Document Transformations and Information States

Staffan Larsson
Dept. of linguistics
Göteborg University
Sweden
sl@ling.gu.se

Annie Zaenen
Xerox Research Centre Europe
Grenoble Laboratory
France
Annie.Zaenen@xrce.xerox.com

Abstract

We discuss ways to explore how instructional material needs to be structured to be presented with various degrees of interactivity. We use the TRINDI[1] information state approach to model three different degrees of interactivity and present IMDiS, a small experimental implementation based on the GoDiS dialogue system.

1 Introduction

Document transformations is becoming a hot topic in industrial research on document creation. The reason is practical: with the new presentation possibilities, the advantages of being able to adapt the 'same' document content to different uses - where the difference can lie in the support devices, audiences, languages or modes of interaction - becomes very attractive. It not only becomes attractive, it also becomes necessary: one needs to present material in various contexts (oral presentations, internet portals, etc.) and it is very costly to develop presentations from scratch for these various contexts.

This situation raises an old question and opens a new area of research: can one separate content from presentation? The philosophical answer might be 'no', but in practice one doesn't need an absolute answer. As this area of research arises more out of practical necessity than pure intellectual curiosity, the

---

[1]TRINDI (Task Oriented Instructional Dialogue), EC Project LE4-8314, www.ling.gu.se/research/projects/trindi/

engineering is preceding the science and it will take some time before it rest on explicit solid foundations.

Here we look only at one small aspect of the problem: how can we model small changes in presentation that are due to various degrees of interactivity between participants in instructional exchanges. We start from a traditional manual and make some assumptions about minimal interactivity which are modeled through dialogue moves. We conclude that in this way we can make the presentation of the material more flexible. An important limit on the flexibility is, however, the detail with which the discourse structure of the manual encodes the task plan underlying the activity.

2 Degrees of Interactivity and the difference between monologue and dialogue

We take here the position that the main difference between dialogue and monologue is that the former implies interactivity. With interactivity we mean here that the participants can influence each other's moves. With respect to the area that interests us here, giving instructions to repair devices, a traditional written manual influences the user but not vice versa (except through notes to the author). The user can, however, influence the order in which she accesses the material: it is easy to stop, to go back or to consult an other section (traditional printed material might be argued to be better in that respect than presentation on a screen, we ignore that difference here). We can consider this as a limit case of interactivity.

Note that interactivity does not necessarily imply shared initiative. The literature makes a distinction between task and dialogue initiative (e.g. (Chu-Carroll and Brown, 1998)) but one can have dialogue with both types of initiative staying with one side. In the cases we discuss below the task initiative stays completely with the manual and the dialogue initiative only switches to the instructee in the case where she can indicate that information about some subprocedures can be skipped.

There is another dimension that often intervenes in discussions about the difference between dialogue and written discourse: the former is spoken, the latter is written. Given the way things are in a natural setting, the written medium tends not to allow interactivity, whereas the spoken medium is used mainly in interactive settings. Technical changes, however, allow us to separate the written/spoken opposition from that between interactive and non, or minimally, interactive discourse. Instructional material can be presented in the aural mode without becoming more interactive e.g. when a recording is played. This can be considered as a plus for instructional material because it allows the instructee to use her hands and eyes for the task itself but it is not an unqualified advantage given that reading gives much more flexibility than listening to a tape. To cash in on the advantages of the aural presentation, we need to recapture the flexibility of access that the written medium allows.

3 Instructions and Interactivity

It is obvious that instructional situations profit from an interactive setting. Instructional situations are typically situations in which some participants (the instructors) know a lot that the other participants (the instructees) need to know to achieve the common goals. In these kinds of situations it is important that all the required and, preferably only the required, knowledge gets transferred at the moment the instructees need it. To achieve this, it is not enough that the instructor have all the necessary knowledge, she needs also to know which state the in-

structee is in and how that state changes to adapt the transfer of knowledge, hence the instructee needs to be able to inform the instructor about his state and influence in this way the course of the interaction.

