Modelling Sub-Utterance Phenomena in Spoken Dialogue Systems

Okko Buß **David Schlangen**
Department of Linguistics
University of Potsdam, Germany
{okko|das}@ling.uni-potsdam.de

Abstract

Dialogue systems that process user contributions only in chunks bounded by silence miss opportunities for producing helpful signals, such as backchannel utterances concurrent with the ongoing utterance—to the detriment of interaction quality. We survey in some detail what we call 'sub-utterance' phenomena, which such an approach misses, and then discuss two approaches to overcoming this limitation. The first is to add a 'reactive layer' to an otherwise unchanged, utterance-based dialogue manager. The other approach, less often taken, is to make the dialogue manager itself capable of producing such reactions. To explore the viability of such an approach, we sketch a dialogue management strategy capable of working at a sub-utterance level.

1 Introduction

It is generally agreed that human language processing in dialogue proceeds continuously (i.e., not at certain points in bulk, but at all times during the contribution of the other participant) and incrementally (with new bits of information building on previous ones, forming larger units); see e.g. (Marslen-Wilson and Tyler, 1981). This is a fact that manifests itself in subtle phenomena like eye movement patterns, studied in psycholinguistics (Tanenhaus et al., 1995), but also in 'surface-phenomena' like backchannel utterances ("uhu"), studied in Conversation Analysis for example (e.g., Schegloff (1982)).

Dialogue systems, but also dialogue theories, are, in contrast, more utterance-oriented. They typically discretise dialogue into utterance-chunks, either for technical reasons (simpler segmentation boundary detection) or for theoretical ones (propositions as smallest unit).[1]

In this paper, we focus on what we label 'sub-utterance phenomena', and ask how we can equip dialogue systems with the capability to produce and understand those. We discuss two possible approaches, one where additional machinery takes care of such phenomena, and another where the dialogue manager is adapted more fundamentally. As the latter is an approach that is less represented in the literature than the former, we sketch an approach to dialogue management that promises to enable this strategy.

The remainder of the paper is structured as follows. We first survey in some detail the phenomena of interest, looking at what may be appropriate reactions to user contributions that exhibit them, and what may be internal events that might require a system to produce them. We then discuss the two approaches to modelling such phenomena and sketch our attempts at the second type of approach.

2 Sub-Utterance Phenomena

So far we have used the term "sub-utterance phenomenon" informally. To make more precise what we mean by it, we need to explain in a bit more detail the way that most current dialogue process user contributions: In such systems, the speech recogniser delivers output only once it has detected silence of a certain duration (often something between 750 and 1500ms, (Ferrer et al., 2002)). Hence, the unit for processing for later modules is a continuous stretch of user speech ending in silence. This segmentation is performed without input from later modules (like parsers of dialogue managers), and those later modules only see complete utterance units. "Sub-utterance phe-

[1] With the notable exceptions of, on the theoretical side, PTT (Poesio and Traum, 1997; Poesio and Rieser, 2010) and Dynamic Syntax (Cann et al., 2005), and on the implementational side (DeVault and Stone, 2003; Aist et al., 2007; Skantze and Schlangen, 2009).

Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue

nomena", in our use of the term, then are all phenomena that make reference to units smaller than such silence-bounded speech chunks.

Table 1 gives an illustration of the types of phenomena we are interested in here. We divide our analysis of the phenomena into what possible system reactions are to user-produced sub-utterance phenomena, and what internal system events could be that require a system to produce them.

2.1 System Reactions to User-Produced Sub-Utterance Phenomena

Hesitations The first phenomenon we look at, hesitations, or more specifically, unfilled pauses, poses a direct problem to the approach to contribution segmentation sketched above. As this approach uses silence to endpoint the user utterance, there is a danger of wrongly endpointing during a hesitation, and being left with an incomplete utterance (and confusion when the user then resumes talking). Hence, as a 'minimal' type of reaction to a hesitation we would ideally like the dialogue system to not confuse it with an utterance end, and to simply wait for the speaker to continue. A more sophisticated system could offer signals of support, such as backchannel utterances, or even cooperative replies such as suggesting words the speaker may be looking for, or even completing the utterance for her. (See for example (Clark, 1996) for a discussion of such strategies.)

