Natural interaction with a virtual guide in a virtual environment

A multimodal dialogue system

Dennis Hofs · Mariët Theune · Rieks op den Akker

Received: 6 April 2009 / Accepted: 12 November 2009 / Published online: 5 December 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract This paper describes the Virtual Guide, a multimodal dialogue system represented by an embodied conversational agent that can help users to find their way in a virtual environment, while adapting its affective linguistic style to that of the user. We discuss the modular architecture of the system, and describe the entire loop from multimodal input analysis to multimodal output generation. We also describe how the Virtual Guide detects the level of politeness of the user's utterances in real-time during the dialogue and aligns its own language to that of the user, using different politeness strategies. Finally we report on our first user tests, and discuss some potential extensions to improve the system.

Keywords Conversational agent · Application · Social interaction · Politeness · Multimodal analysis and generation

1 Introduction

Some years ago we built a virtual music theatre, a 3D virtual replica of the 'Music Centre' in our hometown. It turned out that for many users navigating through the virtual environment by mouse or keyboard was hard, and for casual visitors the great music hall on the second floor was sometimes hard to find [16]. Therefore we decided to add a Virtual Guide,

a multimodal dialogue system with an embodied humanoid representation that helps users to find their way in the theatre by means of a multimodal dialogue combining gesture and spoken or typed natural language (Dutch).¹

Figure 1 shows the user interface of the system. The right part shows the embodied representation of the Virtual Guide standing behind the information desk in the hall of the virtual music centre. The top left part of the screen shows a 2D map of the building. The user can use either speech or typed textual input and support this with gesturing via mouse actions on the map. He can, for example, ask "What is this?" or request "Could you bring me here?" while pointing with the mouse at a location on the map. The system can show a route on the map by drawing lines between marked locations, and describe the route using speech and gestures. The ongoing dialogue is printed on the bottom left of the screen.

Developing user interfaces for natural interaction is one of the primary interests that motivates building multimodal dialogue systems with embodied conversational agents. For navigation, multimodal interaction via speech and pointing at a map is quite natural. Unlike the map-based multimodal systems discussed in [31], our system does not recognize arm or hand gestures captured by an electronic eye, but it does offer *symmetric multimodality*, which according to Wasinger and Wahlster creates "a natural experience for the user (...) by allowing both the user and the system to combine the same spectrum of modalities for the input as for the output" [40, p. 298].

The system implements the complete cycle from input processing to output generation. It demonstrates topics such as natural language processing and generation, dialogue management, speech technology, 3D animation and social

D. Hofs · M. Theune (✉) · R. op den Akker
Human Media Interaction, University of Twente, P.O. Box 217,
7500 AE Enschede, The Netherlands
e-mail: m.theune@ewi.utwente.nl

D. Hofs
e-mail: d.h.w.hofs@ewi.utwente.nl

R. op den Akker
e-mail: h.j.a.opdenakker@ewi.utwente.nl

¹You can try the system at <http://wwwhome.ewi.utwente.nl/~hofs/dialogue/>.

Fig. 1 User interface

interaction. As a readily available application with loosely integrated components, the Virtual Guide provides us with a platform for experiments and demonstrations of different strategies for the various topics. The components largely follow a generic design, which is supported by the fact that the architecture has been used in multiple applications including the virtual tutor of [17].

As soon as a system is embodied in the form of a conversational agent that users can interact with through natural language, the interaction gets the flavour of a social encounter. Language is not only a code system to send information or requests for information; it becomes a social code, with all its social and emotional connotations. To do justice to this aspect of the interaction, several approaches are possible: for example enabling the agent to perform small talk [10] or domain-oriented conversation instead of task-specific dialogue [4], and developing ‘relational agents’ that

are specifically geared toward building a long-term relationship with their users [5]. In our system, we addressed the issue by implementing a politeness model, enabling the Virtual Guide to determine the level of politeness from the user’s verbal behaviour and adapt its linguistic style accordingly.

The paper is structured as follows. First we give a high-level overview of the system architecture (Sect. 2). After that we discuss how the user’s multimodal input is processed (Sect. 3), how dialogue management is carried out (Sect. 4), and how output is generated, focusing on multimodal route descriptions (Sect. 5). Then we discuss how the Virtual Guide adapts its linguistic style to that of the user (Sect. 6). Finally we present some initial user tests (Sect. 7), ending with a conclusion and pointers to future work (Sect. 8).

2 Architecture

Figure 2 shows the architecture of the system. The multi-agent system is built using our distributed agent platform implemented in Java. Agents, specialists in certain tasks, communicate by means of one or more Facilitators (not shown in Fig. 2) that receive and send typed messages to those agents that subscribed to this type of messages. The Facilitator agents resemble those of the Open Agent Architecture [13]. The global architecture is similar to that of other state of the art multimodal agent based dialogue systems such as the WITAS system [29], where a helicopter robot is instructed through natural dialogue using a map, or the COMIC system, where input modalities are speech and pen and output is embodied by an avatar that helps the user navigate through a bathroom design protocol [7].

User input comes from speech, text or mouse clicks in the 2D map. Recorded speech goes through a speech detector, which either works in manual mode (the user clicks

Fig. 2 Architecture

a button when he starts and stops speaking) or in an automatic mode based on the energy level of the audio signal. Speech segments are sent to a speech recognizer, which is grammar-based in order to obtain high-quality recognition results in real time. The speech recognizer produces the first best hypothesis as a string of words, similar to text input, except that each spoken word has a start and end time.

The string of words from speech or text input, is sent to a natural language parser agent and parse trees—actually feature structures, see Sect. 3—are sent to the fusion agent (implementing a form of multimodal semantic fusion; see [43]), which also receives mouse clicks from the 2D map. Linguistic (text or speech) input is the primary modality, as the fusion agent only produces output after receiving a parse tree. The fusion agent annotates the parse tree with information from the mouse clicks, and it tries to find objects in the world to match with expressions in the utterance. The output of the NLP agent goes to the dialogue act determiner, which tries to determine the possible types of dialogue acts in terms of the DAMSL scheme [1]. In the next step, the reference resolver binds referring expressions to objects in the world. Finally, the alignment manager determines the level of politeness of the user's utterance.

Now the dialogue manager decides what action to perform. An action generates a system dialogue act containing text output. The style of the text depends on the current state of system politeness, which in turn depends on the politeness of the user as detected during input processing. The text is realized through speech synthesis, to which the gestures of the 3D avatar are synchronized. An action may also paint routes and highlight locations in the 2D map.