Currently we have manuals, whose content can be presented aurally or in a written form but where both the content and the presentation are uniquely determined a priori (modulo, the speed and order of reading mentioned above). Or we have interactions that can be at a distance but where a human instructor needs to be available at the time of the action. Making humans with the required competence available is expensive and one would want to achieve some interactivity without this. But computers tend to be frustrating participants in interactive settings when one compares them to human beings and the study of dialogue concentrates mainly on making them as human as possible. When one considers the possibility of transferring the interactivity from humans to machines, there are, however, many intermediate possibilities between no interactivity and full blown interactivity in free-wheeling dialogue where the participants can ask each other questions about anything and nothing (for a more thorough discussion about dialogues between humans and computers see (Clark, 1999)). In this paper we consider how minimal interactions can be modeled on the basis of information which is available in traditional instructional manuals.

In looking at the problem this way one has to keep in mind that instructional manuals, although not interactive, are cooperative constructs: they assume that they participate with the user in a rational cooperative task and they are built on an implicit reader model, specifically they make assumptions about what the user knows and what she doesn't know and the granularity of the task descriptions that they have to provide. They obey in their own way Grice's Maxim of Quantity but they need to leave open a range of possibilities so they need to provide more detail than is necessary in all circumstances. In what follows we can only consider

cases of over-informedness as the information needed to remedy under-informedness is not available.

4   The TRINDI model

The TRINDI project has developed both a framework and a toolkit to model various types of interactions in terms of information state updates. The framework, whose main ingredients are information states, dialogue moves and updates, is described in (Traum et al., 1999). We use the term *information state* to mean, roughly, the information stored internally by an agent, in this case a dialogue system. A *dialogue move engine updates* the information state on the basis of observed dialogue moves and selects appropriate moves to be performed. Information state updates are formalised as information state *update rules*. The importance of the framework is that new interactive hypotheses can be modeled with minor extensions. The information state approach is implemented in the TRINDIKIT (Larsson et al., 2000); (Larsson and Traum, To appear), a toolkit for experimenting with the implementation of information states and dialogue move engines and for building dialogue systems. It is used in the experimental implementation described here.

Various instantiations of the framework articulate further what information states, moves, and update rules contain. In this paper we use one formal representation of information states that has been developed in the TRINDI, SDS[2] and INDI[3] projects, and implemented in the GoDiS dialogue system (Bohlin et al., 1999). The central parts of the information state in GoDiS are *dialogue plans* and *Questions Under Discussion* (QUD), a notion borrowed from Ginzburg (Ginzburg, 1998).

5   Modeling various degrees of interactivity in TRINDI

We envision the following cases:

- 1. Traditional manual: no overt interaction, we will consider this as the limit case

- 2. Manual can ask yes/no questions and understand two types of user responses:
  - yes/no
  - done/don't understand
  - how?

- 3. User can indicate whether she already knows certain (sub)procedures

5.1   GoDiS/IMDiS information states

To model the types of interactions above, we started from the GoDiS system which is designed to deal with information-seeking dialogue. The IMDiS information state type is shown in Figure 1.

$$
\begin{bmatrix}
\text{PRIVATE} & : & \begin{bmatrix}
\text{PLAN} & : & \text{StackSet(Action)} \\
\text{AGENDA} & : & \text{Stack(Action)} \\
\text{TMP} & : & \text{(same as SHARED)}
\end{bmatrix} \\
\text{SHARED} & : & \begin{bmatrix}
\text{BEL} & : & \text{Set(Prop)} \\
\text{QUD} & : & \text{StackSet(Question)} \\
\text{ACTIONS} & : & \text{Stack(Action)} \\
\text{LU} & : & \text{Utterance}
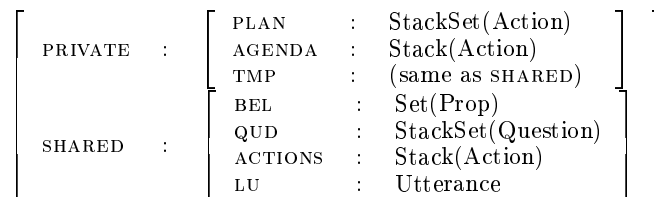\end{bmatrix}
\end{bmatrix}
$$

