# A multi-speaker dialogue system for computer-aided language learning

**Peter Vlugter and Alistair Knott**
Dept of Computer Science, University of Otago

## Abstract

The main topic of this paper is how to configure a dialogue system to support computer-aided language learning. The paper also serves to introduce our new multi-speaker dialogue system, and highlight some of its novel features.

## 1 Introduction: language-learning dialogues

The dialogue system described in this paper is intended for use in a language-learning application. Computer-aided language learning (or CALL) is an interesting application for dialogue systems, for several reasons. Firstly, from a practical point of view, a language-learner's grammar and vocabulary are smaller than those of a native speaker, and therefore the problem of interpreting and generating unrestricted free text may be simpler in this domain than in other domains. On the other hand, the student's knowledge of dialogue management largely carries over from her native language to the language being learned. Elements such as clarification subdialogues, and checking questions are common in CALL dialogues—perhaps even more common than in ordinary dialogues. Thus a language-learning dialogue might be quite complex at the level of dialogue structure, even if its constituent utterances are syntactically restricted. This makes CALL dialogue an interesting test domain for a practical dialogue system.

Equally importantly, dialogue is a natural medium for language-learning. What the student is learning to do is to speak to other people in the new language, so in talking to the tutor, the student is actually performing the very task to be learned. The tutor essentially plays two roles: firstly, as a conversational partner he engages in a real dialogue, about some chosen topic or task; secondly, as tutor he gives feedback about the student's utterances, and is available to answer questions about the language.

Language-learning dialogues often take place between two participants: a student and a tutor. However, many of the most important elements of a new language being learned are hard to exercise in this envrironment. In any new language, one of the important early topics is the system of **personal pronouns**: referring expressions, or possessives, or relative pronouns. To provide a natural environment for these expressions, more than two participants are needed in the conversation. A similar consideration applies to verb inflections (which in many languages must agree with the person and number of the subject and/or object in the sentence), and addressee terms (which are only needed if there is more than one potential addressee).

In this paper, we will describe a multi-speaker dialogue system which we have configured for use in a CALL environment for learning M¯aori, the indigenous language of New Zealand. In Section 2, we introduce the theoretical background for the system and the basic dialogue management principles. In Section 3, we describe how the system supports multi-speaker dialogue. In Section 4, we show how the multi-speaker system provides a natural environment for practice with the M¯aori pronoun system (which is particularly fiendish). In Section 5, we describe how the system implements a set of educational goals for a given interaction, and monitors the progress of these goals by maintaining a student model. In Section 6, we describe how the system-played characters choose their contributions with an educational goal in mind, to create opportunities for the student to acquire targeted constructions. Throughout the paper, we emphasise the topic of personal pronouns, to motivate the need for a multi-speaker CALL environment.

## 2 Background to the dialogue system

### 2.1 Semantic framework

Our system, called Te Kaitito, is designed to teach the M¯aori language. Our treatment of utterance

*brandial 2006: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue*, pages 18–25.

syntax and semantics has been described elsewhere (Vlugter *et al.* (2004); Bayard *et al.* (2002)) so we will provide a short summary of these topics here.

The system supports 'deep' semantic processing. The user enters her utterances as written text. Each utterance is first parsed, using an HPSG-style grammar, which delivers a semantic representation as output, in the formalism of Minimal Recursion Semantics (MRS; Copestake *et al* (1999)). The MRS representation is converted into a DRT-like format, which represents an utterance as an **context-update** operation. Each update is specified for a **dialogue act**, a **speaker**, an **addressee**, and a **message**. The message comprises a set of **presuppositions** which must be satisfied against a representation of the common ground, and a **nucleus**, which holds the propositional part of the utterance, whose force is different for different dialogue acts.