3 Input processing

3.1 Natural language parsing

The NL parser encapsulates a bottom-up left-corner predictive chart parser for unification grammars [35]. The parser loads a Dutch unification grammar, and uses an application specific lexicon, containing about 2500 lexical feature structures. It builds a chart containing all possible complete and incomplete unification structures, encoding the (partial) dependency trees as well as semantic features. The parser handles complete sentences as well as fragments such as noun phrases, but it is sensitive to spelling and grammar mistakes in the input. For speech input this problem is reduced, since the speech recognizer is grammar-based and therefore mainly produces syntactically correct input. On the other hand the speech recognizer is less capable of handling disfluencies in the speech input.

Throughout the system, parse trees are compared with *parse templates* by feature structure unification to check if

```
<s_parse id="1" surface="YNQ">
  <subj>
    <np_parse id="2">
      <head>
        <pn_parse id="3" person="2" />
      </head>
    </np_parse>
  </subj>
  <aux_verbs>
    <verb>can could</verb>
  </aux_verbs>
</s_parse>
```

Fig. 3 Parse template

certain properties hold, such as whether a noun phrase may be a referring expression, or whether an utterance may be a question. A parse template has a similar structure as a parse tree, but only specifies those elements that must be present in the parse tree. For individual words, a template may contain a simple list of allowed words or word types. These functions are an easy yet powerful way to cover a variety of utterance structures. Unique IDs for all elements of a parse template enable easy references to subparses.

An example is shown in Fig. 3. This template matches sentences with the surface form of a yes/no question (YNQ). The sentence must have a noun phrase subject whose head must be a second-person pronoun (referred to as ID 3 in the template), and the sentence must contain the auxiliary verb “can” or “could”. It matches for example “Could you help me?” This template is used in the dialogue act determiner (Sect. 4.1) to detect requests. ID 3 could be used to extract the actual parse of the second-person pronoun from a matching parse tree. In addition, the alignment manager uses the template to determine the politeness level of the request, which in this case will be fairly high due to the presence of the auxiliary verb softening the request (see Sect. 6).

3.2 Multimodal fusion

The fusion agent has two tasks. The first is to merge a parse tree with pointing gestures, in the case of deictic expressions. These are referring expressions that need to be resolved with respect to the situational context; for example, utterances such as “that door” (while pointing at a door). We use the pointing action in this situation to determine which door is meant, since the demonstrative determiner “that” indicates that the door is somehow contextually given. A pointing gesture is represented as a cylinder with an origin and direction. (Mouse clicks in the 2D map result in a cylinder from the ceiling straight down.) We make the assumption that a pointing gesture is only meaningful when it accompanies a deictic expression, and when we find a referring expression that is accompanied by a pointing gesture, we assume that it is a deictic expression.

The fusion agent uses parse templates to find all referring expressions in the user's verbal input, and for each of them, it tries to find a pointing gesture that occurs more or less 'at the same time', i.e., within a time span of 500 ms before the demonstrative word starts and 100 ms after the demonstrative word ends. If more than one pointing gesture meets that constraint, the preferred pointing gesture ends before the demonstrative word ends. If still more than one gesture is left, the gesture whose end is closest to the end of the demonstrative word is chosen. If the parse tree did not come from speech input, then the words do not have start and end times, and we just bind pointing gestures to referring expressions in the order in which they occur.

The second task of the fusion agent is to find a set of objects in the world for all noun phrases found in the user input. For the non-deictic noun phrases, the system just queries the world using the parse tree of the noun phrase. There are three possible kinds of results:

- A set of objects, for noun phrases with a content word, where the world found one or more matching objects.
- An empty set, for noun phrases with a content word, where the world did not find any matching objects.
- Nothing or null, for noun phrases without a content word (e.g. first and second-person pronouns).

For each deictic expression we saw that there is a parse tree of a linguistic fragment and one pointing gesture. First we define the set of objects of those two elements separately. If the linguistic fragment is a noun phrase, its set of objects is defined as above. For other linguistic fragments the set is null. For each pointing gesture, the (possibly empty) set of objects consists of those objects that are in the range of the cylinder. Note that pointing at an empty space, just to indicate a location, is currently not supported by our system.

The 'linguistic set' and the 'gesture set' are merged to obtain the final set of objects. If both sets are not null and not empty, we take their intersection. If the intersection is empty, we take the gesture set, so gesture input has priority over linguistic input. The motivation for this is that the user might not know the correct name for the object he pointed at, or the speech recognizer might have misunderstood the user. If only one of the object sets is not null and not empty, we take that set. Otherwise both sets are null or empty, and we assign an empty set to the deictic expression.

In summary, we note that only noun phrases with a content word, and deictic expressions have a set of objects. For all other parses in the parse tree, the object set is null. The meaning of these object sets will become clear when we look at reference resolution in the next section.

3.3 Reference resolution

The task of the reference resolver is to find the referents of all referring expressions, by looking at the dialogue history.

Since there may be multiple candidates of equal standing, it is not always clear what the referent is. Therefore we introduce the concept of *referent sets*: what the reference resolver assigns to a referring expression is not a referent, but a referent set that contains zero or more objects in the world. The reference resolver takes as input the initial object set that was assigned to a referring expression by the fusion agent and an up-to-date dialogue model.

If the fusion agent already assigned a set of one or more objects to the referring expression, as described in Sect. 3.2, the reference resolver looks in the dialogue history and tries to find a referent set that is a subset of the initial object set, so the initial object set may be restricted to a smaller set. From the dialogue model we can retrieve all qualifying referent sets. If there are any, then we take the most salient one. If there are none, then the referent set is the same as the initial object set. For expressions to which the fusion agent did not assign an initial object set, such as demonstrative pronouns without a pointing gesture, we just take the most salient referent set from the entire dialogue model.

The dialogue model is essentially a list of referring expressions and their referent sets, ordered according to dialogue turns. Each referring expression in the model has a set of salience factors with associated values. Our method to determine salience is based on the pronoun resolution algorithm by Lappin and Leass [28], which assigns weights to referring expressions using salience factors such as recency, presence of pointing gestures and grammatical role. Note that in order to compute recency values, every turn in the dialogue needs to be included in the referent set model, even if it does not contain a referring expression for this set. Every time a new turn occurs, the contribution of old references to the salience value is cut in half.

Figure 4 shows an example dialogue model where the referent set contains only one object: the toilet. Here, the referring expression in the first user utterance (U1) has

Fig. 4 Dialogue model

salience value 100 (recency) + 80 (subject) + 50 (non-adverbial) + 80 (head noun) = 310. This is the salience value of the referent set after U1. If there had been multiple references to the toilet in U1, they would have shared their set of salience factors; in other words, each assigned salience factor counts only once per turn. Now we continue with the system utterance (S1). Since a new turn occurred, the contribution of the previous turns is cut in half, so the contribution of U1 becomes $310/2 = 155$. Since direct objects are less salient than subjects, S1 has value 300, making the total 455. In the next turn (U2) the value decreases to 180, because the phrase “there” is not in a very salient grammatical position. This means the salience value after U2 is $455/2 + 180 = 407.5$. In S2 there is no reference to the toilet at all, so the final salience value after S2 is just half the previous value, 203.75, a more steep decrease.