A more subtle effect of hesitations, and disfluencies in general, has been discussed in the psycholinguistics literature: under certain conditions, dialogue participants seem to draw conclusions from the fact that a speaker is disfluent. When producing descriptions of objects, disfluencies can be taken as indication that an object with a non-obvious name is being described (Brennan and Schober, 2001; Bailey and Ferreira, 2007; Arnold et al., 2007); a system that can attend to sub-utterance phenomena could make use of such implications (Schlangen et al., 2009).

Backchannel Utterances Backchannel utterances are "short messages such as *yes* and *uh-huh*", which are given "without relinquishing the turn" (Yngve, 1970, p. 568). Ward and Tsukahara (2000, p.1182) give a more formal definition:

Back-channel feedback:

> (D1) responds directly to the content of an utterance of the other,
> (D2) is optional, and
> (D3) does not require acknowledgement by the other.

From these descriptions, we can directly derive what reactions of a dialogue system to user backchannel utterances should be. First, they should not be taken as an attempt by the user to take the floor (D3). This requires fast recognition of the user utterance as a backchannel (and not an interruption, discussed below). This would enable the system to simply ignore such utterances. But such a strategy would disregard observation (D1), namely that these utterances nevertheless are *reactions* to the content of the system's own ongoing utterance. It seems more appropriate, then, to take into account the role `BC` plays for *grounding*, the process of reaching a common understanding of the ongoing dialogue (Clark and Schaefer, 1989; Clark, 1996). At the very least, `BC`s signal 'continued attention', and it may be useful to represent this fact in the system (perhaps in order to draw inferences from the *absence* of such signals). If this effect is to be modelled, then timing information becomes important, in order to determine to which parts of the utterance the `BC` may be reacting.

Interruptions Concurrent speech from the user that is not classified as a `BC` should be treated as an interruption. Again, there are various degrees of sophistication possible in reactions to interruptions. A sensible default behaviour perhaps is to simply stop talking. Ideally, a system would also be informed where exactly, after which parts of the own utterance, the interruption occurred. However, an interruption need not necessarily lead to a turn-change: in certain situations, a system may be interested in trying to hold the turn and to continue talking.

Turn-Taking One of the immediately striking features of human–human dialogue is that transitions between speakers are often seamless, with very little gap or overlap ((Jaffé and Feldstein, 1970; Sacks et al., 1974; Beattie and Barnard, 1979)). Such seamless transitions can obviously not be achieved with the segmentation model sketched above, which relies on gaps to determine whether a speaker wants to release the turn. Other cues within the utterance must hence be responsible for the other being able to determine when to take the turn. (This is why we include this under "sub-utterance" phenomena here, given our

Phenomenon	Example
Hesitation (HES)	A: From Boston *uhm* on Monday.
Backchannel (BC)	A: From Boston on Monday. B: Mhm
Interruption (INT)	A: From Boston on- B: Sorry, Boston airport is closed!
Turn-Taking (TT)	A: From Boston. B: Erm, hang on, I'll check.
Relevant Non-Linguistic Act (RNLA)	A: From Boston on Monday Sys: [Boston lights up on map]

Table 1: Examples of Sub-Utterance Phenomena in Dialogue

endpointing-based definition of utterance.) There is a rich literature on what exactly the nature of these signals might be, syntax, semantics / pragmatics or prosody (see, *inter alia*, (Ferrer et al., 2002; Ford and Thompson, 1996; Caspers, 2003; Koiso et al., 1998; Sato et al., 2002; Ferrer et al., 2003; Schlangen, 2006)). Assuming that our system could detect such signals, what would be the appropriate reaction? Looking at human–human dialogue, it seems that even in cases where a contentful reply isn't immediately ready, it is a good strategy to acknowledge the obligation to take the turn by producing non-committal "hedges" such as *erm* (Norrick, 2009), as in the TT example in Table 1.