The common ground is represented as a DRS. At any time, the common ground contains a set of **referents**, and a set of propositions about these referents. To support the interpretation of plurals, these referents can be **group entities**, whose semantics are defined as in Kamp and Reyle (1993). For example, the referent associated with the coordinated NP *John and Mary* (in fact presupposed by this NP) would be as shown in Figure 1.

| |
|---|
| x1 x2 x3 |
| x3 = x1 $\oplus$ x2 |
| named(x1, "John")<br>named(x2, "Mary") |

Figure 1: Group entity presupposed by *John and Mary*

## 2.2 Dialogue model for two-speaker dialogue

For a two-speaker dialogue, our model of dialogue structure includes a fairly standard set of dialogue moves. Questions can be QUERY (a genuine question), CLARIFY (for a clarification question) and CHECK (for a question which explicitly checks the student's knowledge). Assertions can be ASSERT (for a new fact apropos of nothing) and ANSWER (a question). Yes-no questions and their answers are distinguished from *wh*-questions in each case. We also have various acts for greeting and farewell.

The dialogue model specifies legal sequences of dialogue acts. The simplest structures involve a forward-looking act paired with a backward-looking act; for instance QUERY followed by ANSWER, or ASSERT followed by ACCEPT. More complex examples can be formed by nested subdialogues. Several types of utterance can be followed by a CLARIFY question; as is now common, such a question is pushed onto a dialogue stack, so that a subsequent response is understood as addressing the most recent question. A resolved clarification subdialogue is removed from the stack altogether (and as a side-effect, the semantics of the utterance which has been clarified is updated).

## 2.3 Mixed-initiative dialogue

Our system supports a variety of mixed-initiative dialogue with the user. When there is nothing on the stack, the user is given the initiative, but she can choose to concede it simply by hitting 'return'. If the user takes the initiative, the system will interpret the utterance, and then generate a response utterance. If the user concedes the initiative, the system will generate an initiative of its own, and then ask for a response from the user. If the system's initiative was a question, the user must answer it (or ask a clarification); if the system's initiative was an assertion, the user can give an explicit acknowledgement, or generate a new initiative (which will be treated as an implicit acknowledgement of the assertion). Thus there are two situations in which the system interprets the user's utterance—first if it is an initiative, and second if it is a response to a system initiative.

## 2.4 The utterance interpretation pipeline

Interpretation of a user utterance involves the same sequence of processes, whether the utterance is an initiative or a response. The sentence is first parsed, producing a set of alternative parses. Each parse is associated with a semantic interpretation. Each interpretation is converted into an update, yielding a set of candidate updates for the sentence. To choose the most appropriate update, an attempt is made to attach each candidate to the common ground, by resolving its presuppositions and establishing a relationship with dialogue acts on the stack. The update which is easiest to attach is selected as the preferred interpretation. Any ties are resolved by statistical parsing techniques and various types of clarification question; for details see Lurcock *et al* (2004).

## 3 Our current multi-speaker system

To support a multi-speaker environment, our system is designed to 'play' several different characters, who have separate knowledge bases, and who can communicate both with the user and with each other. The system-played characters are relatively autonomous, keeping their own private version of the common ground, and updating it independently. The multi-speaker system is described and motivated in detail in Knott and Vlugter (in press); in this section, we summarise the most important aspects of the system.

### 3.1 The conversation management algorithm

The interaction between the system and the user is governed by a high-level **conversation manager**. Basically, the user and the system take turns to contribute to the conversation; but while the user contributes at most one utterance, the conversation manager implements a number of loops over all system-played characters. There are three basic loops, which operate at different points in the conversation.

#### 3.1.1 The system initiative loop

At any point when the stack becomes empty, initiative is passed to the user, but the user can choose to concede the initiative by hitting 'return'. If the user concedes, the system must generate a new initiative.

Since the system plays several characters, generating an initiative is not simply a matter of choosing what to say: it must also be decided which character will talk next. These two decisions interact, because each character has a separate knowledge base, and therefore the options open to one character may not be open to another.

In our two-speaker system, initiative selection is done by generating a set of candidate utterances and giving each a score, with the highest-scoring utterance being delivered to the user (Slabbers, 2005). In the multi-speaker system, the highest-scoring utterance is computed for each character, and the character with the highest score is selected as the one to talk next. There is a certain amount of private communication between system characters in this process; basically, the best utterance is selected without regard for who will say it.

