality: each module (up to the Rendering module) adds information to the script, without ever throwing information away. This allows a module to use the information added at any previous stage, without compromising the pipeline. (See also Conclusion section, under "Architecture and Processing Model".) The following section explains how this incremental process works.

[3] This module is also called Text/Concept-to-Speech synthesis, because its input can be text but also more abstract conceptual structures.

3.2. The rich representation language (RRL)

The modules of Fig. 5 presuppose a representation language that is expressive enough to represent all the information that these modules produce (except the Player-Specific Rendering at the end of the pipeline), and all the information that they consume (except the Scene Generator, at the start). A variety of structures, usually XML compliant, have been designed to allow for the specification of multimodal information, but we were unable to find any that was expressive enough to represent everything from (Discourse Representation Theory-based, see [40]) semantic information to words, speech and body language. To put NECA's representation language in context, we compare it briefly with other languages that are associated with ECAs.

Mark-up languages typically define sets of mark-ups that allow a non-expert user (e.g. a web designer) to annotate a text with high-level expert information. See for instance VoiceXML (http://www.voicexml.org) for creating voice enabled web applications, or VHML (http://www.vhml.org, Virtual Human Markup Language) for creating interactive multimodal applications with Talking Heads or full bodied ECAs. Other examples of markup languages where text is annotated with high-level concepts are APML [22], MPML [78]. Languages of this kind were not built to represent detailed syntactic, semantic and pragmatic information.

Representation languages (in our sense) are unlike mark-up languages, because they have a system-internal, rather than a Human-Computer Interfacing function. Existing languages of this kind have a very limited function, in the interface between two system components [15,44,46,86]. Our own representation language had to be more general, extending and combining different aspects of existing representation languages, which is why we designed the so-called Rich Representation Language (RRL). RRL [63] combines information at all levels: the semantic level (where the content of the utterance is specified), but equally the textual string of words that make up an utterance, and also information about speech and body language.[4]

NECA's RRL[5] is used for specifying a multimodal dialogue at its various stages of generation, as more and more detail is added to the dialogue script. At the end of the pipeline, the RRL script contains sufficient information to be mapped to a chain of low-level animation directives. We start by describing the structure of the abstract script of the dialogue (i.e., the Scene Description), which contains (1) a representation of the initial common ground, (2) a representation of the participants of the dialogue, (3) a representation of the dialogue acts, and (4) a temporal ordering of the dialogue acts. This is the information available after Affective Scene Generation.

In the following we will explain the elements of an RRL script in more detail. A full specification of the RRL XML Schema can be found at www.ofai.at/research/nlu/NECA/RRL/index.html.

1. *Common Ground*. The initial common ground captures the information shared by the interlocutors at the start of their conversation. It identifies the referents and specifies their properties in terms of n-ary predicates. The information in the common ground is used by the MNLG module for the generation of referring expressions. All semantic information of the dialogue is formalised making use of Discourse Representation Theory [33].
2. *Participants*. Each dialogue participant is provided with person data such as name and gender, appearance (= graphics design) and voice (e.g. pitch range). Each character is also equipped with information on its personality and its role in the scenario. In the eShowroom scenario, for instance, the roles of the interlocutors are seller and buyer.
3. *Dialogue Acts*. A dialogue is represented by means of individual acts which can be verbal or non-verbal. Each dialogue act is represented as an xml element with a number of subelements including a characterisation of the act's communicative function (encoded in <domainSpecificAttr/>, cf. Fig. 7), its semantic content (as a Discourse Representation Structure [33]), and the prevalent emotion expressed (cf. Fig. 7, <emotionExpressed/>), as computed by the Affective Reasoner.
4. *Temporal Ordering of Dialogue Acts*. The temporal ordering of the individual acts of a dialogue is specified via a <temporalOrdering/> element. Usually, verbal dialogue acts follow a sequence of speaker contributions. Non-verbal acts such as backchannelling typically occur in parallel with dialogue acts of the speaker. Ac-

[4] Languages such as XSTEP [37], and ABL [55]. incorporate both declarative and procedural knowledge. They function more as programming languages for behaviour generation than as behaviour markup or representation.

[5] A full specification of the RRL XML Schema can be found on www.ofai.at/research/nlu/NECA/RRL/index.html.

cordingly <temporalOrdering/> has two subelements <seq/> and <par/> which take dialogue acts as their subelements.

To generate text interleaved with gestural information, Multimodal Natural Language Generation (Section 5) processes the communicative function, the emotion and the semantic content, adding <sentence> and <gesture> elements to the dialogue act. (See example below.) In eShowroom, <gesture> is a small animated clip (using 3D Charamel animation) that combines hand-arm gesture, posture and facial expression. In Socialite, facial expression and hand-arm gesture are encoded in separate <gesture> elements, using 2D Flash animations.

Specification of <dialogueAct> after MNLG: Example

```
<dialogueAct>
  <gesture
    meaning="takingcommand"
    modality="body"
    identifier="hips"
    id="g1" alignto="s1"
    aligntype="seq_before"/>
  <sentence id="s1">
    How much fuel does it
    consume?
  </sentence>
</dialogueAct>
```

This example comprises a gesture g1 and a sentence s1. Their relative alignment is encoded by means of alignto and aligntype.

The elements are aligned to each other sequentially, e.g., g1 $<_{\text{seq\_before}}$ s1 (i.e., the speaker first makes a posture shift g1, then utters sentence s1. (In other cases, a gesture may be aligned with the start of a sentence, causing the two to overlap.)

The information encoded in <sentence> is sent to Speech Synthesis. Synthesis produces a sound file, and an RRL script in which <sentence> encodes the address of the sound file, the SAMPA-encoded (www.phon.ucl.ac.uk/home/sampa/home.htm) phonetic transcription of the text including syllable structure, and TOBI-encoded accentuation and prosodic boundaries [6]. See the example below.

Specification of <sentence> after Speech Synthesis: Example

```
<sentence id="s001"
  src="s001.mp3">
  Hello
  <word id="w_1"
    accent="H*" pos="UH"
    sampa="h@l-'@U">
    <syllable id="syl_1"
      sampa="h@l">
      <ph dur="75" p="h"/>
      <ph dur="48" p="@"/>
      <ph dur="100" p="l"/>
    </syllable>
  </word>
  <prosBoundary
    breakindex="4"
    dur="200" p="_"
    tone="H-L%"/>
</sentence>
```

The sentence s001 comprises a single two- syllabic word, ”Hello” (h@l-@U) with a rising-falling prosodic contour (H-L%), followed by a 200- millisecond pause (i.e., prosodic boundary).

For brevity, the information relating to the second syllable of the word is omitted.

Phonetic, prosodic and timing information are encoded by means of the <word> element and its sub-elements <syllable> and <ph> (phoneme). This representational structure is necessary for fine-grained temporal integration of speech and gesture.

The output of the Gesture Assignment Module is an RRL specification of the animation stream, using a subset of SMIL (Synchronised Multimedia Integration Language http://www.w3.org/TR/smil20/). All linguistic information in <dialogueAct> is replaced by an audio element which holds the name and duration of the speech sound file. The alignment between gestures and language-related entities (e.g. sentences, words, syllables) is made precise. The result

is encoded in <animationSpec/> which is then input to style sheets that transform the RRL representation into a player-specific one.

Specification of <animationSpec> after Gesture Assignment: Example

```
<animationSpec>
  <par>
    <gesture id="g001"
      key="ge20"
      dur="1650"/>
    <gesture id="g002"
      key="ge03"
      begin="259" dur="1200"/>
    <par>
      <audio src="s001.mp3"
        dur="653"/>
      <seq>
        <gesture
          modality="viseme"
         identifier="v_h"
         dur="75"/>
        <gesture/> … <gesture/>
      </seq>
    </par>
  </par>
</animationSpec>
```

The <animationSpec> comprises two gestures (g001, g002) and an audio file (s001.mp3) which are played in parallel. As only duration information is specified for g001 and the sound file, both start at the same time t0, whereas g002 starts at an offset of 259 milliseconds.

In parallel to the audio file the sequence of related visemes and their durations is specified. In the example we omit all except the first viseme, which is the mouth movement v_h associated with the first phoneme of the word "Hello".

4. Affective scene generation

We aim to produce a large variety of believable dialogues. Each dialogue should match the personality of its participants (as specified by the user, see Fig. 1). Moreover, the dialogues should match the interests of the user, as reflected by their choice of value dimensions (see Fig. 2), and the characters have to display the types of emotions that fit the situation. The module that produces "skeletal" dialogues should therefore take all these factors into account.

What follows is a description of the plan-based approach to affective scene generation, employed in NECA's eShowroom scenario.[6] The approach is an extension of previous work on the generation of dialogue scenes for animated presentation teams [2] and on integrating models of personality and emotions into lifelike characters [3]. In NECA we combine the dialogue act generation for the car sales domain with our mechanisms for emotion elicitation and computation. The result is a sequence of dialogue acts that do not only specify the semantic content of the utterance but also the affective state of the speaker.

4.1. Domain modelling

Domain modelling is an essential prerequisite for automatic dialogue generation. In the eShowroom scenario the domain model consists of two parts. The first part is a factual description of the different cars that comprises the kind of information one typically finds in a car sales brochure. In our model, each car is characterised by the following attributes: price, horsepower, maximum speed, fuel consumption, spaciousness of interior, spaciousness of luggage compartment, proportion of recyclable materials used in the manufacturing, and the availability of optional features (e.g., anti-lock brakes, airbags, broad tires, power windows, leather seats, and a catalytic converter). This information is stored in a knowledge base and accessed by the dialogue planner both to inform the selection of dialogue strategies and to specify the propositional content of the individual dialogue acts as explained in the next section.