Relevant Non-Linguistic Actions So far, we have restricted the discussion to *linguistic* reactions. In situations where other modalities are available (that is, in face-to-face settings), it can also be appropriate to react non-linguistically to ongoing utterances. An example for such a reaction is shown in Table 1. We will discuss more examples below in Section 4. (One could also subsume under this heading non-*verbal* actions expression functions listed above, such as head-nods as backchannel signals. We however restrict this category here to actions that are non-linguistic in a wider sense.)

With regards to their immediate discourse effect, RNLA are related to (one aspect of) backchannels: they indicate some degree of understanding of the ongoing utterance. In fact, they give much deeper evidence of understanding than BCs, in that they *display* what this understanding is (Clark and Schaefer, 1989). A system capable of registering a user's RNLA should then use them to check whether the displayed understanding is congruent with the intended effect of the utterance-so-far. If it is, that part of the utterance can be taken as understood; if not, corrective measures can be taken, such as providing more information or directly correcting the user's understanding.

2.2 System Conditions Triggering Production of Sub-Utterance Phenomena

We now turn to a discussion of the conditions under which a system might want to produce such phenomena itself.

Hesitations Hesitations in human speech are normally seen as reflecting planning problems (Levelt, 1989; Clark and Fox Tree, 2002), for example due to problems with lexical access. Given current language generation architectures (Theune, 2003) and how they differ to human language generation (for example, with lexical access as an error-free database look-up, no incremental formulation, etc), there doesn't seem to be a natural reason for dialogue systems to produce hesitations. It is an interesting, but to our knowledge unresearched question whether *simulating* such problems could have interactional benefits—one could speculate that inferences from the fact that production is disfluent (as mentioned above; a disfluent description might be of a hard-to-name object) could be usefully triggered in this way.

Backchannel Utterances The situation is different for backchannel utterances. Systems working in more conversational settings may well profit from being able to produce backchannel utterances. Mirroring what was said above about the interpretation of BCs, their production should ideally reflect a desire to signal the grounding status (as 'acoustically perceived') of material concurrently to the continuation of the utterance. Alternatively, one could tie the production of BCs closer to the user's utterance, assuming that there

are indeed 'backchannel-inviting cues' (Ward and Tsukahara, 2000; Gravano and Hirschberg, 2009); we discuss this approach below.

Interruptions A system that is continuously monitoring the user's input may want to interrupt for the same reasons that a human might do so: to be able to immediately address some parts of the utterance (for example challenging its truth), or to make a choice in a long list of alternatives. Another reason for interrupting the user can be that some other event occurs that requires notification of the user with high priority; this could happen in applications where the system controls and monitors some real-world objects (Boye et al., 2000; Lemon et al., 2002). As such interruptions are thematically unrelated to the user utterance, they can be performed without continuous understanding of the user utterance.

Turn-Taking Ideally, a dialogue system would plan its utterances in such a way that turn-endings can be projected easily by the user. As discussed above, there is a variety of candidate cues that may be appropriate here; one that may be in reach is the variation of prosodic structure (e.g., using rising pitch to indicate non-finality; see e.g. (Caspers, 2003)). Moreover, devices for preparing for longer turns could be used ("Let me list the possible options.").

Relevant Non-Linguistic Actions Together with BCs, RNLAs form the class of production behaviours that seem most promising for systems in the near-term. If a modality other than speech is available, it may be advantageous to use it for displaying understanding. We will discuss examples below.