In the multi-speaker system, the character who is selected to speak next must also decide who to *address*. Again, since there is nothing on the stack when an initiative is taken, the addressee is unconstrained. In our system, we implement a simple rule, which is dictated by the educational domain: system characters always address the user, so that the user is maximally involved in the conversation.

#### 3.1.2 The inter-character update loop

We assume that all system characters are 'active overhearers'—what Goffman (1976) calls **ratified side-participants**—who actively interpret each utterance in the conversation, even if it is not addressed to them. Therefore, when one system character generates an utterance, each other system character must update its common ground accordingly.

Naturally, the other characters do not need to interpret the utterance from scratch. However, since they all maintain their own version of the common ground, each must do a certain amount of work independently. The point in the interpretation pipeline where the private work begins is the point where the MRS of the selected parse (i.e. its semantic representation) is converted to an update. Therefore, after every utterance by a system character, the MRS associated with that utterance is passed to each other system character, and each processes the MRS privately. Effectively, system characters communicate with each other at the level of MRS representations, rather than sentences. This eliminates the possibility of syntactic ambiguity, but note that the potential for pragmatic ambiguity remains, if any discrepancies arise between different characters' versions of the common ground.

#### 3.1.3 The system response loop

If the user chooses to take an initiative, it can be an assertion or a question. In either case, each system character needs to interpret it, and update their common ground. Then those system characters who were addressed need to give responses.

**Interpreting the user's utterance** There is no need for each character to parse the user's utterance, or disambiguate it. A single character is chosen at random to perform these processes, and to engage in clarification subdialogues with the user if necessary. The result of these processes is a single update; the MRS associated with this update is then communicated to all system characters. Each character then creates their own version of the update for this MRS, and incorporates this update into their private common ground.

**Group responses** In a multi-speaker conversation, an utterance can be addressed to the full group of participants, or to a specific participant, or to a subset of participants: i.e. for every utterance there is an **addressee group**. A basic principle in our system is that every participant in the addressee group of a forward-looking utterance has the right to respond to it—and that at least one member of this group *must* respond to it. To keep things simple, we specify that responses by different members of the addressee group should not overlap.[1] Thus group responses are implemented as strict sequences of utterances by members of the addressed group. There is some evidence that group responses are generated in this way in spoken conversation, at least for some dialogue acts; see e.g. Ginzburg and Fernández (2005).

The system's response to a user initiative is implemented as a loop over all system characters, chosen in an arbitrary order. Naturally, not all of these characters will be in the addressee group. Each character determines independently whether they are an addressee. (See Section 3.2 for a discussion of how this is done.) If they are not an addressee, they remain silent. If they are an addressee, and a response has not yet been given, they are obliged to give one.

Responses are not delivered directly to the user. Each new utterance is added to a buffer containing the full group response, which is delivered in one piece to the user when complete. Thus the user is not able to interrupt in between respondents. Also, each character who delivers a response broadcasts the MRS for their response to all other system characters, who process it and add the update to a private data structure called the **group response record**, which they consult when generating a subsequent response of their own.

If a character adds a response to a non-empty group response record, they can choose to preface the response with an **inter-respondent dialogue act**, signalling agreement or disagreement with the previous respondent before presenting its own response.[2] A character can also choose to give no response, which is interpreted as signalling agreement with all the responses given so far.

A group response generated by our system is given in Dialogue 1.

(1)
User: John, Bill, where do Mary and Sue live?
Bill: Well Mary lives in Dunedin.
John: That's right. And Sue lives in Auckland.

Here, the user addresses a question to two system characters, John and Bill, who respond in turn. Bill responds first, providing a partial answer (introduced by by *Well*, to signal that it is incomplete). John responds next; he begins by agreeing with Bill's answer, and then fleshes this out with an answer of his own. More details about how group responses are generated can be found in Knott and Vlugter (in press).