The second part of the domain model relates the attributes to the set of value dimensions that users can select to express their preferences: operational costs, safety, sportiness, comfort, prestige, family- and environmental friendli-

[6] In Socialite, emotions are not computed at runtime, but essentially a hard-wired part of the templates used by its MNLG module (cf. Section 5).

ness (see Fig. 2). The dimensions were adopted from a study of the German car market because they are particularly relevant for people purchasing a car. The domain model characterises an attribute in two ways. First, how relevant it is for a certain value dimension: low, medium or high. For example, the "(fuel) consumption" attribute's relevance for the value dimension "operational costs" is high. Second, the valence of an attribute's value which is determined by an evaluation function: positive or negative. For example, a consumption of 10 litters per 100 kilometres is rated negative in the "operational costs" dimension. The same value (e.g. "230 HP") can sometimes be rated positive in one dimension (e.g. "sportiness") and negative in another one (e.g. "safety").

4.2. Dialogue act generation

The domain model determines to a large extent what the virtual characters can talk about, since nearly all questions and answers in the car sales dialogues refer to the cars' attributes. However, such a factual description does not say anything about how this information is to be presented. This knowledge is contained in the dialogue model that specifies both the global and the local structure of the conversation in terms of dialogue strategies. Our sales dialogues start with a greeting phase, followed by the customer's request for information about a specific car. Subsequently, a question–answer game between the customer and the sales person develops in which the features of the cars are discussed. Finally, the customer communicates a purchase decision and, in a closing phase, the dialogue ends.

In the eShowroom scenario, the dialogue planner generates the initial version of a Scripted Dialogue as a sequence of dialogue acts. A dialogue act represents an abstract communicative function, such as requesting information (e.g. requestIf), answering a question in the affirmative, or giving feedback (e.g. agreeing). Such communicative functions can be realised in many different ways depending, for example, on the personality of the actor. Dialogue acts usually follow each other in a typical order. For example, a question about the availability of some feature might be followed by a positive or negative answer, which is then further discussed by the dialogue participants. In the dialogue model such combinations of dialogue acts that are frequently observed in the genre at hand are represented as dialogue strategies. Following our plan-based approach, dialogue strategies are encoded as plans that can be selected and executed by the dialogue planner. Fig. 6 is an example of a plan for the dialogue strategy "QuestionAnswer:Boolean" introduced in the previous example. The customer requests information about a Boolean attribute, i.e. an attribute that the car either has or does not have (e.g., airbags). The dialogue planner retrieves this information from the domain model, and depending on the attribute's value, the sales person will confirm or disconfirm the availability. Finally, a new dialogue strategy is triggered in which both actors discuss this new piece of information.

Plans are referenced by their name. Their applicability in a given dialogue context is defined through a goal expression and a precondition. Both sections can contain instantiated and uninstantiated variables (in the example denoted as strings preceded by a dollar sign). Uninstantiated variables get their bindings when plans are selected and instantiated. The precondition specifies the initial conditions that must be fulfilled before a plan is scheduled for execution. As shown in Fig. 6 this typically requires that some facts can be established by retrieving them from the dialogue planner's knowledge base. Goal expressions are matched against the set of goals currently pursued by the dialogue planner. To increase the variation of dialogues, multiple plans with the same goal expression (and optionally with different preconditions) can be specified. To inform the selection of dialogue strategies, the utility of these plans (reflecting their goodness of fit in a particular situation) can be specified by an integer value.

The dialogue planner constantly checks which plans are applicable by matching the goal expressions of all specified plans against its current set of goals. Plans that match and whose preconditions are fulfilled are added to the set of applicable plans. The dialogue planner then chooses the plan with the highest utility value. If the choice is still ambiguous, i.e. if there are at least two applicable plans with the same utility value, one of them is randomly chosen and executed. By providing multiple plans with the same utility for a given situation non-determinism can be introduced in the dialogue act generation process, so that different dialogue act sequences are generated each time the dialogue planner is invoked. During plan execution the actions in the body section of a plan are executed. The plan body is a procedural specification that defines how a plan's goal can be achieved, typically by spawning some subgoals that will trigger new dialogue strategies. In this way, a plan tree is incrementally built by the dialogue planner in which the nodes represent dialogue strategies and the leaves represent the individual dialogue acts to be performed by the interlocutors. Plans may be interrupted and suspended at any time if a new plan with a higher utility becomes applicable. This mechanism is used, for example, to adapt the dialogue generation process to the affective state of the virtual characters as explained in the next section.

```
Plan {
Name: "QuestionAnswer:Boolean"
Goal: PERFORM QuestionAnswer $car $attribute;
Precondition:
  FACT type $attribute "Boolean";
  FACT role "customer" $actor1;
  FACT role "salesperson" $actor2;
Body:
  PERFORM DialogueMove $actor1 "requestIf" $attribute;
  ASSIGN $value (getValue $car $attribute);
  IF ($value == "true") THEN {
    PERFORM DialogueMove $actor2 "confirm";
  } ELSE {
    PERFORM DialogueMove $actor2 "disconfirm";
  }
  PERFORM DiscussValue $car $attribute;
Utility: 0
}
```

Fig. 6. Plan of a dialogue strategy for requesting information.

```
<dialogueAct id="v_9" reactionTo="v_8">
   <domainSpecificAttr type="positiveResponse"/>
   <speaker id="tina"/>
   <addressee id="ritchie"/>
   <emotion>
      <emotionExpressed person="tina" type="relief" intensity="0.7"
        activation="0.021" evaluation="0.231" power="-0.021" />
   </emotion>
</dialogueAct>
```

Fig. 7. RRL representation of a dialogue act structure.

A single dialogue contribution is encapsulated in a DialogueMove plan. The plan creates an abstract *dialogue act structure* containing information about the speaker, the speaker's dominant emotion, the dialogue act type, the propositional content if needed, and the temporal alignment with previously generated dialogue acts. Fig. 7 shows the RRL representation for such a dialogue act structure.

As described in Section 2, users can assign roles and personalities to the actors, and select the value dimensions that interest them. These parameters are used in the precondition of the plans and influence the course and style of the ensuing conversation by constraining the selection of the available dialogue strategies. For example, depending on their mood, the two actors display different degrees of criticism or enthusiasm when discussing the car's properties.

4.3. Affect computation

Affect computation in the eShowroom scenario is performed by the Affective Reasoner/Emotion Engine, based on the cognitive model of emotions developed by Ortony, Clore, and Collins [61]. The OCC model defines emotions as positive or negative reactions to events, actions, and objects. Events are evaluated in terms of their desirability, actions in terms of their praiseworthiness, and objects in terms of their appeal. The subjective appraisal of the current situation is based on an agent's goals, standards, and attitudes. The result of the appraisal process is a set of *Emotion Eliciting Conditions* (EECs) which describe, for example, the degree to which an event is desirable and the likelihood of a future event. The Emotion Engine maps EECs to emotion categories and their intensity. An event that is undesirable for someone who is disliked by the agent, for example, triggers the emotion category "gloating" whereas the same event would have elicited "pity" if the other person was liked. The intensity of the generated emotions depends on the EEC variables (e.g. the degree of blameworthiness) and on the personality traits specified for each agent. A decay function models the fact that emotions diminish over time [29]. Although sometimes criticised for its limitations as

```
Plan {
Name: "BreakOffDiscussion"
Goal: PERFORM BreakOffDiscussion;
Precondition:
  FACT role "customer" $actor1;
  FACT emotion $actor1 $type $intensity;
  (AND (== $type "distress") (> $intensity 0.7));
  FACT ObjectInFocus $car;
  FACT role "salesperson" $actor2;
  FACT LastDialogueAct "finished" "true";
Body:
  REMOVE PERFORM Discuss $car;
  PERFORM DialogueMove $actor1 "initiateClosingNegative";
  PERFORM DialogueMove $actor2 "completeClosingNegative";
Utility: 100
}
```

Fig. 8. Plan of a dialogue strategy to break off the discussion.

a psychological theory, the OCC model has, for the time being, established itself as a reference model for emotion synthesis, at least for cognitively modelled embodied agents.

For generating affective dialogues we combine the dialogue generation process described in the previous section with our mechanism for emotion elicitation and computation. This is done via the concept of a *Current Discourse State* (CDS) and a set of *appraisal rules*. The CDS includes the previously-generated sequence of dialogue acts, the object in focus (e.g., a particular car), and the current goals, standards, and attitudes of the agents. When a new dialogue act is generated, the appraisal rules are applied to the CDS. For example, suppose the sales person cannot answer a customer's question. This is appraised by the sales person as an "undesirable event" since it endangers his/her goal to come across as competent. The degree to which this is undesirable depends on how relevant this information is for the value dimensions representing the customer's interest. The generated EEC *very undesirable* is then mapped to the emotion category *distress* with a certain intensity. The customer can appraise the action as blameworthy if she believes that the sales person is hiding unfavourable information. This time the EEC *somewhat blameworthy* is mapped to the emotion category *reproach*. The inferred emotions are used for updating each character's affective state. In the end, the emotion with the highest intensity is assigned to the dialogue act representation.

When the dialogue planner determines the next dialogue move, it takes the new affective states into account by evaluating the preconditions of the dialogue strategies and by selecting the one that best matches the affective states of the characters. For instance, if the sales person repeatedly says "I don't know", the customer will get more and more frustrated. If the intensity of the elicited *distress* emotion exceeds a certain threshold, the question-answer game is interrupted and the closing phase is initiated. The plan for this dialogue strategy which is shown in Fig. 8 has a higher utility value than the currently executing plan for the goal "PERFORM Discuss $car" which will therefore be interrupted and suspended by the dialogue planner. In order to avoid an interruption in the middle of the dialogue act generation process (which could result in a corrupted dialogue act structure) an additional check has been included at the end of the precondition to make sure that the last dialogue act has been finished. The first action in the body of the "BreakOffDiscussion" plan removes the suspended goal from the set of goals pursued by the dialogue planner and in the next actions the subsequent dialogue moves of the customer and the salesperson are performed.