The referent set in our example includes only one object. However, the reference set assigned to a referring expression may also include multiple referents. In that case, the system will ask the user for clarification to find out which referent is meant. An example from our user tests, which are discussed in Sect. 7, is the following:²

U: I’m looking for the stairs.

S: Could you be more precise? There are more than one, and I don’t know which of them you mean.

4 Dialogue management

The main tasks of the dialogue manager are the interpretation of the user input in terms of a dialogue act type, a task performed by the dialogue act determiner, and the selection of the system actions in response to the recognized user acts.

4.1 Dialogue act determiner

In order for the Virtual Guide to respond appropriately to the user utterance the system tries to identify the type of dialogue act performed by the user. The set of possible dialogue act types and the lexical surface forms that can be used to express these acts is specified by means of dialogue act templates. The modular specification is easy to extend with new templates and we consider this a manageable way compared to the statistical approach to dialogue recognition as in [6] or [24]. The current selection of dialogue act types and templates is motivated by task analysis and experience with the system. Dialogue act types have a backward or forward looking tag as in the DAMSL (Dialog Act Markup in Several Layers) scheme [1]. The possible sequences of dialogue moves are specified by preferred pairs of forward and

backward looking tags. This is based on the idea that interactions are organized by adjacency pairs (a forward looking question or proposal is followed by a backward looking answer, or accept act). A dialogue act template has two parts: a parse list and a list of possible corresponding dialogue act types. The parse list contains parse templates (see Sect. 3.1). The dialogue act determiner tries to unify the parse result received from the fusion agent with the parse templates in the parse list and when a match is found the list of corresponding dialogue acts is returned.

For every forward tag, the dialogue act determiner holds an ordered list of preferred backward tags that can follow it. For example after a WHQ forward tag (a question), the preferred backward tag is ANSWER. A special backward tag is NULL, which is used for the first utterance in the dialogue. The preference order of backward tags influences the way the dialogue manager determines the dialogue structure: a user dialogue act can be a continuation of the current subdialogue as well as a continuation of the enclosing dialogue. If the user could end the current subdialogue—that is if the last dialogue act in the enclosing dialogue was performed by the system and the user started the current subdialogue—the dialogue act determiner will always try to end the current subdialogue by connecting the user’s dialogue act to the enclosing dialogue. So it first considers the forward tag of the last dialogue act in the enclosing dialogue. Only if none of the preferred backward tags for this forward tag is available, the forward tag of the last dialogue act in the current subdialogue is considered. If still none of the preferred backward tags is available, the dialogue act determiner returns the first dialogue act in the input list.

4.2 Action selection

The Dialogue Manager uses action templates to map a user dialogue act to one or more system actions. An action template contains a parse template combined with a forward or backward tag and a system action. When a user utterance matches the parse template, the corresponding action is put on the Action Stack. Then the dialogue manager executes actions on the stack, until the stack is empty or the action on top is not executable (it may need extra user input for example). If a user utterance contains multiple dialogue acts, then they are processed in sequence. For each dialogue act an action is created and put on the stack, and the actions are executed (i.e., realised using text and speech and possibly other modalities), before the next dialogue act is processed.

There is a strict logical order in the execution of the updates of dialogue information, the selection of goals and the execution of actions after the system has received user input. This sequential approach has some advantages and some disadvantages if we compare it with more integral approaches such as in [12] and [26]. In the former approach

²All dialogue excerpts have been translated from Dutch to English.

the speech recognition, dialogue manager and speech production each have their own asynchronous processing. In the latter paper the authors mention and discuss the functionalities of a multi-modal dialogue and action management module. They have chosen a stack of ATN's, but because during the dialogue the user can return to a previously closed dialogue topic it should be possible to return in the history and a previously popped ATN should remain accessible.

Our way of dialogue and action management allows mixed-initiative dialogues—one of the basic minimal functionalities required [12]—and several types of subdialogues. Either the system or the user can take the initiative by asking for clarification instead of directly answering a question. An action stack stores the system's actions that are planned and a subdialogue stack (the stack of 'questions under discussion') keeps track of the current dialogue structure. Besides, all dialogue acts are kept in a history list of dialogue acts, so they can be retrieved for later use.

Although dialogue management is basically implemented with alternating user and system turns, the system may decide to perform two or more dialogue acts or none at all in its actions. In the latter case it is perceived as the user being allowed to perform more than one dialogue act in one turn. The current system assumes that the user only reacts to the last dialogue act performed in a system turn, which need not be its most recent turn. To enable interruptions, the system returns the turn to the user as soon as it has planned a dialogue act and updates the dialogue state as if the act has already been performed.

4.3 Example

As an illustration of the process performed by the dialogue manager, consider the user utterance "Where is the great hall". This results in the following structure:

```
S [WHQ] -> ADV V NP -> where is the great
hall [great_hall]
```

Here WHQ refers to the surface form of the parse, which often, as in this case, corresponds to the forward tag of the dialogue act, but that is not necessarily so. The possible dialogue acts assigned to this example will have backward tags NULL, ACK and HOLD. The set of possible tags assigned to an utterance is motivated by the possible dialogue acts performed by the speaker when he uses this utterance. The assignment of *great_hall* to the noun phrase indicates that the noun phrase is bound to the 'great hall' object in the virtual world. In fact, there is an action template TELL_LOCATION that matches all 'where is' questions and takes one argument: the object bound to the noun phrase.

In the previous example the assigned dialogue act is not actually discerning as to the selection of the system action. This is more the case when the user utterance is a partial

sentence, such as a simple noun phrase. The utterance "(and) the great hall" can be interpreted differently depending on the dialogue context. After a wh-question, it is likely to be an answer to the question. After a statement, it is likely to be an elliptical question. The example results in the following parse:

```
S [NP] -> NP -> the great hall [great_hall]
```

However the assigned possible dialogue acts are [STATEMENT, ANSWER] and [WHQ, ACK]. The dialogue act determiner has a preference for the former act if the last system dialogue act had forward tag WHQ, and for the latter if the forward tag was STATEMENT.

5 Output generation

As seen in Sect. 4, in response to the user's input the dialogue manager puts a number of actions on the system action stack. System actions are specified for each type of action to be performed by the system: there is for example a GREETACTION, and also an action called ACTIONTELLPATHTO. Most of the actions have parameters that are filled during reference resolution. As soon as the system obtains the turn, the actions on the stack are executed when possible.

The system turn realiser (see Fig. 2) is an agent that uses sentence templates and a politeness model for generating the appropriate natural language output, adapting the system's level of politeness to that of the user. Whereas a system such as POLLY [19, 20] uses a general purpose linguistic realiser for generating utterances at different levels of politeness, no such realisers are available for Dutch. Therefore we follow [39] in using templates with politeness tags, which has the advantage of allowing us full control of the system's linguistic style. How our politeness model works is described in Sect. 6; here we focus on the generation of route descriptions by the Virtual Guide.