Figure 1: IMDiS information state type

The main division in the information state is between information which is private to the agent and that which is shared between the dialogue participants. The private part of the information state contains a PLAN field holding a dialogue plan, i.e. is a list of dialogue actions that the agent wishes to carry out. The plan can be changed during the course of the conversation. The AGENDA field, on the other hand, contains the short term goals or obligations that the agent has, i.e. what the agent is going to do next. We have included a field TMP that mirrors the shared fields. This field keeps track of shared information that has not yet been grounded, i.e. confirmed as having been understood by the

other dialogue participant. The SHARED field is divided into four subfields. One subfield is a set of propositions which the agent assumes for the sake of the conversation. The second subfield is for a stack of questions under discussion (QUD). These are questions that have been raised and are currently under discussion in the dialogue. The ACTIONS field is a stack of (domain) actions which the user has been instructed to perform but has not yet performed. The LU field contains information about the latest utterance.

To adapt GoDiS to instructional dialogue, we added a subfield of SHARED.ACTIONS to (the shared part of) the information state. The value of this field is a stack of actions which the system has instructed the user to perform, but whose performance has not yet been confirmed by the user.

In building the experimental IMDiS, we have made several simplifications. We have ignored all the natural language generation problems and all the problems related to making text or dialogue natural, e.g. problems related to the use of pronouns and other referential expressions. To handle these we would not only have to discuss basic interactivity but also the medium in which the interaction takes place: speech or written text.

The monologue mode (case 1) uses only 2 moves (**Instruct**, and **Inform**). Since there is no user to confirm that actions have been performed, all actions are automatically confirmed using the update rule **autoConfirm**.

RULE: **autoConfirm**

CLASS: **integrate**

PRE: $\left\{\begin{array}{l} \text{fst(SHARED.ACTIONS, } A \text{)} \end{array}\right.$

EFF: $\left\{\begin{array}{l} \text{pop(SHARED.ACTIONS)} \\ \text{add(SHARED.BEL, done(}A\text{))} \end{array}\right.$

The dialogue version (cases 2 and 3) uses 9 move types, basically the 7 used in GoDiS (**Ask**, **Answer**, **Inform**, **Repeat**, **RequestRepeat**, **Greet**, **Quit**) plus instructions (**Instruct**) and confirmations (**Confirm**). Confirmations are integrated by assuming that the current topmost action in SHARED.ACTIONS has been performed, as

seen in the update rule below.

RULE: **integrateUsrConfirm**

CLASS: **integrate**

PRE: $\left\{\begin{array}{l} \text{val(SHARED.LU.SPEAKER, usr)} \\ \text{assoc(SHARED.LU.MOVES, confirm, false)} \\ \text{fst(SHARED.ACTIONS, } A \text{)} \end{array}\right.$

EFF: $\left\{\begin{array}{l} \text{set\_assoc(SHARED.LU.MOVES, confirm, true)} \\ \text{pop(SHARED.ACTIONS)} \\ \text{add(SHARED.BEL, done(} A \text{))} \end{array}\right.$

This rule says that if the user performed a **Confirm** move, which has not yet been integrated, and A is the "most salient" action, then integrate the move by putting the proposition **done**(A) in the shared beliefs, and taking A off the action stack.

Elliptical "how"-questions from the user are interpreted as applying to the currently topmost action in the SHARED.ACTIONS stack.