Emotions do not only affect the sequence of dialogue acts generated by the dialogue planner, but also the way in which these are processed by subsequent modules. In particular, the speaker's most dominant emotion will be used as an additional parameter for text generation, gesture alignment, and speech synthesis. For the latter, however, the emotions generated by the Emotion Engine will be mapped to another model of emotion that is thought to be better suited for speech (see Section 6.2).

The Emotion Engine used for affect computation in the eShowroom scenario forms the basis for Gebhard's "A Layered Model of Affect" (ALMA, [30]). This model integrates emotions, moods and personality, covering short, medium, and long-term affect respectively. The plausibility of the generated emotions and moods was demonstrated in an empirical evaluation involving textual dialogues between two or more characters. Subjects were asked to assess the plausibility of the computer-generated emotions and moods for each character, based on these dialogues. The results

indicated that ALMA provides authentic believable emotions and moods [31]. Since NECA uses basically the same functions as ALMA for computing emotion types and intensities, these results can also be interpreted as support for the principles behind NECA's Emotion Engine.

5. Multimodal natural language generation

The aim of the Multimodal Natural Language (MNLG) module is to express the Scene Description (see Fig. 5) in natural language and gestures appropriate to the situation. This implies, in particular, that the emotion and personality of the speaker, as well as the factual information in the dialogue act, are taken into account. Here we sketch the design philosophy behind the MLNG module. For details, see [64] and [67].

Since the MNLG module differs from most existing NLG systems, its task and architecture are worth examining in some detail. MNLG sits between the Scene Generator and Text/Concept to Speech Synthesis. The Scene Generator provides the MNLG with a specification of the content (semantic, pragmatic and emotional) and the structure of a dialogue. The MNLG maps this specification to a representation of the verbal and non-verbal behaviours that constitute a fully-fledged dialogue. The result, a multimodal output representation, describes the combination of words, grammatical constructions and gestures that make up the dialogue. (Phonetic and prosodic realisation, and detailed timing, are left to subsequent modules.) The output of the MNLG is not intended for human consumption; instead, it consists of a machine-readable description of a dialogue which a team of animated agents is expected to act out.

Before delving into the details of the MNLG, let us briefly highlight in what respects it differs from other natural language generators.

The generator described in [60] resembles the MNLG's approach to semantics. Both generators can operate on unordered sets of statements, rather than the highly structured inputs that are required for many off-the-shelf surface realisers such as fuf/surge [27]. The generator in [60] is, however, unimodal and unable to cope with pragmatic constraints, for example regarding the personality and emotions of the speaker.

The MNLG's functionality resembles that of the *microplanner* of an NLG system [71]. Most microplanners, however, have been designed for sentence generation rather than *multimodal* dialogue act generation [11,56]. Some, like the SPUD generator ([79]) can be adapted to multimodal generation [15]. But, like most systems specifically designed for ECAs (e.g., [41,53,70]), SPUD uses an algorithm based on integrated planning, whereas we advocate a highly modular system (see Fig. 9), in order to support fast generation. Integrated approaches have been motivated by psycholinguistic plausibility [42]. We make no psycholinguistic claims for the approach advocated here, but would like to point out that some of the most widely cited psycholinguistic models of speaking are modular, and essentially pipelined [52].

5.1. Requirements for the MNLG

NECA's MNLG module was built with the following requirements in mind:

(1) *Integration of heterogeneous generation resources.* One of the main determinants of a dialogue act is its semantic content. The semantic content that the Scene Generator can provide for a dialogue act depends on the content of the underlying database. For some dialogue acts, such as greetings ('Hello, my name is Ritchie'), it seems impossible to generate from first principles, starting with the semantic content. MNLG therefore needs the capability to combine full generation with templates created by human writers.
(2) *Integration of different factors* (*emotion, personality, etc.*). The realisation of a dialogue script depends on more than just semantic content. To obtain believable presentations, factors such as the personality of the speaker, their gender and their emotional state should play a role. Therefore, we require that the MNLG take a variety of such factors into account when choosing how to put a given message into words.
(3) *Variation of expression*. People say different things on different occasions, even if the circumstances regarding the aforementioned factors are more or less identical. This means that the MNLG needs to be capable of non-deterministic generation.
(4) *Performance*. Long delays would decrease the appreciation of end-users. For this reason, the MNLG should be able to produce output almost instantaneously.

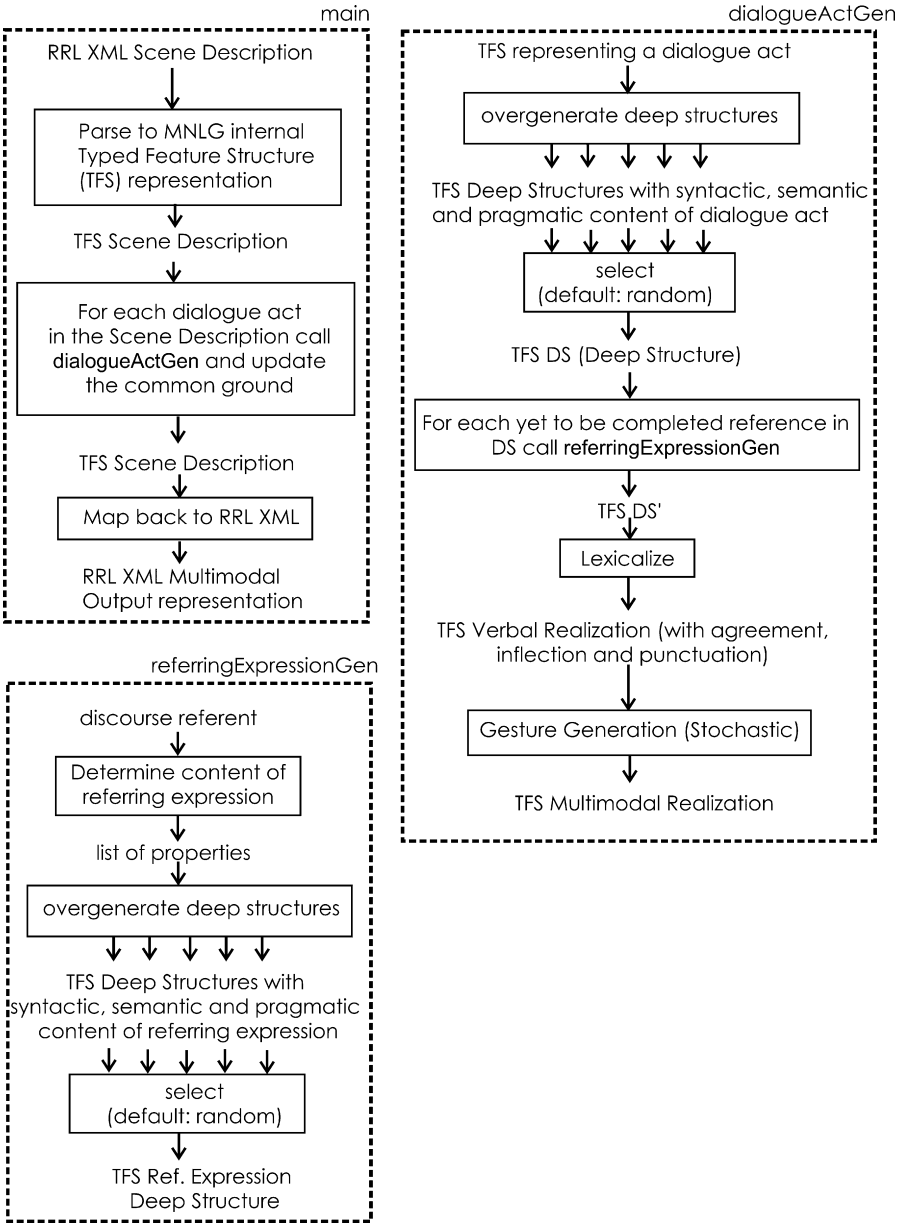

Fig. 9. Schematic representation of MNLG architecture.

(5) *Re-use*. The MNLG is intended to be application independent. It should be easy to port to new applications, thus saving the developers of such applications time and effort.

The next section will explain how these requirements were met.

5.2. Outline of the MNLG module

Like the overall NECA system, the MNLG module has a pipelined architecture (Fig. 9). The module *dialogue-ActGen* generates individual dialogue acts. It, in turn, calls *referringExpressionGen* for the referring expressions that need to be incorporated into the realisation of a dialogue act.

Requirement 3, regarding variation in the output, is addressed by having a number of non-deterministic steps in the generation process: deep structure generation, for dialogue acts and the referring expressions they contain, consist of over-generation followed by (random) selection. Gesture selection also operates through random selection of a gesture from a set of appropriate alternatives.

In order to facilitate maintenance and re-use, the MNLG is divided into (Sicstus Prolog) modules (Requirement 5). Application-specific data are separated from generic generation algorithms so that development of new applications only requires modification of the data files. The highly modular setup in combination with a pipeline architecture also contributes to the high performance (in terms of generation times) of the system; see the next section on evaluation (Requirement 4).

One of the main tasks of the MNLG is the generation of "deep structures" for dialogue acts (i.e., pairings of content with verbal and non-verbal realisations) which satisfy a given set of syntactic, semantic and pragmatic constraints. These constraints constitute the input to the MNLG and are dictated by the Scene Generator. Formally, the collection of input constraints is represented by a typed feature structure. The typing of the structures facilitates reuse and maintenance of the system (since an explicit representation of the data structures is kept separately). The structures are manipulated using Prolog's fast built-in unification algorithm (through the Profit library in [28]). The linguistic resources are represented as trees whose nodes are also typed feature structures. Together, these trees make up the *MNLG's tree repository*.