When the dialogue manager sends action ACTIONTELLPATHTO to the output generation module, this results in a multimodal route description: the route is projected on the map of the music centre (see Fig. 1) and presented by the Guide using speech and gestures. Below we briefly describe how the route description text is generated, how it is augmented with gestures, and how the result is presented using speech synthesis and animation. See [36] for more details.

5.1 Language generation

The input for the route description consists of the shortest path from the starting point to the destination, specified as a list of *markers*: 3D coordinates in the virtual music centre. The path is computed based on a network of predefined paths in the virtual environment. Two connected markers

form a segment, and the first step of the language generation algorithm is to calculate the angle between each pair of subsequent segments. Based on these angles, a turn direction is determined for each marker (straight ahead, sharp left, left, etc.) and added to the path. Multiple subsequent markers associated with the direction ‘straight ahead’ are filtered out.

The next step is to describe the locations where the turns are to be made in terms of landmarks, i.e., salient objects or other reference points. In buildings, typical landmarks include stairs, hallways and signs. For the selection of potential landmarks, a cylinder-shaped area originating from a marker is used to capture the objects that are near the location of the marker. From the objects located within the cylinder, the generator picks the one best suited for use as a landmark. This is done using a variation on Dale and Reiter’s [15] Incremental Algorithm for the generation of referring expressions, which reduces the set of potential landmarks to one based on their values for properties such as size, movability (immovable objects are more reliable landmarks than movable objects), colour and shape. The algorithm goes through this list of properties one by one, each time reducing the set of potential landmarks to those objects that (a) have a preferred value for the current property or (b) have an optimally distinguishing value for that property, i.e., a value that distinguishes them from the highest number of alternative landmarks. If going through the properties does not reduce the set to one landmark, the algorithm simply chooses the landmark that is closest to the turn location.

In our model, landmark selection is only based on the absolute proximity and salience of potential landmarks. A more sophisticated model has been proposed by Kelleher and Costello [25]. In their approach, landmark selection is based on *relative* proximity, which is influenced by the presence of other objects. They argue that this is important for dialogue systems that operate in a real world setting, since visual scenes in the real world are complex and contain multiple objects. However, our Virtual Guide operates in a virtual environment with relatively few objects, for which our simpler approach seems sufficient.

The final path specification is a sequence of markers associated with turn directions and landmarks. From this input, a route description is generated using a collection of sentence templates based on the Exemplars system [42].³ The templates are organized in a specialization hierarchy, where specialized templates can augment or override the more general ones. At each level of the hierarchy a number of equivalent templates is available, from which a random choice is made. For example, *At <landmark>, go <direction>* and *Turn <direction> at <landmark>*. After a first version of the route description is generated using simple sentence structures like the examples above, the text is revised

by combining some sentences and adding cue phrases such as *then* and *after that*, making the output text more fluent.

5.2 Gesture generation

To generate appropriate gestures to accompany the verbal route description, the words in the route description are associated with tags representing different types of gestures. The marked-up text is sent to the animation planner, which generates the required animations in synchronization with text-to-speech output.

Gesture selection in our system follows a ‘suggest-and-reduce’ approach that is somewhat similar to that of the BEAT system [11]. First, the system creates a collection of all possible gestures that could be used to accompany the landmark references and direction words in each sentence of the route description. For example, references to landmarks can be accompanied by (1) a pointing gesture to the absolute location of the object (‘objective viewpoint’), (2) a pointing gesture to the location of the object relative to the position of a person who is walking the route (‘subjective viewpoint’) or (3) an iconic gesture, reflecting the shape of the object. After the full route description has been generated, a selection from all possible gestures is made, based on weighted randomization. The weights are currently determined by hand; more realistic weights might be determined empirically based on the results of video analysis [30]. Currently the system always selects pointing gestures made from an objective viewpoint. This choice is based on the results of an experiment where 32 participants judged three movies in which the Virtual Guide gave the same route description, each time with a different type of gestures. Of the participants, 68% preferred the movie with the objective viewpoint gestures.

Finally, the gestures are animated and synchronized with the speech output using a modified version of the animation planner developed by [41]. The text of the route description is sent to a speech synthesizer which not only pronounces the text but also returns an estimation of the durations of the phonemes in the utterance. This information is used to synchronize the gesture animations with the words they are associated with.

The pointing gestures that the Virtual Guide makes from an objective viewpoint are generated dynamically, using the location of the target object as input parameter. The other gestures, however, are generated using canned animations. An example is a horizontal, tube-like iconic gesture that can be used in references to corridors and tunnels. Given the limited number of potential landmarks in the Virtual Music Centre, this simple approach is the most efficient choice. A more flexible approach would be to automatically create iconic gestures that appropriately reflect landmark shape. This would require a gesture generation model such as that

³ Similar templates are also used for generating system utterances other than route descriptions; see Sect. 6.3.

of Kopp et al., which links visual object properties to gesture features such as hand shape and trajectory [27]. Their model has been applied in NUMACK, an embodied conversational agent that functions as a virtual guide for the Northwestern University campus.

6 Alignment and politeness

Evidence from psycholinguistics has shown that the linguistic representations used by dialogue partners automatically become aligned at many levels [32]. In other words, dialogue partners tend to copy (parts of) each other's language. Previous work on implementing alignment in dialogue systems has focused on the syntactic and lexical levels. In their implementation of syntactic alignment in the CRAG-2 system, Isard et al. [21] use n-gram language models enabling two dialogue agents to mirror the syntactic structure of each other's utterances. Buschmeier et al. [9] present a language generation system in which linguistic structures are activated based on recency and frequency of use by either the system (self-alignment) or its human interlocutor. The help desk system developed by Janarthanam and Lemon [22] can automatically adapt its choice of referring expressions to that of the user, reflecting the user's lexical knowledge.

Bateman and Paris [3] suggest extending the alignment model of Pickering and Garrod [32] to affective alignment, i.e., having the system adopt the same affective style (or 'register') as the user rather than directly copying his or her language. Following up on Pickering and Garrod's claim that production and interpretation of language are closely linked, Bateman and Paris stress the importance of interpretation: in order to achieve affective alignment, the system must be able to recognize stylistic variations in the utterances of the user, thus creating a closed affective loop. We have followed their suggestion for the Virtual Guide, focusing on politeness. Implementing politeness models in virtual dialogue agents may help to give them a believable personality [38], make them appear more socially intelligent [2] and yield better learning outcomes for pedagogical agents [39].

By making the Virtual Guide capable of affective alignment, we intend to achieve a human-like style of interacting with the user. When in 'alignment mode', politely asking the Guide a question will result in a polite answer, while a rudely phrased question will result in a less polite answer. Below, we describe how the level of politeness of the user's utterances is determined, and how appropriate responses are generated. First, we briefly review the relevant literature on politeness and linguistic style.