5.2 Domain task, manuals and dialogues

Let's now see how a monologue and a dialogue version of the same task are related. Below we have an example from the user manual for the HomeCentre, a Xerox MFD.

- Reinstalling the print head

- Caution: Make sure that the green carriage lock lever is STILL moved all the way forward before you reinstall the print head.

- 1. Line up the hole in the print head with the green post on the printer carriage.

- Lower the print head down gently into position.

- 2. Gently push the green cartridge lock lever up until it snaps into place.

- This secures the print head.

- 3. Close the top cover and reattach the scanner.

- 4. Press and release the yellow LED button.

- The printer will prepare the cartridge for printing.

- Note: If the carriage does not move from the center position after you press the cartridge change button, remove and reinstall the print head.

From this text, one can (re)construct a task plan for reinstalling the print head. Such a plan may be represented as in figure 2. Note

NAME   reinstall(print_head)

PRE   moved_forward(carriage_lock)

DEC

line_up(hole, post)

lower(print_head)

push(lever)

close(top_cover)

reattach(scanner)

press_and_release(yellow_button)

yes   moved_from_center(print_head)   no

remove(print_head)

reinstall(print_head)

□ action

▣ complex action / plan
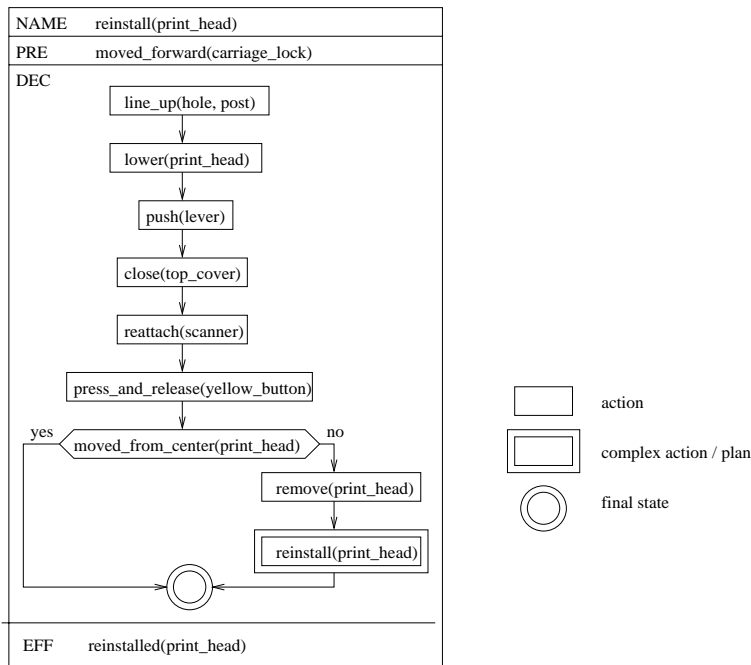
◎ final state

EFF   reinstalled(print_head)

Figure 2: Task plan

that this is a conditional plan, i.e. it contains branching conditions.

From this task plan, IMDiS generates two plans: a monologue plan and a dialogue plan. This is done using the "translation schema" in Figure 3.

The difference between the text plan and the dialogue plan is in the way that conditionals in the task plan are interpreted. In the monologue plan, they correspond to simply informing the user of the conditional. In dialogue mode, however, the system raises the question whether the condition holds. When the system finds out if the condition holds, it will instruct the user to execute the appropriate guarded action.

Here we can clearly see how dialogue differs from monologue as viewed by Carlson or Van Kuppevelt ((Carlson, 1983), (van Kuppevelt, 1995)). Under these views the writer anticipates the questions the user might have asked but given the user is not present the writer has to make up for the lack of interactivity. The questions that can be reconstructed (or accommodated) are different in that case. For instance in the example given here, the question could something like "What should the user/I make sure of?". These questions are valuable to help figure out the discourse structure of a monologue. They can also be valuable tools to illustrate the differences between dialogue and monologue but they do not give much insight in the effects of various degrees of interactivity.