Generation consists of matching the input feature structure with the root nodes of the trees in the repository. Matching trees may have incomplete daughter nodes (i.e., daughters that are not yet fully realised). These are recursively expanded by being matched with the trees in the repository, until all daughters are complete. Daughter nodes whose semantics give rise to referring expressions are dealt with by the *referringExpressionsGen* module (see Fig. 9 and [67]).

The formalism for the trees in the repository is able to represent linguistic resources of a wide variety, including lexical entries, spans of canned text, templates and full-fledged grammar rules. For a given input, the resources often allow the construction of *multiple* deep structure trees, one of which is selected at random. Emotion and personality are stored in the attribute currentAct, and they influence selection of words, phrases and gestures from sets of alternatives that express the same semantic content (Requirement 2). Fig. 10 shows an example of a tree representing a template. The usual angled-brackets notation for feature structures is used; types are in italics and attributes in small capitals. Sharing of values is represented by co-indexing.

When this template is called, the value of "Speaker" is unified with the name of the speaker of the current dialogue act, which ends up in the realisation of the template. Note that the template provides a full syntactic structure for any sentence generated with it, blurring the distinction between real and template-based generation in line with current thinking in Natural Language Generation [83]. Note that the meaning of the sentence is not 'computed' composition-

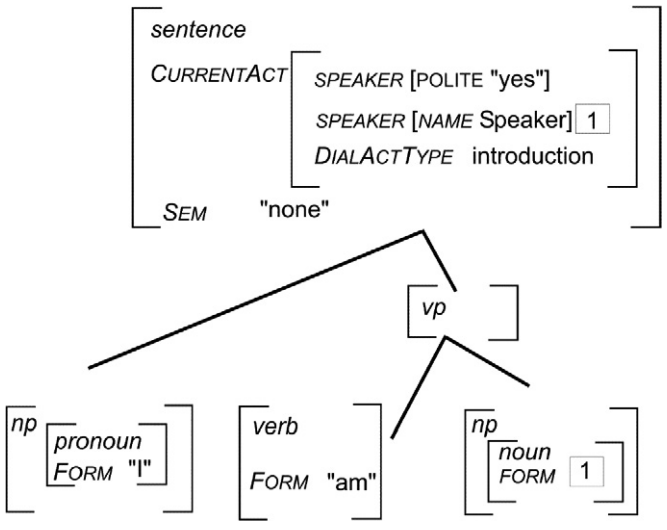

Fig. 10. Template for "I am NP".

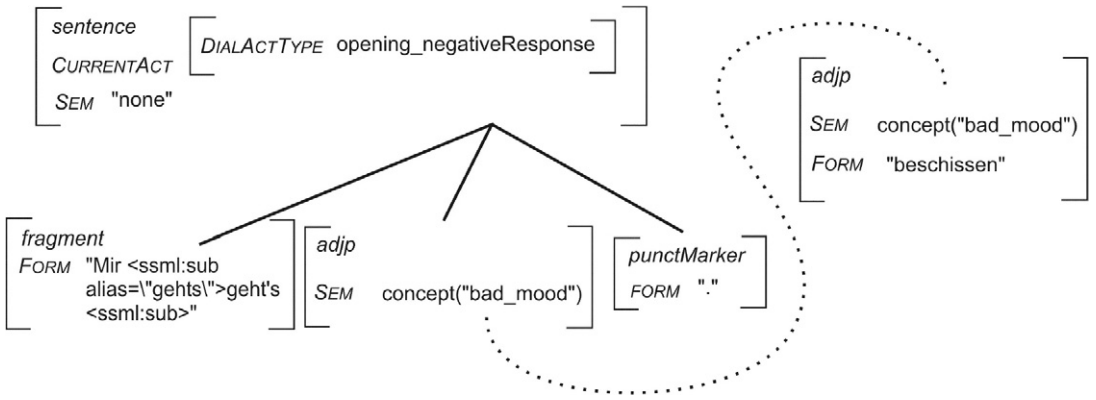

Fig. 11. Socialite templates for "Mir geht's" ("I feel...") and "beschissen" ("all fucked up").

ally from the meaning of its parts. Grammar rules with a compositional semantics are only useful where the input to the generator consists of complex semantic representations in the first place. For example, in eShowroom this holds for the description of cars; for these, the underlying database allows us to derive complex semantic input representations.

The example in Fig. 11 derives from the Socialite application, as one might guess from its colourful use of language. It provides an example of a linguistically fully specified template (for adjp). This template combines (as indicated by the dotted line) with a sentence template whose semantics is radically underspecified (value is "none") and which has a linguistically underspecified constituent (the node labelled "fragment").

5.3. Evaluation of the MNLG module

The MNLG goes a long way towards meeting the requirements introduced earlier in this section. We have seen that requirements (1) to (4) have been addressed through specific design decisions. Requirement (5), involving system performance, was evaluated by running tests to measure average generation times on a range of examples [64]. These tests provided satisfactory results, with generation taking between $1/100$ and $4/100$ of a second per dialogue act (based on a tree repository consisting of 138 generation trees, and using a Pentium III 1200 MHz processor). Requirement (6), on re-use of the MNLG, was evaluated by porting the MNLG: it was re-used in the Socialite demonstrator, then once again, outside the NECA project, in the epoch iGuide Virtual Tour Guide system [26].

Most Socialite generation templates (e.g., Fig. 11) were originally written by professional script writers, in a format different from that used by the MNLG. The tree formalism proved to be flexible enough to accommodate these pre-existing templates: Perl scripts were written for automatically transforming these into MNLG trees, resulting in a tree base of 1170 trees. The experience of implementing the epoch iGuide Virtual Tour Guide system's generation component using the MNLG was similarly encouraging.

We investigated the effect of different MNLG settings, focusing on NECA's eShowroom demonstrator, comparing dialogues with and without gestures [65]. Neither subjective user experience (as measured through a questionnaire) nor scores on a retention test differed significantly between the two conditions (between subjects; $N = 28$). However, users in the *with-gestures* condition complained significantly more about the quality of the speech, perhaps because the gestures detracted from the on-screen speech bubbles that accompanied speech. Furthermore, to evaluate an extension of eShowroom with *backchannelling* gestures by the hearer, we compared dialogues with and without hearers' gestures, keeping speakers' gestures constant (between subjects; $N = 12$; see [9] and [67]). We found that subjects in the *with-hearer* gestures condition did significantly worse on the retention test, possibly because the hearer gestures were too intrusive. This would be consistent with [87], where the presence of a highly expressive *talking head* was argued to diminish task performance in some cases, because it can distract attention. A possible alternative explanation for our findings is that the rendering of the feedback gestures may not have been good enough. For the purpose of this particular study, we used the Microsoft Agents technology, which does not always render simultaneous gestures by multiple agents adequately: gestures can be a bit abrupt, for example.

Finally, in a study with $N = 40$ (see [67]), we found a small effect as a result of varying the algorithm for the generation of referring expressions. A more "ego-centric" algorithm (an agent ignoring the contributions of his interlocutor) caused the agent to appear less friendly.

6. Speech

The generation of speech is performed using the text-to-speech (TTS) system MARY [77]. While existing TTS technology is of sufficient quality to be intelligible, there is much room for improvement, particularly if personality and emotion are to be taken into account. NECA makes two contributions to this long-term goal: linguistically appropriate prosody in a dialogue, and emotional expressivity.

6.1. Prosody reflecting information structure

The term "prosody" covers the supra-segmental aspects of a speech utterance: pitch, duration, and loudness. Prosody can not only convey information about the affective state of the speaker, but also about the linguistic structure of the utterance, for example by accenting new or important words, and by inserting pauses. Despite work by, for example, [35,57,69], and [36], existing TTS technology usually does not take such effects into account. Systems based on NLG, however, are well placed to do better. This is particularly true for NECA, whose incremental processing model (Section 3) guarantees that semantic, syntactic and pragmatic information is available to the Speech Synthesis module. This makes it easy, for example, to look up whether a given object represents "given" or "new" information, without having to parse and interpret text.

Information structure is realised by an interplay of various linguistic means or strategies. These means are either syntactic (e.g. word order and specific constructions like clefts, passives and parallelism), morpho-syntactic (e.g. specific particles), or prosodic (e.g. (de)accentuation and intonational phrasing) in nature and are employed by different languages to different degrees (e.g. [85]). In English, intonation is the predominant linguistic marker of information structure, which also holds for German, although word order plays a more important role here.

NECA's treatment of prosody is based heavily on the RRL and our incremental processing model while, empirically, it is informed by extensive perception tests. Here we summarise some of our main results for German [5,7,8]. Broadly speaking, the results confirmed the familiar idea that new information should carry an accent while textually given information is de-accented (e.g. [84]). We also found, however, that when the *type* of accent is taken into account, it is necessary to distinguish more finely than is usually done, by taking a third type of information into account, which is sometimes called "accessible" [18,50]. Such information is neither totally new nor totally given but inferable from the situational or textual context. For textually inferable referents, we found that the nature of the semantic relation with the antecedent determines whether an item should be accented, and which type of accent it should carry. For example, synonyms (*elevator–lift*) and the anaphors in part-whole relations (*page–book*) tend to behave similarly to given information and are usually de-accented, while e.g. the anaphor in a whole-part relation (i.e. the reverse order of the inclusion relation, e.g. *book–page*) is more similar to new information and should be accented.

The type of accent on the subordinate expression is different from an accent marking new information, however. It could be shown that an early peak accent (transcribed as H+L* in terms of the often-used (G)ToBI categorisation; see [32]) is most appropriate for marking this type of accessibility, whereas a medial peak accent (symbolised as H*) is best for marking new information.