### 3.2 Addressee selection

When interpreting a user utterance, whether it is an initiative or a response, each system character must determine to whom it is addressed, to decide whether or not they should formulate a response. There are various types of information which contribute to this decision in real-life dialogue (see Sacks *et al.* (1974), and much subsequent work). Some of these involve eye gaze or gesture, and are beyond the scope of our current system. But there are nonetheless several different sources of information about the addressee which are available; in this section, they are discussed in turn.

Most obviously, the addressee of an utterance can be specified linguistically, as a modifer of the utterance, as in the first utterance of Dialogue1. Much has been written about the grammar and semantics of addressee terms, which are quite different from ordinary referring expressions (see e.g. Longobardi (1994)). Clearly, an explicit addressee term unambiguously identifies the intended addressee of an utterance.

If there is no explicit addressee, the structure of the preceding conversation provides some indication of the addressee group. Two types of context can be distinguished. If there is a forward-looking dialogue act on the stack, and the next utterance responds to this act, the addressee of the response utterance should obviously be the speaker of the act being responded to. Thus in Dialogue 2, the user's response should be understood as addressing Bill.

(2)
Bill [to User]: What is your name?
User: My name is Ursula.

We refer to the addressee of a backward-looking dialogue act as a **structurally-defined addressee**.

[1] In fact, given that our system uses written text as an interface, rather than spoken text, it is not even clear what overlapping responses would look like.

[2] In fact, only agreement is implemented so far.

The structural constraint seems quite strong; if an explicit addressee term is given which conflicts with it, the result is quite jarring. However, if there is a conflict, our system gives precedence to the explicitly specified addressee. This element of our addressee selection algorithm is very similar to that proposed by Traum (2004).

If there is no forward-looking dialogue act on the stack, the next utterance is a new initiative, which can in principle be addressed to any subgroup of participants. However, we suggest that if the user produces a new initiative without an explicit addressee term, (s)he is probably continuing an interaction with participants who have just been involved in the conversation, rather than making a switch to a new group of participants. We therefore define a **default addressee group** for an utterance making a new initiative, which consists of the group of participants involved in the most recent subdialogue (minus the speaker of the utterance). In Dialogue 3, for instance, it seems likely that Sue's second utterance is addressed to both Bill and Bob, rather than to one or the other separately, or (still worse) to someone else entirely.

(3)
Sue: What's the time?
Bill: Five o'clock.
Bob: Yup, five o'clock.
Sue: Do you want to go for tea?

This notion of default addressee is somewhat different from that proposed in Traum (2004); see Knott and Vlugter (in press) for a more detailed comparison. The default addressee is obviously a fairly weak notion, and an explicit addressee overrides it very easily. However, it makes for natural dialogues if the user fails to specify an addressee.

A final interesting issue in our system concerns the identity of the addressees of utterances occurring as part of a group response. Who is each member of a responding group addressing in their utterance? The most recent respondent? The speaker of the forward-looking act being responded to? The full group of respondents? Our solution is to sidestep these questions, as they do not seem to have clearcut answers. Since the notion of the addressee group is primarily used to decide who goes next, our solution is simply to specify a prearranged sequence of speakers during group responses, as described in Section 3.1.3.

### 3.3 Pronoun interpretation and generation

Recall that the purpose of our multi-speaker dialogue system is to create an environment in which a language learner can exercise various aspects of the language being learned. In our case, to provide a natural environment for exploring the system of personal pronouns, we needed a system which allows a conversation between multiple speakers. In this section, we describe how the multi-speaker system supports the generation and interpretation of a range of personal pronouns, focussing on our target language, M¯aori.

The pronoun system in M¯aori is more complex than that of English. Pronouns are specified for 'number' and 'person', but there are three values for the 'number' feature (singular, dual and plural); in addition, for first-person dual and plural pronouns, there is a distinction between **inclusive** pronouns (which include the addressee) and **exclusive** pronouns (which do not). Thus, for instance, the pronoun *māua* means 'us two (not including you)', while *tātou* means 'us three (or more), including you'. Clearly, these are expressions which a student will need to practice.