6.1 Politeness and linguistic style

Like most previous work, we took our inspiration from Brown and Levinson's famous politeness model [8], which

is based on the idea that speakers are polite in order to save the hearer's face: a public self-image that every person wants to pursue. The concept of face is divided in *positive face*, the need for a person to be approved of by others, and *negative face*, the need for autonomy from others. Whenever a speech act goes against either of these needs, this is called a Face Threatening Act (FTA). Brown and Levinson discuss various linguistic strategies to express an FTA at different levels of politeness. For example, when using the *bald on-record strategy*, the FTA is phrased in direct terms without accounting for face threat, e.g., by using an imperative. This strategy is used when there is no time or need to care about the hearer's face, for example in emergency situations ("Help me!"). It can also be safely used for speech acts that hardly pose a face threat, for example straightforward Informs such as "The car is in the parking lot". The *off-record strategy* is an indirect way of phrasing an FTA so that it allows for a non-face threatening interpretation. For instance, when someone says "This weather always makes me thirsty" this is probably a hint that he would like a drink. However, for the hearer it is easy to ignore the indirect request and treat the utterance merely as an informing act instead.

Presumably the first attempt at implementing such politeness strategies in virtual agents was made by Walker et al. [38], with a recent follow-up in the POLLY system [19, 20]. In their approach, the desired level of politeness of an utterance depends on the social distance between the dialogue participants, the power one has over the other, and the estimated face threat posed by the speech act. Other related work includes [2, 33, 39]. These all generate tutoring responses based on Brown and Levinson's theory.

All of the above-mentioned systems perform generation based on static input parameters indicating the desired level of politeness. As future work, Walker et al. [38] mention exploring a 'reciprocal feedback loop' where the relevant parameters are not set in advance but change dynamically over the course of the dialogue, leading to interesting changes in the way the conversational partners interact with each other. Achieving such a system is exactly what we set out to do by adding style alignment to the Virtual Guide.

6.2 Input analysis for alignment

It is always the user who initiates the dialogue with the Virtual Guide, giving it the opportunity to scan the user's input before deciding what linguistic style it should use. Our approach to analysing the level of politeness of the user's utterances is similar to that of [18], who apply emotional or attitudinal tags to grammar rules to extract affective information from user utterances. For example, the use of words such as "please" increases the politeness of an utterance.

Data on politeness and linguistic style in Dutch are available from the work of Vismans [37]. To investigate the influence of sentence structure on politeness, Vismans asked 24

Table 1 Some sentence structures that can be recognized by the grammar. Politeness values (P) are based on the ratings from [37], converted to a scale from −5 (very impolite) to +5 (very polite)

Form	Example sentences	P
IMP	Toon (me) de zaal.* (Show me the hall.)	−3
DECL	Je moet me vertellen waar de zaal is.* (You have to tell me where the hall is.)	−2
	Ik moet/wil naar de zaal. (I have to/want to go to the hall.)	−1
	Ik zoek de zaal. (I am looking for the hall.)	0
INT	Waar is de zaal? (Where is the hall?)	0
	Waar/hoe vind ik de zaal? (Where/how do I find the hall?)	0
	Waar/hoe kan ik de zaal vinden? (Where/how can I find the hall?)	1
	Wil je de zaal tonen?*	2
	(Do you want to show me the hall?)	
	Zou je de zaal willen/kunnen tonen?*	3
	(Would/could you show me the hall?)	
	Weet je waar de zaal is? (Do you know where the hall is?)	3

native speakers of Dutch to rate the politeness of 9 variations of a request to close the door. The subjects rated imperative sentences (IMP) as least polite, and interrogative sentences (INT) as most polite. Requests phrased as a declarative sentence (DECL; e.g., “You should. . .”) were rated in between. Based on Vismans’ results we associated the parse templates produced by our unification grammar (see Sect. 3.1) with politeness values on a scale from −5 (least polite) to +5 (most polite). These politeness values are used by the alignment manager to update the system’s level of politeness, as described below.

Table 1 shows some variations of the request “Show me the hall” (Toon me de zaal) that can be handled by our grammar, together with their associated politeness value P.⁴ The sentence structures marked with * were among those originally investigated by Vismans (not all 9 structures he investigated are shown in the table). The P value of those sentence structures is based on the ratings from Vismans’ experiment, converted to the scale used in our model.

As shown in Table 1, our grammar can recognize more sentence structures than those tested in Vismans’ experiments. For the additional sentence structures we estimated a politeness value based on the use of forceful verbs such as “must” or mitigating verbs such as “could”. Declarative sentences such as “I’m looking for the hall” might be seen as examples of Brown and Levinson’s *off-record* strategy.

⁴The politeness value of the English translations provided in this paper may slightly differ from the Dutch originals.

However, we felt that such utterances do create an expectation for the addressee to act that is hard to ignore, so we assigned them a neutral rather than a positive politeness value. Declaratives using forceful phrases such as “I want. . .” and “I must. . .” are ranked as slightly impolite. Dialogue act types not illustrated in Table 1 include opening and closing acts (greetings/farewells) (+2) and thanking acts (+3).

Besides sentence structure, the level of politeness of an utterance is also influenced by the use of modal particles such as “perhaps” or “possibly”, as in “Could you perhaps show me the hall?” Vismans separated Dutch modal particles in *reinforcers* (dan, nou, ook, toch, eens) and *mitigators* (even, maar, misschien, soms) [37]. He experimentally showed that reinforcers apply more pressure to the hearer of the speech act, while mitigators do the opposite. However, the stronger the force of the FTA, the weaker the added effect of reinforcers or mitigators.

In our model, the politeness value of a user utterance (UP, for User Politeness) is calculated by adding the effect of modal particles (MP-Effect) to the politeness level of sentence structure (P):

$$UP = P + \text{MP-Effect} \quad (1)$$

The effect of modal particles in combination with sentence structure is approximated by the following formula:

$$\text{MP-Effect} = (5 - |P|)/5 * \text{MP} \quad (2)$$

Here, MP is the basic politeness value of the modal particle, which we set at +1 for mitigators and −1 for reinforcers. The formula for computing MP-effect ensures that the effect of mitigators and reinforcing particles is strong for sentences that have a relatively neutral structure (P close to 0) but weak for utterances that are already at the extreme ends of the politeness scale, based on their sentence structure (P close to −5 or +5). Besides the presence of modal particles, the system also checks the user’s utterance for formal or informal wording (e.g., “lavatory” versus “toilet”) using a so-called ‘shaded lexicon’. This aspect of the alignment model is not discussed here.