Semantic-pragmatic properties of a referring expression including its degree and type of givenness are provided by the NLG component. This information is used to assign the tags "+given" and "+accessible" (if applicable) to the respective items in the RRL script. Furthermore, a contrastive usage of a referring expression can be explicitly flagged.

These markers of a referring expression's information status are communicated to the MARY prosody module, where they affect accent placement and form: Tokens marked "+given" are ignored during accent assignment, i.e. they are de-accented, whereas tokens marked as "+accessible" are assigned an H+L* accent, and "+contrast" tokens receive a rising accent (L+H*) with a particularly high pitch range. The default nuclear accent type assigned to new adjectives and nouns is H*. These rules enable the system to generate contextually appropriate intonation patterns.

6.2. Emotionally expressive speech

We have argued that it is often crucial for the dialogues produced in Scripted Dialogue to be expressive in terms of the emotional state of the speaker. Two types of generating emotionally expressive speech can be distinguished: "playback" and "model" approaches. The first approach (e.g., [10,38]) treats emotions holistically by creating speech synthesis voices from recordings spoken in certain expressive styles (e.g., angry voice, friendly voice). While this approach is likely to lead to highly natural emotion expression, it suffers from a lack of *flexibility*: Only the emotional states which have been recorded can be "played back". Clearly, NECA's goal of creating dialogues that are highly varied makes flexibility a key issue: the alternative would be to record and store prohibitively large amounts of speech.

The second approach (e.g. [58]) models emotions in terms of the acoustic synthesis parameters corresponding to various emotions. This approach requires a high degree of control over acoustic parameters. Rule-based formant synthesis enables the modelling of a wide array of acoustic parameters, which is why it has been the technology of choice for a number of emotional speech synthesis undertakings; however, due to lack of *naturalness*, it has nearly disappeared from the landscape of commercial speech synthesis systems. Promising new approaches, such as data-driven formant synthesis [13], are still in an early development phase. Unit selection yields the highest degree of naturalness in speech for one speaking style (usually: neutral), but does not provide a fine-grained control over the prosodic parameters. Indeed, unit selection synthesis draws its naturalness from *not* interfering with the recorded speech signal, and thus rarely allows for an explicit modelling of prosody. This limitation currently makes unit selection synthesis unsuitable for model-based approaches to emotional speech synthesis.

A compromise between degree of flexibility/control and natural-sounding synthesis is diphone synthesis, which allows fine-grained prosody modelling with a limited degree of distortion. It is based on the concatenation of small recordings of human speech, so-called "diphones" (ranging from the middle of one phone segment to the middle of the following phone segment), followed by a signal processing step to generate the desired prosody. Unfortunately, the voice quality inherent in the diphones appears to be inappropriate for certain emotions [58].

The current work pursues a model-based approach to synthesis, i.e. it is based on an explicit model of the vocal correlates of emotions, realised using a diphone synthesis enhanced with a limited control over voice quality [78]. We start with the decision on how to represent the emotional states themselves [19]. Consistent with the state of the art in speech research, we have chosen to use *emotion dimensions* [20,74], a continuous framework for the representation of essential properties of emotional states. The two emotion dimensions which have emerged as most important from a large number of studies are evaluation (sometimes called valence or pleasure) and activation (sometimes called arousal). These two are sometimes complemented with a third dimension, called power or dominance.

The main task in building the model is to find a mapping from a point in the emotion dimension space to the corresponding acoustic correlates. We constructed such a mapping based on a database analysis and a literature survey [75,78]. We used the Belfast Naturalistic Emotion Database, which contains recordings of 124 English speakers exhibiting relatively spontaneous emotion [24]. This database is one of the largest collections of natural emotional speech available, and it is labelled according to emotion dimensions. The emotion dimension coordinates of each clip in the database were correlated with a number of acoustic measures that were semi-automatically extracted from the database. Robust correlations were found, especially for the activation dimension, but also—if to a lesser extent—for the evaluation and power dimensions. These correlations were accompanied by quantified linear prediction coefficients, allowing a relatively simple deduction of rules for synthesis.

As a second source of information, a literature survey was conducted. The assorted evidence found in a dozen publications was brought together, most of which studied English speech (see [66] for details). While these articles only gave qualitative trends on correlations between emotion dimensions and acoustic parameters, they provided a solid baseline for what can be expected to be conveyed through acoustic voice parameters. Essentially, strong trends were found only for the activation dimension.

All this evidence was consolidated into a model that predicts prosodic and voice quality changes for each point in emotion dimension space. The evidence confirmed that the emotion dimension best conveyed in speech is *activation* (or "arousal"), i.e. the degree of excitation vs. passivity. According to our model, increased activation is conveyed in the voice through prosodic effects, such as increased pitch and speaking rate, as well as the voice quality, particularly an increased vocal effort caused by higher muscle tension.

The importance of voice quality modelling for expressing emotions in synthetic speech is still a matter for debate [78]. In essence, the frequent presence of voice quality effects in human expressions of emotion make it desirable

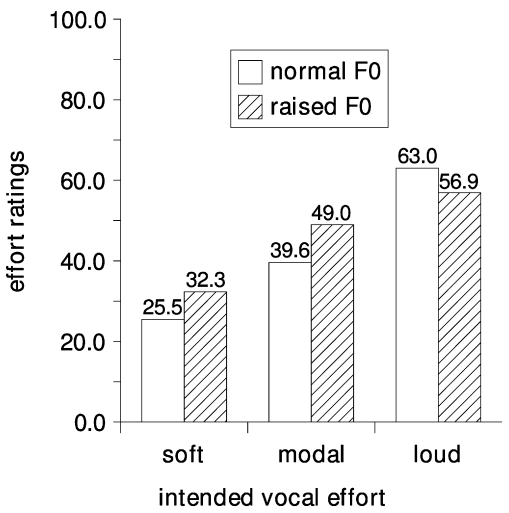

Fig. 12. Effort ratings for the male diphone voice by German listeners (from [76]).

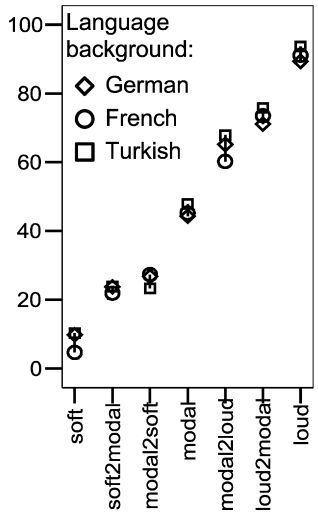

Fig. 13. Effort ratings for female diphone voice and interpolated versions, by German, French and Turkish listeners (from [82]).

to model voice quality in synthetic speech. Since there are no instruments yet for modelling voice quality in diphone synthesis (in spite of promising developments, see [43,80]), we recorded separate diphone databases for three levels of vocal effort, for one male and one female speaker. Both voices are publicly available for non-commercial use (http://tcts.fpms.ac.be/synthesis/mbrola.html), as the MBROLA [25] diphone databases de6 (male) and de7 (female).

We tested the perceptual adequacy of our male voice in two perception tests [76]. A first test was carried out to test the hypothesis that the three diphone sets are sufficiently similar to be recognised as belonging to the same person. We prepared pairs of sentences, where the first and the second sentence were synthesised either using the same voice or a different one, at the same or a different pitch levels. Subjects were asked whether the stimuli in each pair were produced by the same speaker. Results showed that the effect of vocal effort on perceived speaker identity was relatively small – 79.9% of the sentences differing in vocal effort were perceived as the same speaker. However, there was a strong effect of pitch level. A modification of pitch slightly beyond the range typically used in non-emotional synthesis (but still moderate in view of emotional speech) caused speaker identity ratings to drop to around or below chance level. Next, we tested the hypothesis that the effort intended during recording is perceived in the synthesised material. Stimuli differing in intended vocal effort and in overall pitch level were played to subjects, who rated the stimuli on a continuous scale from "without effort" to "with great effort". Since the stimuli were amplitude-normalised, subjects were instructed to base their ratings on the "sound of the voice" rather than the "loudness". Results confirmed that the effort was perceived as intended (Fig. 12).

While being able to select one of three levels of vocal effort is a step forward, this is clearly a very limited amount of control. A further step towards more flexibility can be afforded by the use of voice interpolation. From the original recordings of the three female voice databases, we created new databases with intermediate levels of vocal effort using a simple spectral interpolation algorithm [82]. A listening test was performed to evaluate the intended vocal effort in the original female databases and the interpolated ones. The results show that the interpolation algorithm can create the intended intermediate levels of vocal effort given by the original databases. This effect was largely independent of the language background of the subjects (Fig. 13).

7. Generating dialogue-accompanying gestures

We have arrived at the last step, where facial expressions, hand-arm gestures and postures are chosen and aligned with speech. Since we were able to build on established techniques and procedures in this area, the description of this part of the NECA approach will be comparatively brief. Dialogue-accompanying gestures such as facial expression, hand-arm gestures and body postures are typically generated in two phases: a *planning* phase and a *realisation* phase (see [46] and their proposal of the SAIBA framework). We discuss these two phases in turn.

7.1. Multimodal behaviour planning

During Multimodal Natural Language Generation (MNLG), gestures are planned on the basis of the semantic and pragmatic content of the utterances and are aligned to the respective nodes in the MNLG tree. (See the example of <dialogueAct> in Section 3.2.) The actual point in time for the start of the gesture is still unknown at this stage of processing. When the Gesture Assignment module starts, information on body behaviour is underspecified. This is firstly because only information on relative alignment of verbal and non-verbal behaviour is available (see ALIGNTO and ALIGNTYPE features in the example of <dialogueAct> after MNLG in Section 3.2), and secondly because the choice of gestures is only restricted by the features IDENTIFIER, and MODALITY (i.e. in our example all gestures involving body and hips are suitable).