In our system, pronouns are presuppositional; different pronouns presuppose groups with different properties. For the sake of generality, all pronouns presuppose a group entity: a singular pronoun requires the group entity to have cardinality 1, a dual pronoun requires it to have cardinality 2, and so on. A first-person pronoun requires a group entity which includes the speaker as one of its members; a second person pronoun requires a group entity which includes the addressee group, and which does not include the speaker. A third-person pronoun requires a group entity which includes neither the speaker nor the addressee group. Finally, all pronouns presuppose groups which are linguistically **salient**. The speaker and addressee group are salient automatically, by virtue of their role in the current interaction. In our implementation, the only other way a group can become salient is through having been referred to in the recent conversation. (Our precise algorithm for determining salience is very simple, but could of course be elaborated in many different directions.)

Having specified the appropriate properties for pronouns, the interpretation of an utterance containing a pronoun is handled fairly automatically by the presupposition resolution system. If the referent of a pronoun is unclear (typically because there are several candidates with roughly equal salience), a referential clarification question is asked (e.g. *WHICH man is your friend?*). However, the generation of utterances containing pronouns is more complex. First and second-person singular pronouns are unambiguous, and thus relatively straightforward, but third-person pronouns and plural pronouns are often ambiguous; for instance, *you(pl)* could refer to several different groups including the addressee. If the alternative groups have similar salience, the system chooses to generate a full referring expression (which will involve coordination for plural referents). But if the intended referent is clearly more salient than its competitors, a pronoun is allowed.

## 4 Multi-speaker conversation as a CALL environment for learning pronouns

The previous section summarised how the important aspects of our multi-speaker environment are implemented. In this section, we provide an example of the kind of CALL dialogue which our system supports.

The interaction can be divided into two phases. In one phase, the student engages in an open-ended dialogue with the system characters, and information is exchanged between all participants. An example of this phase is given in Figure 2. Note that while the student is relatively passive in this interaction, she is exposed to a range of different personal pronouns. In a second phase, the system generates a sequence of checking or comprehension questions for the student. An example of this phase is given in Figure 3. Note that checking questions are generated by different characters; this is to ensure that a good mixture of referring expressions is generated, to test the student's knowledge of these concepts.

## 5 Implementing educational goals and student modelling

The dialogues in Figures 2 and 3 may appear serendipitous; the phase 1 dialogue happens to introduce a number of groups, which allow the phase 2 checking questions to target certain pronouns. In fact, the initiative-selection routine in our system is configured to encourage the right groups to be introduced, and the right checking questions to be asked. The initiatives taken by system characters are evaluated primarily on their educational usefulness, rather than on their content *per se*. In

| | |
|---|---|
| User: | Hello. |
| Mary: | Hello. |
| John: | Hi. |
| Bill: | Hello. |
| User: | ⟨return⟩ |
| Mary: | Ursula, where are you going today? |
| User: | I am going to the beach. |
| Mary: | Okay. |
| User: | ⟨return⟩ |
| John: | I am going to the beach too, Ursula. |
| User: | ⟨return⟩ |
| Mary: | Bill and I are going to a restaurant, Ursula. |
| User: | ⟨return⟩ |
| Mary: | We(dual,excl) always go to Comme Chez Soi. |
| User: | ⟨return⟩ |
| John: | I always go to Comme Chez Soi too, Ursula. |
| User: | ⟨return⟩ |

Figure 2: First phase of a CALL dialogue (translated from M¯aori)

| | |
|---|---|
| John: | Where are we(dual,incl) going today, Ursula? |
| User: | We(dual,incl) are going to the beach. |
| John: | Well done! |
| Mary: | Where are you and John going today, Ursula? |
| User: | We (dual,excl) are going to the beach. |
| Mary: | Well done! |
| John: | Where do Bill and Mary always go, Ursula? |
| User: | They(dual) always go to Comme Chez Soi. |
| John: | Well done! |
| Mary: | Where do Bill, John and I always go, Ursula? |
| User: | You(plur) always go to Comme Chez Soi. |
| Fred: | Where do John, Bill and Mary always go, Ursula? |
| User: | They(plur) always go to Comme Chez Soi. |

Figure 3: Second phase of a CALL dialogue: checking questions (translated from M¯aori)

this section, we describe how this evaluation takes place.