Finally, the system checks for the use of formal or informal ways of addressing. In Dutch, like other languages such as French and German, dialogue partners can use either formal or informal personal pronouns to address each other. We call this *T-V distinction*, after the Latin informal and formal personal pronouns “tu” and “vos”. The T-V distinction also affects other words and phrases that incorporate personal pronouns, as illustrated in Table 2.

In our model, T-V distinction is represented by a value of either 1 or 0. If formal versions of the phrases from Table 2 are detected in the user input, this value is set to 1, otherwise to 0. Our main reason for keeping the T-V value separate from politeness, even though it is clearly related to it, is a

Table 2 T-V distinction in Dutch

Formal	Informal	Translation
u	je/jij	you
uw	je/jouw	your
dankuwel	dankjewel	thank you
alstublieft	alsjeblieft	please

practical one: if the system's choice of pronouns (Sect. 6.3) would depend on the user's current level of politeness, this might cause our Virtual Guide to switch between the use of formal or informal pronouns without the user ever having made a change in T-V distinction at all.

6.3 Generating aligned utterances

After having analysed the user's utterance for politeness and pronoun use, the system has to generate a response at the appropriate level of politeness. The generation process takes as input (1) a system act, selected by the dialogue manager (see Sect. 4), and (2) the system's current level of politeness and T-V value. As the dialogue advances, the Guide adapts its level of politeness according to the following formula, where the degree of alignment is set by the variable α :

$$SP(i+1) = \alpha * SP(i) + (1 - \alpha) * UP(i+1) \quad (3)$$

Here, $SP(i)$ is the system's current level of politeness, and $UP(i+1)$ is the politeness value of the user's last utterance. The value of α varies between 0 and 1 and determines how changeable the system's politeness level is. The closer α gets to 1, the slower the Guide will adapt its language to that of the user. Changing the value of α allows us to experiment with different alignment settings, varying between no alignment at all ($\alpha = 1$) and full alignment ($\alpha = 0$).

Based on its current politeness level, the system selects a surface realisation for the dialogue act to be carried out. First, an appropriate politeness tactic, in the form of a sentence template, is selected. Certain slots in this template are filled with more or less formal words, as described in [23]. Personal pronouns and other relevant words (see Table 2) are filled in according to the current T-V value.

Currently the Guide has 21 different politeness tactics at its disposal, including those from Table 1; for a full list see [23]. For generation we do not attach a specific politeness value to each tactic, but instead we group the tactics in clusters with an associated politeness range, e.g., from +4 to +5. The ranges were determined partly based on the politeness ratings from [37] and partly on intuition, see Sect. 6.2. During generation, the Virtual Guide randomly selects a tactic from the appropriate range given its current level of politeness. This way, an appropriate tactic is guaranteed to be found (unlike when an exact match is required)

and some variation is achieved even when politeness stays at the same level during the dialogue. To further increase variation, some Inform acts—which make up a large part of the system actions—are formulated as Requests. For example, the act of informing the user that the Guide has marked a location on the map can be phrased as a request to look at the map.

Although achieving lexical and syntactic alignment is not the main aim of our system, it sometimes occurs as a side-effect. When the Virtual Guide reacts to the user at the same level of politeness, it is likely to use the same linguistic structure as the user, since analysis and generation are based on the same politeness tactics. Similarly, at maximum alignment the Guide may use the same words as the user, because the system uses the same lexicon for both formality analysis and generation. For example, when the user formally greets the Guide with “Good evening!”, the Guide's response has a high chance of being “Good evening” too.

6.4 Evaluation

To evaluate our politeness model, we let 25 speakers of Dutch rate 21 system utterances, each embodying a different politeness tactic, on their level of politeness. Informal pronouns were used in all sentences; the influence of T-V distinction was not tested in this experiment. The results revealed that the average user ratings of the politeness tactics did not always fall within the range assigned to them by our model. The largest deviations occurred with indirect tactics (e.g., “Someone should try again” and “The question is if you want to phrase it differently”). Those were rated as much less polite than predicted. Similar results were found in a cross-cultural evaluation of politeness tactics by [19]: they found that indirect tactics, which in theory should have been the most polite, were rated most impolite of all.

Large deviations from the model were also found for the tactics *Ability* (“Is it possible for you to look at the map where I've marked the hall?”) and *Mind* (“Would you mind looking at the map where I've marked the hall?”). A frequent comment on the latter tactic was that subjects found the phrase “If you don't mind” out of place in the context of the request to look at the map. They said “Why would I mind?”, indicating the absence of any threat to autonomy. Finally, although the *Inclusive* tactic (“Shall we try again?”) was found patronizing by some participants, its average rating was much more polite than predicted.

Given that we used only one instance of each politeness tactic in our evaluation, we cannot draw strong conclusions from the results; using the tactics to express other speech acts might have resulted in different ratings. Nevertheless, based on the results we made some preliminary adjustments to the politeness model; see [23] for more details.

7 Interactive user tests

As a first attempt to test our system in action with real users, we asked 5 people to carry out a number of unsupervised, typed dialogues with the Virtual Guide. The users were 3 male and 2 female Computer Science masters students, who had no previous experience with or knowledge of the Virtual Guide. We asked them to have one dialogue with a non-aligning version of the Guide and two dialogues with the Guide at full alignment ($\alpha = 0$, see Sect. 6.3). For dialogue two (the first aligned dialogue), we asked them to be polite to the Guide, and for dialogue three (the second aligned dialogue) we asked them to be impolite. The users were free to determine the content of the dialogue—while staying within the direction giving domain—and were asked to report on their experiences afterwards. Also, logs of their dialogues with the Guide were collected.

Due to a system malfunction, one user could only carry out the first, non-aligned dialogue with the Virtual Guide. The other four reported that they clearly noticed the effect of alignment in the second and third dialogues. Most of them said they liked the Guide's linguistic style adaptation in the polite second dialogue, but they found it less appropriate in the impolite third dialogue. One reason they gave for this was the nature of the application: it is the Guide's job to provide a service to the user, and the participants felt that in this role the Guide should always be polite, even to an impolite 'customer'. Nevertheless, they did think it was fun to see how the Guide adapted its language to theirs, resulting in exchanges such as the following:

U: Then how do I get there from the exposition, pal.

S: I didn't understand what you said, mate.

This suggests that our alignment model as it is now would certainly be a useful addition in applications where user and system have a fairly equal status, e.g., in chat agents. However, to make the model more general, determining the system's level of politeness should not only be a matter of alignment to the user's linguistic style, but also take other variables into account. In Brown and Levinson's theory, the weight of a face threat (and thus the need for politeness) depends on social distance, relative power, and imposition ranking of the speech act [8]. Currently our model does not take these factors into account, unlike other work on politeness in dialogue systems [2, 19, 38, 39].

Besides the status issue, another reason why the users found some of the Guide's impolite utterances somewhat inappropriate was the Guide's limited understanding. Many of the users' utterances could not be interpreted by the system. As a result, the Guide often had to ask them to rephrase. A typical dialogue excerpt illustrating this is the following.