Conditionals are treated as follows by the system in dialogue mode: When the system has found out what the user's task is, it will load the appropriate dialogue plan into the PRIVATE.PLAN field of the information state. It will then execute the actions in the appropriate order by moving them to the agenda and generating appropriate utterances. When a conditional statement is topmost on the plan, IMDiS will check whether it has been established that the condition holds (by checking the SHARED.BEL field). Since the system has previously asked the user and the user has answered, either the condition or its negation will be in the set of established propositions. If the condition or its negation holds, the conditional will be popped off the plan and replaced by the first or second guarded action (respectively).

DOMAIN	MONOLOGUE	DIALOGUE
precondition P	**Instruct**(check(P))	findout(P); if_then(not(P), **Instruct**(achieve(P)))
action A	**Instruct**(A)	**Instruct**(A)
if_then(C, A)	**Inform**(if_then(C, A))	findout(C); if_then(C, **Instruct**(A))
effect E	**Inform**(E)	**Inform**(E)

Figure 3: Plan conversion table

5.3 Monologue and Dialogue Behaviour

In the monologue mode in IMDiS, the control module does not call the input and interpretation modules. The text is output "move by move" as a sequence of utterances from the system.

```
S: Reinstalling the print head.
S: Make sure that the green carriage lock
lever is STILL moved all the way forward
before you install the print head.
S: Line up the hole in the print head with
the green post on the printer carriage
```

Compared to the monologue mode, even a very restricted dialogue mode offers several advantages:

User attention and control The user can direct her attention to the machine and does not have to look at the manual. As we noted in when one goes from written to aural presentation, one gains the advantage that the user has free hands and eyes but if nothing more is done this advantage has to be weighted against the disadvantage that the user looses all control over the order and the speed with which the information is presented. We can avoid these drawbacks by allowing some limited grounding behaviour. Very simple interactions like 'done' (**Confirm**) or 'don't understand' (**RequestRepeat**) give back to the user a limited control over the speed and the order of the presentation (at least up to allowing repetition): the user decides when to move on to the next action, by confirming that the previous action is done, and by 'don't under-

stand' she can indicate that she would want a repetition of what was said immediately before. Here we see how to take advantage of the advantages of a different mode of presentation (written versus aural) we also have to change the type of interactivity.

```
S: Has the carriage moved from the center
position?
U: I didn't understand
S: Has the carriage moved from the center
position?
```

Avoid irrelevant information When the action to be taken depends on a condition, the system does not give irrelevant information.

```
S: Has the carriage moved from the center
position?
U: yes
S: The print head is now installed
```

Because there is no feedback from the user, a manual has always to give all the possibilities regardless of which one actually pertains. The possibility to ask yes/no questions allows us to do away with this redundancy.

5.4 More complex task plans

In the example above we illustrated how a simple task plan can give rise to a dialogue and a monologue rendering. We can get some added flexibility by giving more structure to the task plan. For instance in the example above, one can argue that the reinstallation proper of the print head is described in point 1 to 2 and that 3 and 4 describe termination

NAME reinstall(print_head)
PRE moved_forward(carriage_lock)
DEC
- secure(print_head)
- close(top_cover)
- reattach(scanner)
- press_and_release(yellow_button)
- moved_from_center(print_head) — yes / no
- remove(print_head)
- reinstall(print_head)

EFF reinstalled(print_head)

NAME secure(print:head)
PRE
DEC
- line_up(hole, post)
- lower(print_head)
- push(lever)

EFF secured(print_head)