### 5.1 Lesson authoring

To begin with, we need a method for telling the system what the educational agenda is for a given lesson. In our system, we create a sequence of lessons, each of which takes the form of a dialogue, with new topics about the language introduced in each lesson.

To create the content for a given lesson, characters need to be given an appropriate knowledge base. To build these knowledge bases, our system supports a special kind of dialogue called an **authoring dialogue**, in which a human author simply tells each character what they know (Slabbers, 2005). The authoring dialogue for each lesson is preprocessed to build the educational agenda for the lesson. The agenda in our system is simply a list of the lexical types which appear in the current authoring dialogue, but not in previous ones. The student must demonstrate understanding of each of these new types during the dialogue in order to move to the next lesson.

### 5.2 Authoring the pronoun agenda

To create an agenda for pronouns, it is somewhat inconvenient to work directly from the authoring dialogue, since this requires the author to set up the right environment for using each pronoun to be included in the agenda. In our system, we simply introduce a **command** which the author can type instead of an utterance: the keyword `pronoun:` followed by a specific pronoun. This adds the pronoun in question to the agenda for the current lesson. For instance, the command `pronoun: tāua` would add *tāua* to the pronoun agenda.

### 5.3 The student model

The student model is an overlay on the educational agenda, indicating the degree to which each construction in the agenda has been assimilated by the student. At a given point in a lesson, for instance, the agenda might contain the pronouns *tāua* (we/dual-inclusive) and *māua* (we-dual-exclusive), and the student might have shown some assimilation of the former, but none of the latter.

There are two ways in which the student model can be updated. Firstly, if the student uses a sentence which can be successfully attached to the common ground, the assimilation score for each construction in the sentence is incremented. Secondly, if the student correctly answers a question, the assimilation score for each construction in the question is incremented. These methods are quite simplistic—and at present we have no method of *decrementing* assimilation scores in response to student errors. These are topics for future research. Basically, the lesson continues until the student has shown sufficient assimilation of each construction on the agenda.

## 6 Educational content selection strategies

The goal of our dialogue system is that the student assimilates all the constructions on the agenda for the current lesson. This is very different from the goal of most dialogue systems. Typically the goal is content-based—for instance, communicating a particular set of facts to the user, or obtaining a particular set of facts from the user. In our system, the goal is *form-based*—we require that the user correctly employs, or shows understanding of, a certain range of grammatical constructions.

One way the system can facilitate achievement of the goal is through a targeted content selection strategy. There are two ways content selection can help achieve the goal. The most direct way is to give the student the opportunity to display knowledge about a particular construction. For instance, by asking a checking question using a particular pronoun, or one whose answer requires the use of a particular pronoun, the student is given an opportunity to show they can use it by giving the right answer.

However, this direct approach is not always possible for pronouns, because it is not always possible to use a given pronoun felicitously. A more indirect content selection strategy should therefore encourage the creation of an *environment* in which the targeted pronouns can be felicitously used. For instance, to allow the student to use *tāua* (we-dual-inclusive), there needs to be a salient group containing exactly two individuals, which includes the student and the addressee. So a useful character initiative would be one which creates this group, if it does not already exist. What is more, there has to be something predicated of the group, to serve as the content of an utterance about the group.