U: Tell me how to go to exposition from toilet.

S: Unfortunately I did not understand what you said. Used words I don't understand. Try again?

U: From toilet to exposition.

S: I heard "from toilet to exposition", but I don't know what to do with that. You will have to say it differently.

In another dialogue, an exasperated user said "Help!" after a long sequence of misunderstandings. Unfortunately, the Guide could not handle this request, but classified it as an impolite phrasing due to its syntactic (imperative) structure, and made matters worse by reacting with increasingly impolite phrasings:

U: Help!

S: I understand you said "help!", but I can't do anything with that. Say it differently.

U: Help.

S: I understand what you said, but what should I do with that? Say it differently.

U: I need help.

S: Not understood. I don't get it. Try again.

Users noted that since the repeated misunderstandings were the Guide's 'fault', the Guide ought to remain polite no matter what, while the users should not be blamed for being impolite due to frustration. Problems like these could be avoided using the approach proposed by André et al. [2]. Their politeness model keeps track of the user's emotional state and the events that caused it, preventing the use of inappropriate politeness tactics if the user feels frustrated due to a system failure. This approach would also keep the system from being overly polite, which could be interpreted by the users as irony or sarcasm.

Our test runs also revealed other unintentional effects of the Guide's over-use of politeness tactics. For example, using the inclusive "we" (see Sect. 6.4) was perceived as strange in the context of utterances such as "We have used words I cannot understand". Also, the users thought that indirect utterances such as the following, intended to be polite, made the system sound insecure instead:

U: Could you tell me how to get from toilet to keyboard.

S: It looks like I have been able to mark the route from toilet to keyboard on the map.

Concerning the dialogue in general, the test users found it difficult to determine what they could or could not talk about with the Guide. Meta-questions aimed at finding out about this (e.g., "Which locations are there in the music centre?") were mostly not understood. Adding to the confusion was the fact that the Virtual Guide sometimes said that it had understood the user, when it clearly had not; see the "Help" dialogue above for an example. In this and other cases the Guide successfully parsed the user's utterance but did not know how to react to it, because it lacked an appropriate

action template. However, this distinction between identification and understanding (cf. Clark's levels of joint communicative action [14, p. 153]) was not always clear to the user.

The users were also confused by the fact that the system's references to objects were not always consistent. On the other hand, they did appreciate the system's capability of keeping track of the dialogue history and correctly resolving anaphors, as in the following dialogue excerpt:

U: What is this? (while pointing).

S: That is exposition.

U: How do I get there?

S: I have marked the route on the map. You go left at the main entrance. Then (...).

We intend to analyse the dialogue logs and extend the system's parsing and interpretation capabilities accordingly.

8 Conclusions and future work

In this paper we have described how the Virtual Guide analyses the user's multimodal input, how it decides how to react and generate its own multimodal output, and we have presented an alignment model that regulates how the Guide adapts its linguistic style to the user's level of politeness. The Virtual Guide is fully implemented and functional. Some aspects still need to be improved, though. Our first test runs with naive users have exposed several gaps in the grammar used for parsing user utterances and the action templates available to the dialogue manager, and they also revealed some limitations of the alignment model. For example, our politeness model currently does not take the face threatening potential of different speech acts into account. This sometimes leads to the unnecessary use of politeness tactics in non-threatening Inform acts, which in the worst case may create the unintended impression that the Guide is poking fun at the user. Not only the type of speech act should be taken into account, but also its content. For example, some Inform acts may pose a face threat while others do not, which means that certain politeness tactics are appropriate for one Inform act ("It seems I didn't understand you correctly") but not for another ("It seems I have marked the exposition on the map"). Similarly, the use of an imperative should be interpreted as impolite when the user is demanding information from the Guide, but not when he is desperately seeking help. It is clear that a more refined approach is needed here.

Our user tests as described in Sect. 7 have provided us with a first impression of how our alignment model functions during interaction. For a more thorough investigation on how to incorporate politeness in an interactive system, more formal user experiments will have to be carried out, involving larger numbers of users of various ages and backgrounds, and using systematically varied parameter settings.

For instance, varying the Guide's initial level of politeness and the degree of alignment would allow us, in principle, to model different professional attitudes or personalities for the Guide. So far we have only tested the Guide at the highest degree of alignment, where the Guide is strongly and immediately influenced by the user's behaviour. We should also test the effects of lower degrees of alignment, where the Guide only becomes more impolite after repeated 'provocations' by the user. This may be more appropriate for the current application, since the participants in our user test indicated that as a service provider, the Guide should remain relatively polite. One participant suggested that the Guide's level of politeness should always be at least one level higher than that of the user. This is something that could easily be built into our model.

Currently, our politeness model is limited to verbal interaction. However, in human interaction there is also a relation between gesture use and verbal politeness strategies [34]. Incorporating the use of gestures in our model is an interesting topic for future work. Also, we would like to investigate the possibility of having the Virtual Guide walk around and lead the user through the environment while giving directions. Finally, our future work will certainly include extending the grammar and action templates used for input analysis and dialogue management, based on the interaction data gathered in our user tests.

Acknowledgements We thank Martin Bouman and Richard Korthuis for their work on the language generation component of the Virtual Guide, Marco van Kessel for his work on gesture generation and animation, and Markus de Jong for his work on the implementation and evaluation of the alignment model. We also thank the participants in our user tests, and the reviewers of a first version of this paper for their useful comments. This work has been supported in part by the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement 231287 (SSPNet).

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Allen J, Core M (1997) Draft of DAMSL: Dialog Act Markup in Several Layers. Tech. rep., University of Rochester
2. André E, Rehm M, Minker W, Buhler D (2004) Endowing spoken language dialogue systems with emotional intelligence. In: *Affective dialogue systems*. LNCS, vol 3068, pp 178–187
3. Bateman J, Paris C (2005) Adaptation to affective factors: architectural impacts for natural language generation and dialogue. In: *Proceedings of the workshop on adapting the interaction style to affective factors at the 10th international conference on user modeling (UM-05)*
4. Bernsen N, Dybkjær L (2004) Managing domain-oriented spoken conversation. In: *Proceedings of the AAMAS 2004 workshop on embodied conversational agents: balanced perception and action*, pp 9–17