The idea of intertwining gestural and syntactic structure has also been proposed in [16]. They describe a mechanism for applying the SPUD natural language generator [79] with its Lexicalized Tree Adjoining Grammar (LTAG) formalism [39,73] to multimodal generation. Integration of gestures and syntax is particularly suitable for gestures that can express semantic content and therefore present an alternative to linguistic expression of the same content. The MNLG also allows for a second type of gesture generation, which is less tightly integrated with syntax. This second type of gesture generation concerns gestures expressing discourse function (e.g., question or assertion). Such gestures are not part of the grammar, but are added by a separate gesture generation module which associates gestures and body postures with particular types of dialogue acts.

7.2. Temporal fine-tuning of behaviours

At a later planning stage, during Gesture Assignment, the relative alignment of utterances and gestures resulting from the behaviour planning stage (MNLG) is transformed into an absolute alignment according to the time constraints imposed by speech synthesis. This approach is typical for ECA systems [44]. More generally, accessibility of prosodic and temporal information produced by the speech synthesis is crucial for a fine-grained alignment of the verbal and non-verbal communication systems.

In NECA, the speech synthesis component provides a sound file of the utterance together with an RRL file containing the phonetic transcription of the utterance.[7] (See Section 3.2, "Specification of <sentence> after Speech Synthesis".) This information together with the constraints coming from MNLG, and the meta-level description of available gestures in the Gesticon (see the next subsection), is then used by the Gesture Assignment module for producing player-independent multimodal animation directives. The animation directives are encoded in the animationSpec-element of the RRL which is then transformed into player-specific formats. For an example of an animationSpec see, once again, Section 3.2.

7.3. Gesture representation—The gesticon

Information on gestures is stored in a RRL-compliant repository of behaviour descriptions which we call *Gesticon*. Analogous to a *Lexicon* in natural language, a *Gesticon* is a central behaviour repository relating form with meaning and function, and moreover connecting the abstract information to concrete player-specific animations.

When defining the *Gesticon* for the NECA applications eShowroom and Socialite, we started out from descriptions comprising some minimal information on the meaning or function of a gesture (e.g. deictic, or greeting) or facial expression (happy, sad, disgusted, etc.), and a high-level description of form features, such as which body parts are involved and the relative duration of gestures and gesture phases [49]. Duration information specifies the extent to which a gesture can be elongated or shrunk without changing its meaning. For hand-arm gestures the relative wrist position at the beginning and the end of the gesture is also stored [47]. This information is used to estimate the time required to move from the end of one gesture to the beginning of the following gesture.

The need for representations of body behaviours that are independent of animation and player technology has arisen from the wish to develop planning components that are independent of individual animation and player technologies.

[7] For an overview on TOBI see http://www.ling.ohio-state.edu/~tobi/.

The *Gesticon* functions as a central behaviour repository relating form with meaning and function, and connecting the abstract information to concrete player-specific animations.

Example: Gesticon entry for a right hand wave	
<gesticonEntry identifier="g_wave_righthand" modality="arm">	
<function>greeting</function>	In the context of the NECA applications a wave signals greeting.
<form> <position> <start right="RU"/> <end right="RU"/> </position> <components> <stroke> <dur min="655" default="655" max="10000" /> </stroke> </form>	The gesture is positioned in the right upper (RU) quadrant of a cube encapsulating the character's body. The duration of the wave must not be shorter than 655 milliseconds and must not exceed a second.
<playercode type="flash" id="61_1"/> <playercode type="charamel" id="wave3"/>	The concrete animations are stored in a Flash file (61_1) and a Charamel file (wave3).

The *eShowroom* animation library consists of 160 animation videos (in Charamel's CharActor format) which define small sequences of overall body behaviour including hand-arm gestures for the male and the female character. The behaviours are built from basic graphical building blocks such as face shapes, eye and mouth shapes, hand shapes, upper arms, lower arms. For the facial display of emotions such as anger and fear, animation directives are formulated in terms of degree of eyebrow and lip corner raise, lip stretch, and so on. In Socialite, character animation is restricted to facial animation and hand-arm gestures. Its animation library is a collection of Flash-encoded hand-arm gestures (53 base gestures) and snapshots of facial expressions (19 for the male and the female character each). Facial expressions in Socialite are based on Ekman's six basic emotions of (happiness, sadness, anger, fear, disgust, surprise) plus a few fagin-style additional labels like 'false laugh', and 'reproach'.

The approach to animation pursued in NECA is comparable to the majority of current work on ECAs where behaviours are realised by selecting from a set of prefabricated animations, see for instance the REA system [15], the NICE project [5], FearNot [30]. These differ from approaches where behaviours are *generated*; see for instance [62] for generating facial expressions from speech, Tepper et al., 2004 for generating direction-giving gestures from semantic representations, or [45] for driving a virtual character by means of form descriptions derived from motion capture.

8. Conclusion

The NECA approach to Fully generated Scripted Dialogue (FGSD), as embodied in the eShowroom and Socialite demonstrators developed in the NECA project build on such predecessors as those described in [14], but it represents

a significant step forward in the construction of systems involving ECAs that are able to engage in a large variety of highly expressive dialogues. In summarising its highlights, it will be useful to distinguish between three issues: (1) the overall paradigm of Scripted Dialogue, (2) the architecture that is used in NECA to produce scripted dialogue, and (3) the individual components of the NECA system.

1. *The paradigm of scripted dialogue.* ECAs are widely thought to have a potentially beneficial effect on the motivation and task performance of the user of a computer application. Lester et al., for example, show that "[...] the presence of a lifelike character in an interactive learning environment—even one that is not expressive—can have a strong positive effect on student's perception of their learning experience", calling this the *Persona Effect* ([51], also [23]). We have argued that Fully Generated Scripted Dialogue (FGSD) is a promising framework in which to purpose these potential benefits. We believe there to be a wealth of applications, ranging from "edutainment" (e.g., VirtualConstructor, [59]) to advertising and e-drama (witness Carmen's Bright IDEAS [47], FearNot! [4,34] and Façade [55]), where it can be useful to generate a dialogue as a whole. Similarly, FGSD could be used to increase the variety of dialogues produced by story generation systems (e.g. [12]), particularly those that are multimodal [17, 55]. Computer-generated animations have become part of mainstream cinematography, as witnessed by films such as Finding Nemo, Monsters Inc., and Polar Express; but automated creation of film *content*, and more specifically, dialogue content, lags behind the possibilities currently explored for graphics. We hope that the FGSD paradigm advocated in this paper will contribute towards closing this gap.

The fact that NECA's dialogues are *fully generated* makes it possible to generate a huge variety of dialogues whose wording, speech and body language are in accordance with the interests, personalities and affective states of the agents. The degree of control can be further enhanced if a revision strategy is applied, which takes the output of the Scene Generator as a first approximation that can be optimised through later operations [66,68]. Consider the eShowroom scenario, for instance. If two or more yes/no questions about a car are similar in structure while also eliciting the same response, then these question/answer pairs can be merged into one aggregated question-answer pair (*'Does this car have power windows and leather seats? Sure, it has both!'*).

2. *Architecture and processing model*. Scripted dialogues can be generated in many different ways. A distinctive feature of the NECA system is the fact that it is based on a processing model that starts from a *scene* generated by the Scene Generator, which is then incrementally "decorated" with more and more information, of a linguistic, phonetic, and graphical nature. The backbone of this incrementally-enhanced representation is NECA's Rich Representation Language (RRL), which is based on XML. Perhaps the best defence for this incremental processing model lies in the experimental and multidisciplinary nature of all work on ECAs. Partly because this is still a young research area, it is difficult to predict which aspects of a given level of representation might be needed by later modules. This difficulty is exacerbated by the fact that researchers/programmers may only have a limited understanding of what goes on in later modules. By keeping the generation process incremental (i.e., monotonically increasing), we guarantee that all information produced by a given module will be available to all later modules.

Consider, for example, the information status of referents in the domain. It may not be obvious to someone working on MNLG that the novelty or givenness (i.e., roughly, the absence or presence in the Common Ground) of an object is of any importance to later modules; but it is of importance since, for example, this information is used by Speech Synthesis when deciding whether to put a particular kind of pitch accent on the Noun Phrase referring to this object (Section 6.1). Our incremental processing model ensures that this information is in fact available. Undeniably, this processing model can lead to XML structures that are large. As a remedy we have implemented a streaming model where after Scene Generation the individual communication acts are processed in parallel. As soon as the player generator has finished processing an act, the result is "streamed" to the user immediately, while subsequent acts are still being processed. This leads to a drastic reduction of response times and thus ensures real-time behaviour of the system.

3. *Individual system components*. When different scientific disciplines join forces to construct an ECA-based system, it can be interesting to compare their respective contributions. Comparisons could be made across modalities, for example, asking how basic concepts such as *information structure* (e.g., focus) are expressed in the different modalities (i.e., text, speech, and body language). Another interesting question is why *emotions* are modelled differently in Affective Reasoning (which uses the OCC model of [61]) and in Speech (where Schlosberg's emotion dimensions are thought to be more appropriate), and in facial expressions (where Ekman's six basic emotions hold sway). For reasons of space, we shall focus on one comparison that is particularly important given NECA's emphasis on generic

technologies that hold promise for the longer term, namely the trade-off between *quality* and *flexibility* which has featured strongly in our discussions of both Natural Language Generation and Speech Synthesis.

The issues regarding quality and flexibility might be likened to a problem in the construction of real estate. Suppose an architect wants to restore an old stone building in grand style. Ideally, she might want to harvest some natural stone in all the colours and shapes that the restoration work requires. But it can be difficult to find exactly the right piece, in which case she can either make do with a natural piece that is not exactly right, or she might have a piece of *artificial* (i.e., reconstituted) stone custom made.