Legend:
- action
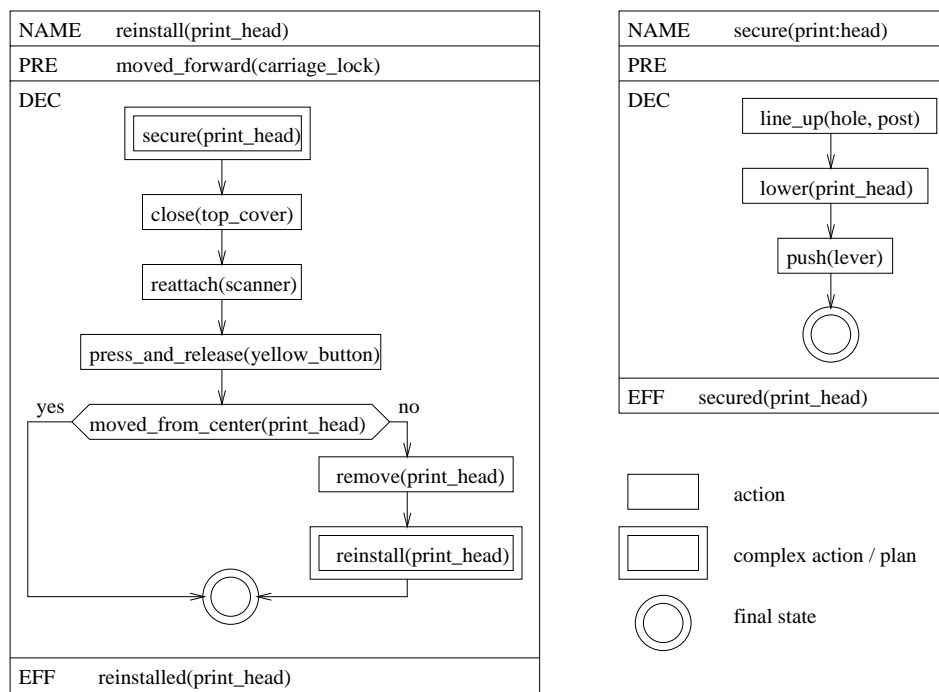- complex action / plan
- final state

Figure 4: Revised Task Plan

conditions. To reflect this we can revise the task plan as follows:

With this structure the user can control the level of detail of the instructions given. If the user does not know how to perform a substep, she can ask the system for more detailed instructions.

```
S: put the print head in place
U: how?
S: Line up the hole in the print head with
the green post on the printer carriage
U: right, ok
S: Lower the print head down gently into
position
U: ok
S: Gently push the green cartridge lock lever
up until it snaps into place
U: ok
S: The print head is now securely in place
```

On the other hand, if the user already knows how to perform a substep, the system moves on to the next step.

```
S: put the print head in place
```

```
U: done, and now?
S: Close the top cover
```

If the user manages to complete the whole action sequence without instructions, she can tell the system this and the system proceeds to the next relevant action.

```
S: put the print head in place
U: how?
S: Line up the hole in the print head with
the green post on the printer carriage
U: right, done
S: Lower the print head down gently into
position
U: done, I remember now (pause) the print
head is put in place
S: Ok. Close the top cover
```

Here, however, we see the importance of the task structure. It is only if we have information that gives the structure of the task with subtasks that we can model this. Very often instructional manuals will give this sub-structure, e.g. in the form of subdivisions of instructions, but they tend not to be com-

pletely consistent in this. It is only when this information is given in a consistent way that we can exploit it in a transformation from a written manual presentation to a more interactive presentation.

6   Discussion and Research Issues

In this experiment we have looked at a few differences that occur in the rendering of the same information under different conditions of interactivity. Our little experiment brought out several differences in the 'rendering' of the same task plan as a written text and as a minimally interactive dialogue.

- Conditionals and preconditions are handled differently if limited confirmations are possible.

- The flexibility of access that written text allows needs to be modeled more explicitly in case of aural presentation. This can be done minimally by allowing the machine to interpret 'done' or 'don't understand' as moves that lead to the presentation of the next instruction or to a repetition of the latest instruction.