5. Bickmore T, Caruso L, Clough-Gorr K, Heeren T (2005) 'It's just like you talk to a friend'—relational agents for older adults. *Interact Comput* 17(6):711–735
6. Black W, Thompson P, Funk A, Conroy A (2003) Learning to classify utterances in a task-oriented dialogue. In: *Proceedings of the 2003 EACL workshop on dialogue systems: interaction, adaptation and styles of management*, pp 9–16
7. Boves L, Neumann A, Vuurpijl L, ten Bosch L, Rossignol S, Engler R, Pfeleger N (2004) Multimodal interaction in architectural design applications. In: *Proceedings UI4ALL 2004: 8th ERCIM workshop on "user interfaces for all"*, pp 384–390
8. Brown P, Levinson SC (1987) *Politeness—some universals in language usage*. Cambridge University Press, Cambridge
9. Buschmeier H, Bergmann K, Kopp S (2009) An alignment-capable microplanner for natural language generation. In: *Proceedings of the twelfth European workshop on natural language generation (ENLG 2009)*, pp 82–89
10. Cassell J, Bickmore T (2003) Negotiated collusion: modeling social language and its relationship effects in intelligent agents. *User Model User-Adapt Interact* 13(1–2):89–132
11. Cassell J, Vilhjálmsdóttir H, Bickmore T (2001) BEAT: the Behavior Expression Animation Toolkit. In: *Proceedings of SIGGRAPH '01*, pp 477–486
12. Catzone R, Setzer A, Wilks Y (2003) Multimodal dialogue management in the COMIC project. In: *Proceedings of the 2003 EACL workshop on dialogue systems: interaction, adaptation and styles of management*, pp 25–34
13. Cheyer A, Martin D (2001) The open agent architecture. *J Auton Agents Multi-Agent Syst* 4(1):143–148
14. Clark HH (1996) *Using language*. Cambridge University Press, Cambridge
15. Dale R, Reiter E (1995) Computational interpretation of the Gricean maxims in the generation of referring expressions. *Cogn Sci* 19(2):233–263
16. van Dijk B, op den Akker R, Nijholt A, Zwiers J (2003) Navigation assistance in virtual worlds. *Inf Sci* 6:115–125. Special series on community informatics
17. Evers M, Nijholt A (2000) Jacob—an animated instruction agent for virtual reality. In: Tan T et al (eds), *Advances in multimodal interfaces—ICMI 2000*. LNCS, vol 1948. Springer, Berlin, pp 526–533
18. Guinn C, Hubal R (2003) Extracting emotional information from the text of spoken dialog. In: *Proceedings of the 9th international conference on user modeling*, pp 23–27
19. Gupta S, Walker MA, Romano DM (2007) Generating politeness in task based interaction: an evaluation of the effect of linguistic form and culture. In: *Proceedings of the eleventh European workshop on natural language generation (ENLG-07)*, pp 57–64
20. Gupta S, Walker MA, Romano DM (2008) POLLY: a conversational system that uses a shared, representation to generate action and social language. In: *Proceedings of IJCNLP 2008, the third international joint conference on natural language processing*, pp 967–972
21. Isard A, Brockmann C, Oberlander J (2006) Individuality and alignment in generated dialogues. In: *Proceedings of the 4th international conference on natural language generation (INLG-06)*, pp 22–29
22. Janarthanam S, Lemon O (2009) Learning lexical alignment policies for generating referring expressions for spoken dialogue systems. In: *Proceedings of the twelfth European workshop on natural language generation (ENLG 2009)*, pp 74–81
23. de Jong M, Theune M, Hofs D (2008) Politeness and alignment in dialogues with a virtual guide. In: *Proceedings of the seventh international conference on autonomous agents and multiagent systems (AAMAS 2008)*, pp 207–214
24. Keizer S, op den Akker R (2007) Dialogue act recognition under uncertainty using bayesian networks. *Nat Lang Eng* 13(4):287–316
25. Kelleher JD, Costello FJ (2009) Applying computational models of spatial prepositions to visually situated dialog. *Comput Linguist* 35(2):271–306
26. Kerminen A, Jokinen K (2003) Distributed dialogue management in a blackboard architecture. In: *Proceedings of the 2003 EACL workshop on dialogue systems: interaction, adaptation and styles of management*, pp 53–60
27. Kopp S, Tepper P, Striegnitz K, Ferriman K, Cassell J (2007) Trading spaces: how humans and humanoids use speech and gesture to give directions. In: Nishida T (ed) *Engineering approaches to conversational informatics*. Wiley, New York
28. Lappin S, Leass H (1994) An algorithm for pronominal anaphora resolution. *Comput Linguist* 20(4):535–561
29. Lemon O, Bracy A, Gruenstein A, Peters S (2001) The WITAS multi-modal dialogue system I. In: *Proceedings EuroSpeech 2001*, pp 1559–1562
30. Neff M, Kipp M, Albrecht I, Seidel HP (2008) Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Trans Graph* 27(1):1–24
31. Oviatt S, Cohen P (2000) Multimodal interfaces that process what comes naturally. *Commun ACM* 43(3):45–53
32. Pickering MJ, Garrod S (2004) Toward a mechanistic psychology of dialogue. *Behav Brain Sci* 27:169–226
33. Porayska-Pomsta K, Mellish C (2004) Modelling politeness in natural language generation. In: *Proceedings of the third international conference on natural language generation (INLG-04)*. LNAI, vol 3123, pp 141–150
34. Rehm M, André E (2005) Informing the design of embodied conversational agents by analyzing multimodal politeness behaviors in human-human communication. In: *Proceedings of the AISB symposium on conversational informatics for supporting social intelligence and interaction*, pp 144–151
35. Sikkel K, op den Akker R (1993) Predictive head-corner chart parsing. In: *IWPT 3, third international workshop on parsing technologies*, pp 267–276
36. Theune M, Hofs D, van Kessel M (2007) The virtual guide: a direction giving embodied conversational agent. In: *Proceedings of interspeech 2007*, pp 2197–2200
37. Vismans R (1994) Modal particles in dutch directives: a study in functional grammar. In: *IFOTT, Vrije Universiteit, Amsterdam*
38. Walker M, Cahn J, Whittaker S (1997) Improvising linguistic style: social and affective bases for agent personality. In: *Proceedings of autonomous agents'97*. ACM, New York, pp 96–105
39. Wang N, Johnson WL, Mayer RE, Rizzo P, Shaw E, Collins H (2008) The politeness effect: pedagogical agents and learning outcomes. *Int J Human-Comput Stud* 66:98–112
40. Wasinger R, Wahlster W (2006) Multimodal human-environment interaction. In: Aarts E, Encarnação J (eds) *True visions: the emergence of ambient intelligence*. Springer, Berlin, pp 293–308
41. van Welbergen H, Nijholt A, Reidsma D, Zwiers J (2006) Presenting in virtual worlds: towards an architecture for a 3D presenter explaining 2D-presented information. *IEEE Intell Syst* 21(5):47–53
42. White M, Caldwell T (1998) EXEMPLARS: a practical, extensible framework for dynamic text generation. In: *Proceedings of the ninth international workshop on natural language generation (INLG-98)*, pp 266–275
43. Wu L, Oviatt SL, Cohen PR (1999) Multimodal integration—a statistical view. *IEEE Trans Multimedia* 1(4):334–341