The trade-offs facing language generation, speech synthesis and gesture assignment are similar. In the case of Natural Language Generation, NECA has used a combination of canned text (cf. natural stone) with fully compositional generation (cf. artificial stone); in the case of speech synthesis, NECA has used a combination of diphone synthesis (comparable with grinding natural stone to a pulp which is then moulded in the desired shape) with limited control over voice quality. In order to create suitable animations, NECA has employed libraries of player-specific, prefabricated animations (cf. giving architects a choice of different rooms, facades, etc.) together with meta-information concerning dimensions of scalability; this approach to graphics is comparable to parameterised unit selection in Speech Synthesis, or to the highly flexible kind of template-based Natural Language Generation advocated in [83].

Closing remarks. The word 'dialogue' can be taken to imply interaction between a computer agent and a person. In this paper, we have examined an alternative perspective on dialogue, as a way to let Embodied Conversational Agents present information (e.g., about cars in the *eShowroom* system) or to tell a story (e.g. about students in the *Socialite* system). NECA's version of Scripted Dialogue happens not to allow very sophisticated interactions with the user. (The interface of Fig. 1, Section 2, for example, only allows the user to choose between four different personalities and 256 different combinations of value dimensions, using a simple menu.) We believe there to be ample space for other, similarly direct applications of the fully-generated scripted dialogue (FGSD) technology, for example because there will always be a place for *non*-interactive radio, film and television. Perhaps most importantly, however, we see a substantial future role for *hybrid* systems that combine FGSD with much extended facilities for letting the user influence the behaviour of the system (as exist in interactive drama, for example, see [4,34,54,55]).[8]

References

[1] E. André, T. Rist, Presenting through performing: On the use of life-like characters in knowledge-based presentation systems, in: Proceedings IUI '2000: International Conference on Intelligent User Interfaces, 2000.
[2] E. André, T. Rist, S. van Mulken, M. Klesen, S. Baldes, The automated design of believable dialogues for animated presentation teams, in: J. Cassell, J. Sullivan, S. Prevost, E. Churchill (Eds.), Embodied Conversational Agents, MIT Press, Cambridge, 2000.
[3] E. André, M. Klesen, P. Gebhard, S. Allen, T. Rist, Integrating models of personality and emotions into lifelike characters, in: A. Paiva (Ed.), Affective Interactions: Towards a New Generation of Computer Interfaces, in: Lecture Notes in Computer Science, vol. 1814, Springer, Berlin, 2000.
[4] R.S. Aylett, R. Figuieredo, S. Louchart, J. Dias, A. Paiva, Making it up as you go along—improvising stories for pedagogical purposes, in: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), 6th International Conference, IVA 2006, in: LNAI, vol. 4133, Springer, Berlin, 2006, pp. 307–315.
[5] S. Baumann, M. Grice, The intonation of accessibility, Journal of Pragmatics 38 (10) (2006) 1636–1657.
[6] S. Baumann, M. Grice, S. Steindamm, Prosodic marking of focus domains—categorical or gradient? in: Proceedings SpeechProsody 2006, Dresden, Germany, 2006, pp. 301–304.
[7] S. Baumann, M. Grice, Accenting accessible information, in: Proceedings Speech Prosody 2004, Nara, Japan, 2004, pp. 21–24.
[8] S. Baumann, K. Hadelich, On the perception of intonationally marked givenness after auditory and visual priming, in: Proceedings AAI Workshop "Prosodic Interfaces", Nantes, France, 2003, pp. 21–26.
[9] M. Bergenstråhle, Feedback gesture generation for embodied conversational agents, Technical Report ITRI-03-22, ITRI, University of Brighton, UK, 2003.
[10] M. Bulut, S.S. Narayanan, A.K. Syrdal, Expressive speech synthesis using a concatenative synthesiser, in: Proceedings of the 7th International Conference on Spoken Language Processing, Denver, Colorado, USA, 2002.
[11] S. Busemann, H. Horacek, A flexible shallow approach to text generation, in: Proceedings 9th International Workshop on Natural Language Generation, Canada, 1998, pp. 238–247.
[12] Ch.B. Callaway, J.C. Lester, Narrative prose generation, Artificial Intelligence 139 (2) (2002) 213–252.
[13] R. Carlson, T. Sigvardson, A. Sjölander, Data-driven formant synthesis, Progress Report No. 44, KTH Stockholm, Sweden, 2002.
[14] J. Cassell, J. Sullivan, S. Prevost, E. Churchill (Eds.), Embodied Conversational Agents, MIT Press, Cambridge, MA, 2000.

[8] Carmen's Bright IDEAS and FearNot! apply interactive drama to education: IDEAS is designed to help mothers of young cancer patients; FearNot! trains school children to cope with bullying. Façade is an interactive game in which the user influences the outcome of the game.

[15] J. Cassell, M. Stone, H. Yan, Coordination and context-dependence in the generation of embodied conversation, in: Proceedings First International Natural Language Generation Conference (INLG'2000), Mitzpe Ramon, Israel, 2000, pp. 12–16.
[16] J. Cassell, H. Vilhjálmsson, T. Bickmore, BEAT: The behavior expression animation toolkit, in: Proceedings ACM SIGGRAPH 2001, Los Angeles, USA, 2001, pp. 477–486.
[17] M. Cavazza, M. Charles, Dialogue generation in character-based interactive storytelling, in: Proceedings AIIDE, 2005.
[18] W. Chafe, Discourse, Consciousness, and Time, University of Chicago Press, Chicago/London, 1994.
[19] R. Cowie, R.R. Cornelius, Describing the emotional states that are expressed in speech, Speech Communication 40 (1–2) (2003) 5–32 (Special Issue on Speech and Emotion).
[20] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, J. Taylor, Emotion recognition in human–computer interaction, IEEE Signal Processing Magazine 18 (1) (2001) 32–80.
[21] A. De Angeli, N. Bianchi-Berthouze (Eds.), Proceedings AVI 2006 Workshop on Gender and Interaction: Real and Virtual Women in a Male World, Venice, Italy, 2006.
[22] B. De Carolis, C. Pelachaud, I. Poggi, M. Steedman, APML, a mark-up language for believable behavior generation, in: H. Prendinger (Ed.), Life-Like Characters. Tools, Affective Functions and Applications, Springer, Berlin, 2004.
[23] D.M. Dehn, S. Van Mulken, The impact of animated interface agents: A review of empirical research, Journal of Human–Computer Studies 52 (1) (2000) 1–22.
[24] E. Douglas-Cowie, N. Campbell, R. Cowie, P. Roach, Emotional speech: Towards a new generation of databases, Speech Communication 40 (1–2) (2003) 33–60 (Special Issue Speech and Emotion).
[25] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, O.V. Vrecken, The mbrola project: Towards a set of high quality speech synthesisers free of use for non-commercial purposes, in: Proceedings 4th International Conference of Spoken Language Processing, Philadelphia, USA, pp. 1393–1396.
[26] K.R. Echavarria, M. Généreux, D. Arnold, A. Day, J. Glauert, Multilingual virtual city guides, in: Proceedings Graphicon, Novosibirsk, Russia, 2005.
[27] M. Elhadad, FUF/SURGE Homepage, Available from: http://www.cs.bgu.ac.il/surge, 19 September 2006.
[28] G. Erbach, Profit 1.54 user's guide, University of the Saarland, December 3, 1995.
[29] P. Gebhard, M. Kipp, M. Klesen, T. Rist, Adding the emotional dimension to scripting character dialogues, in: Proceedings 4th International Working Conference on Intelligent Virtual Agents (IVA'03).
[30] P. Gebhard, ALMA—a layered model of affect, in: Proceedings 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05), Utrecht, Netherlands, 2005, pp. 29–36.
[31] P. Gebhard, K.H. Kipp, Are computer-generated emotions and moods plausible to humans? in: Proceedings 6th International Conference on Intelligent Virtual Agents (IVA'06), Marina Del Rey, USA, 2006.
[32] M. Grice, S. Baumann, R. Benzmüller, German intonation in autosegmental-metrical phonology, in: A. Jun (Ed.), Prosodic Typology. The Phonology of Intonation and Phrasing, Oxford University Press, Oxford, 2005, pp. 55–83.
[33] E. Gstrein, C. Schmotzer, B. Krenn, Report on demonstrator evaluation results, NECA IST report D9e, July 2004, downloadable from http://www.ofai.at/research/nlu/NECA/publications/publication_docs/d9e.pdf.
[34] L. Hall, M. Vala, M. Hall, M. Webster, S. Woods, A. Gordon, R. Aylett, FearNot's appearance: Reflecting children's expectations and perspectives, in: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), Proceedings 6th International Conference, IVA 2006, in: LNAI, vol. 4133, Springer, Berlin, 2006, pp. 407–419.
[35] J. Hirschberg, Pitch accent in context: Predicting intonational prominence from text, Artificial Intelligence 63 (1993) 305–340.
[36] L. Hiyakumoto, S. Prevost, J. Cassell, Semantic and discourse information for text-to-speech intonation, in: ACL Workshop on Concept-to-Speech Technology, 1997.
[37] Z. Huang, A. Eliens, C. Visser, XSTEP: A markup language for embodied agents, in: Proceedings 16th International Conference on Computer Animation and Social Agents (CASA'2003), IEEE Press, 2003.
[38] A. Iida, N. Campbell, S. Iga, F. Higuchi, M.A. Yasumura, Speech synthesis system with emotion for assisting communication, in: Proceedings ISCA Workshop on Speech and Emotion, Northern Ireland, 2000, pp. 167–172.
[39] A.K. Joshi, L. Levy, M. Takahashi, Tree adjunct grammars, Journal of the Computer and System Sciences 10 (1975) 136–163.
[40] H. Kamp, U. Reyle, From Discourse to Logic, Kluwer, Dordrecht, 1993.
[41] M. Kantrowitz, GLINDA: Natural language text generation in the oz interactive fiction project, Technical Report CMU-CS-90-158, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1990.
[42] M. Kantrowitz, J. Bates, Integrated natural language generation systems, in: Aspects of Automated Natural Language Generation, in: D. Roesner, O. Stock (Eds.), NAI, vol. 587, Springer, Berlin, 1992.
[43] H. Kasuya, K. Maekawa, S. Kiritani, Joint estimation of voice source and vocal tract parameters as applied to the study of voice source dynamics, in: Proceedings 14th International Conference of Phonetic Sciences, San Francisco, USA, pp. 2505–2512.
[44] S. Kopp, I. Wachsmuth, Synthesizing multimodal utterances for conversational agents, Computer Animation and Virtual Worlds 15 (1) (2004) 39–52.
[45] S. Kopp, T. Sowa, I. Wachsmuth, Imitation games with an artificial agent: From mimicking to understanding shape-related iconic gestures, in: X. Camurri, X. Volpe (Eds.), Gesture-Based Communication in Human–Computer Interaction, in: LNAI, vol. 2915, Springer, Berlin, 2004, pp. 436–447, http://www.techfak.uni-bielefeld.de/%7Eskopp/download/gesture_imitation_GW03.pdf.
[46] S. Kopp, B. Krenn, S. Marsella, A. Marshall, C. Pelachaud, H. Pirker, K. Thorisson, H. Vilhjalmsson, Towards a common framework for multimodal generation in ECAs: The behavior markup language, in: J. Gratch, et al. (Eds.), Intelligent Virtual Agents 2006, in: LNAI, vol. 4133, Springer, Berlin, 2006, pp. 205–217.
[47] A. Kranstedt, S. Kopp, I. Wachsmuth, MURML: A multimodal utterance representation markup language for conversational agents, in: Proceedings AAMAS'02 Workshop Embodied conversational agents—let's specify and evaluate them! Bologna, Italy, 2002.