Moreover the granularity with which the task plan is represented corresponds to the granularity of the control the user has over the presentations of the instructions. In this example we started from an existing manual text. Starting from a written manual helped us understand the importance of the information about the task structure. This comes of course not as a surprise: when the presentation mode is fixed as non-interactive, the the discourse structure can be very 'flat': things need to be done in a certain order whether they are parts of subtasks or not is not relevant. It can be argued that giving more structure will help a user understand better what the instructions achieve but it will not influence the execution directly. Material that helps the user understand why she is doing something is typically given in introductory sections and not in the procedures themselves in this type of manual. But to make document transformations possible in the sense

described in the beginning, it is important to clearly separate task plans and assumptions about interactions, i.e. about how the information states get updated.[4]

Once the task plan is distinguished from the dialogue plan, assumptions about the type of interactions between participants can change the dialogue plan even when the task plan remains constant.

In practice a completely automatic transformation of a written manual into even limited dialogue is most likely not possible, although one can isolate several linguistic flags for some of the aspects we have been discussing (e.g. expressions like "make sure that..." flag preconditions). A more realistic approach would be to create a blueprint document that is marked up to allow the derivation of several different types of discourse from the beginning on. Such an enterprise would need tools such as the TRINDIKIT to model the various cases [5]

So far, we have only explored one extreme of the monologue-dialogue opposition where the interactivity stays very low. Obvious extensions are to allow the user to ask information that goes beyond the current procedure, e.g. 'where can i find the piece you mention' or 'how long does this take: i have only 1/2 hour here'. Further inquiry into the possible interactions will help us to define which information is needed and how it needs to be structured to fulfill these various needs. And of course we will never reach a system in which every user need can be anticipated but then even human beings are not that type of system.

---

[4]See (Grosz and Sidner, 1986) for a discussion of the importance of task plans in more explanatory dialogue.

[5]It would also need tools that make it easy to model the relation between the linguistic expressions used in the various renderings of the base document. One can see this task as akin to that of multilingual generation or even simple document rendering. Formal approaches used for those tasks could be adapted to such an enterprise. XML supplemented with stylesheets and schemata could be another possibility.

References

P. Bohlin, R. Cooper, E. Engdahl, and S. Larsson. 1999. Information states and dialogue move engines. In J. Alexandersson, editor, *IJCAI-99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

L. Carlson. 1983. *Dialogue Games*. D. Reidel, Dordrecht.

Jennifer Chu-Carroll and Michael K. Brown. 1998. An evidential model for tracking initiative in collaborative dialogue interactions. *User Modeling and User-Adapted Interaction, special issue on Computational Models of Mixed Initiative Interaction*, 8(3+4):215–253.

H. Clark. 1999. How do real people communicate with virtual partners? Proceedings of AAAI-99 Fall Symposium, Pshychological Models of Communication in Collaborative Systems.

J. Ginzburg. 1998. Clarifying utterances. In J. Hulstijn and A. Niholt, editors, *Proc. of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues*, pages 11–30, Enschede. Universiteit Twente, Faculteit Informatica.

B. J. Grosz and C. L. Sidner. 1986. Attention, intention, and the structure of discourse. 12(3):175–204.

Staffan Larsson and David Traum. To appear. Information state and dialogue management in the trindi dialogue move engine toolkit. *NLE Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering*.

Staffan Larsson, Alexander Berman, Johan Bos, Leif Grönqvist, Peter Ljunglöf, and David Traum. 2000. Trindikit 2.0 manual. Technical Report Deliverable D5.3 - Manual, Trindi.

D. Traum, J. Bos, R. Cooper, S. Larsson, I.Lewin, C. Matheson, and M. Poesio. 1999. A model of dialogue moves and information state revision. deliverable D2.1, TRINDI.

Jan van Kuppevelt. 1995. Discourse structure, topicality and questioning. *Journal of Linguistics*, 31:109–147.