In our system, characters create appropriate groups dynamically, simply by making facts up. For instance, if the user has asserted a predicate about themselves, a system character might take an initiative asserting that this predicate is also true of himself, or of a group of participants including himself. This has the additional benefit of adding **cohesion** to the dialogue. For instance, in Dialogue 2, John's utterance *I am going to the beach today too, Ursula* is a very appropriate initiative, since it maintains the topic of the user's previous utterance. However, the main reason for John to generate this utterance is to allow him later to ask the user the checking question *Where are we (dual,inclusive) going today, Ursula?* during phase two of the lesson. Note that the utterance also allows other checking questions; for instance, Mary can ask a question (*Where are you and John going today, Ursula?* which sets up the user to respond with a suitable dual exclusive pronoun.

## 7 Summary and further work

In this paper, we have presented a multi-speaker human-machine dialogue system which is configured to function as a CALL tool. The multi-speaker system is of interest in its own right, as it uses a novel conversation management architecture, and some novel methods for addressee selection. In particular, its approach to the handling of group responses has some novel features. However, it is also of interest in that it provides a novel kind of CALL environment, where the student can exercise her knowlege of elements of the language being learned which depend on multiple speakers. In our case, the focus is on a system of personal pronouns and addressee terms, but in other languages, the same multi-speaker environment would be useful in giving a student practice in other topics, for instance in verb conjugation.

It is interesting to compare the current approach to CALL dialogues to the type of exercises which are supported by conventional CALL software. Typical CALL exercises involve canned materials: a hand-built text is provided, followed by hand-built comprehension questions tailored to test the targeted knowledge. In our system, the user's interaction with the system is a lot more flexible; the user simply engages in a wide-ranging mixed-initiative dialogue. However the system-played characters are acting behind the scenes in the same kind of way as the human author of a canned CALL exercise, to create opportunities to test the student's knowledge.

It is not clear whether the extra flexibility in our CALL interactions is of any value to the student. Obviously, when learning a language it is very beneficial to interact with native speakers in as natural an environment as possible. Our aim is basically to recreate this kind of interaction (or at least to approximate it), while engineering the flow of conversation so as to create useful educational opportunities. In future work, we will evaluate the system to see if this aim is met.

## References

I Bayard, A Knott, and S de Jager. 2002. A unification-based grammar for a fragment of M¯aori and English. In *Proceedings of the 2nd Australasian Natural Language Processing Workshop (ANLP 2002)*.

A Copestake, D Flickinger, I Sag, and C Pollard. 1999. Minimal Recursion Semantics: An introduction. Manuscript, CSLI, Stanford University.

J Ginzburg and R Fernández. 2005. Action at a distance: the difference between dialogue and multilogue. In C Gardent and B Gaiffe, editors, *Proceedings of DIALOR'05: The ninth workshop on the semantics and pragmatics of dialogue*, pages 85–92, Nancy, France.

E Goffman. 1976. Replies and responses. *Language in Society*, 5:257–313.

H Kamp and U Reyle. 1993. *From discourse to logic*. Kluwer Academic Publishers, Dordrecht.

A Knott and P Vlugter. in press. Multi-agent human-machine dialogue: issues in dialogue management and referring expression semantics. *AI journal*.

G Longobardi. 1994. Reference and proper names: a theory of N-movement in syntax and logical form. *Linguistic Inquiry*, 25(4):609–665.

P Lurcock, P Vlugter, and A Knott. 2004. A framework for utterance disambiguation in dialogue. In *Proceedings of the 2004 Australasian Language Technology Workshop (ALTW)*, pages 101–108, Macquarie University.

H Sacks, E Schegloff, and G Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735.

N Slabbers. 2005. A system for generating teaching initiatives in a computer-aided language learning dialogue. Technical Report OUCS-2005-02, Department of Computer Science, University of Otago, Dunedin, New Zealand.

D Traum. 2004. Issues in multi-party dialogues. In F Dignum, editor, *Advances in agent communication*, pages 201–211. Springer Verlag Lecture Notes in AI 2922.

P Vlugter, A Knott, and V Weatherall. 2004. A human-machine dialogue system for CALL. In *Proceedings of InSTIL/ICALL 2004: NLP and speech technologies in Advanced Language Learning Systems*, pages 215–218, Venice.