[48] B. Krenn, B. Neumayr, E. Gstrein, M. Grice, Lifelike agents for the Internet: A cross-cultural case study, in: S. Payr, R. Trappl (Eds.), Agent Culture: Human–Agent Interaction in a Multicultural World, Lawrence Erlbaum Associates, NJ, 2004, pp. 197–229.
[49] B. Krenn, H. Pirker, Defining the gesticon: Language and gesture coordination for interacting embodied agents, in: Proceedings AISB-2004 Symposium on Language, Speech and Gesture for Expressive Characters, University of Leeds, UK, 2004, pp. 107–115.
[50] K. Lambrecht, Information Structure and Sentence Form, Cambridge University Press, Cambridge, 1994.
[51] J.C. Lester, S.A. Converse, S.E. Kahler, S.T. Barlow, B.A. Stone, R.S. Bhoga, The persona effect: Affective impact of animated pedagogical agents, in: Proceedings CHI Conference, Atlanta, Georgia, 1997.
[52] W. Levelt, Speaking: From Intention to Articulation, MIT Press, Cambridge, MA, 1989.
[53] A. Loyall, Believable agents: Building interactive personalities, Ph.D. thesis, CMU, Technical Report CMU-CS-97-123.
[54] S. Marsella, W.L. Johnson, C. LaBore, Interactive pedagogical drama for health interventions, in: AIED 2003, 11th International Conference on Artificial Intelligence in Education, Australia, 2003.
[55] M. Mateas, A. Stern, Facade: An experiment in building a fully-realized interactive drama, in: Game Developer's Conference: Game Design Track, San Jose, California, 2003.
[56] S. McRoy, S. Channarukul, S. Ali, An augmented template-based approach to text realization, Natural Language Engineering 9 (4) (2003) 381–420.
[57] A. Monaghan, Intonation in a text-to-speech conversion system, Ph.D. thesis, University of Edinburgh, 1991.
[58] J.M. Montero, J. Gutiérrez-Arriola, J. Colás, E. Enríquez, J.M. Pardo, Analysis and modelling of emotional speech in Spanish, in: Proceedings 14th International Conference of Phonetic Sciences, San Francisco, USA, 1999, pp. 957–960.
[59] A. Ndiaye, P. Gebhard, M. Kipp, M. Klesen, M. Schneider, W. Wahlster, Ambient intelligence in edutainment: Tangible interaction with life-like exhibit guides, in: Proceedings Conference on INtelligent TEchnologies for interactive entertainment (INTETAIN'05), Madonna di Campiglio, Italy, 2005.
[60] N. Nicolov, C. Mellish, G. Ritchie, Approximate generation from non-hierarchical representations, in: Proceedings 8th International Workshop on Natural Language Generation, Herstmonceux Castle, UK, 1996.
[61] A. Ortony, G.L. Clore, A. Collins, The Cognitive Structure of Emotions, Cambridge University Press, Cambridge, MA, 1988.
[62] C. Pelachaud, N.I. Badler, M. Steedman, Generating facial expressions for speech, Cognitive Science 20 (1) (1996) 1–46.
[63] P. Piwek, B. Krenn, M. Schröder, M. Grice, S. Baumann, H. Pirker, RRL: A rich representation language for the description of agent behaviour in NECA, in: Proceedings AAMAS Workshop Embodied Conversational Agents—Let's Specify and Evaluate Them! Bologna, Italy, 2002.
[64] P. Piwek, A flexible pragmatics-driven language generator for animated agents, in: Proceedings of EACL (Research Notes), Budapest, Hungary, 2003.
[65] P. Piwek, The effect of gestures on the perception of a dialogue between two embodied conversational agents: a pilot study, Technical Report ITRI-03-09, ITRI, University of Brighton, UK, 2003.
[66] P. Piwek, K. van Deemter, Dialogue as discourse: Controlling global properties of scripted dialogue, in: Proceedings AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue, Stanford, California, 2003.
[67] P. Piwek, J. Masthoff, M. Bergenstrahle, Reference and gestures in dialogue generation: Three studies with embodied conversational agents, in: Proceedings AISB05 Joint Symposium on Virtual Social Agents Symposium, University of Herfordshire, UK, 2005, pp. 53–60.
[68] P. Piwek, K. Van Deemter, Generating under global constraints: The case of scripted dialogue, Journal of Research on Language and Computation 5 (2) (2007) 237–263.
[69] S. Prevost, M. Steedman, Specifying intonation from context for speech synthesis, Speech Communication 15 (1994) 139–153.
[70] W. Reilly, Believable social and emotional agents, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, 1996.
[71] E. Reiter, R. Dale, Building Natural Language Generation Systems, Cambridge University Press, Cambridge, 2000.
[72] J. Rickel, W.L. Johnson, Animated agents for procedural training in virtual reality: Perception, cognition and motor control, Applied Artificial Intelligence 13 (1999) 343–382.
[73] Y. Schabes, Mathematical and computational aspects of lexicalized grammars, Ph.D. thesis, Computer Science Department, University of Pennsylvania, 1990.
[74] H. Schlosberg, A scale for the judgement of facial expressions, Journal of Experimental Psychology 29 (1941) 497–510.
[75] M. Schröder, Speech and emotion research: An overview of research frameworks and a dimensional approach to emotional speech synthesis, Ph.D. thesis, Institute of Phonetics, Saarland University (Phonus 7), 2004.
[76] M. Schröder, M. Grice, Expressing vocal effort in concatenative synthesis, in: Proceedings 15th International Conference of Phonetic Sciences, Barcelona, Spain, 2003.
[77] M. Schröder, J. Trouvain, The German text-to-speech synthesis system MARY: A tool for research, development and teaching, International Journal of Speech Technology 6 (2003) 365–377.
[78] M. Schröder, Expressing degree of activation in synthetic speech, IEEE Transactions on Audio, Speech and Language Processing 14 (4) (2006) 1128–1136.
[79] M. Stone, T. Bleam, C. Doran, M. Palmer, Lexicalized grammar and the description of motion events, in: TAG+: Workshop on Tree-Adjoining Grammar and Related Formalisms, 2000.
[80] A. Tassa, J.S. Liénard, A new approach to the evaluation of vocal effort by the psola method, in: WEB-SLS, The European Student Journal of Language and Speech, 2000.
[81] D. Traum, J. Bos, R. Cooper, S. Larsson, I. Lewin, C. Matheson, M. Poesio, A model of dialogue moves and information state revision, Trindi Project Deliverable D2.1, 1999.
[82] O. Turk, M. Schröder, B. Bozkurt, L.M. Arslan, Voice quality interpolation for emotional text-to-speech synthesis, in: Proceedings Interspeech, Lisbon, Portugal, 2005, pp. 797–800.
[83] K. van Deemter, E. Krahmer, M. Theune, Real versus template-based natural language generation: A false opposition? Computational Linguistics 31 (1) (2005) 15–24.

[84] K. van Deemter, What's New? A semantic perspective on sentence accent, Journal of Semantics 11 (1994) 1–31.
[85] E. Vallduví, E. Engdahl, The linguistic realisation of information packaging, Linguistics 34 (1996) 459–519.
[86] H. Vilhjalmsson, Animating conversation in online games, in: M. Rauterberg (Ed.), Entertainment Computing ICEC, in: Lecture Notes in Computer Science, vol. 3166, Springer, Berlin, 2004, pp. 139–150.
[87] M. White, M.E. Foster, J. Oberlander, A. Brown, Using facial feedback to enhance turn-taking in a multimodal dialogue system, in: Proceedings HCI International, 2005.