3 Two Approaches

Having surveyed the phenomena, we now turn to describing two possible directions for changes to the current dialogue system architecture model, which relies on full-utterance-based processing. Our question will be to what extent these changes bring the phenomena into the reach of the dialogue systems—with the underlying assumption that making systems capable of handling these phenomena will make them more natural (Ward et al., 2005; Edlund et al., 2008). Before we do so, we need to say just a little bit more about one component of dialogue systems, the dialogue manager. We define this as that component (or collection of components) which a) computes appropriate updates to the context given the new incoming material (which in traditional systems is the endpointed utterance in one block), and b) computes the system reaction to this update. We can then characterise the two directions according to what they modify. In the *reactive approach*, the dialogue manager still works on full utterances only, but does not remain the only place that computes reactions; rather, a further component is added that works on sub-utterance information and controls some system actions. In the *incremental DM approach*, the dialogue manager remains the only module responsible for computing system behaviour, but the basis on which it does so is changed, by allowing updates triggered by units smaller than the utterance.

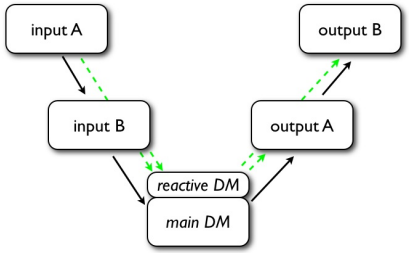


Figure 1: Schematic view of the architectural choices

Figure 1 gives a schematic view of the two possibilities: in a system with a reactive layer, continuous information flows (along the dashed lines) from "input-side" modules (the boxes on the left; let's say speech recognition and natural language understanding module) to a separate reactive DM module. This module can decide independently on the need for 'reactive behaviour', e.g. BCs. Meanwhile, the usual utterance-sized information flows along the normal channel (the solid lines), from the input modules into the dialogue manager, which computes system reactions concerning the "official business" of the dialogue as in normal systems. In a fully incremental system, on the other hand, continuous information can travel along the normal pathway, and the main DM itself is reactive enough to decide on reactions like BCs.

3.1 Keeping the Dialogue Manager, Adding a Reactive Layer

The strategy of adding a 'reactive layer' to an otherwise largely unchanged dialogue system architecture is the more prominent in the literature;

despite differences in detail, (Thórisson, 2002; Lemon et al., 2003; Raux and Eskenazi, 2007) can all be categorised in this way. It is clearly an attractive strategy, as it allows one to keep tried-and-tested traditional dialogue management paradigms; we will explore here to what extent it can cover the behaviours listed in the previous section.

Hesitations We have explained above that hesitations pose a big problem for systems that rely on silence thresholding for determining the end of a user contribution. A system with a reactive layer that has continuous access to more information than just voice activity can improve on this. Raux and Eskenazi (2008) describe such a system, where continuous information from a voice activity detector is combined with continuous information from a language understanding component that works on hypotheses of what was said so far. Using a simple proxy for detecting semantic completeness ("are expected slots already filled?"), their system can classify silences and use optimised thresholds, overall improving the latency of the replies. Note that the architectural changes only concern the additional layer; once the endpointing decision is made, all further updates and computations of system reactions are made by the unchanged dialogue manager.

Backchannel Utterances We've described above as one possible reaction to a user backchannel utterance to simply ignore them. For this, a system with a reactive layer would need the capability to quickly classify incoming audio from the user during a system utterance as backchannel or genuine interruption. A backchannel utterance could then simply be withheld from the rest of the system. (Note that this strategy entails that none of the discourse effects of BCs described above can be modelled.) We are not aware of any implemented system that makes use of such a strategy.

On the production side, a reactive layer could decide to produce a backchannel utterance in response to so called "backchannel-inviting cues" (Ward and Tsukahara, 2000; Gravano and Hirschberg, 2009), which are prosodic and lexical features of the utterance. (Their presence could presumably be detected on the basis of continuous information comparable to what was described in the previous section.) (Beskow et al., 2009) have shown that it is indeed possible, at least for short whiles, to plausibly accompany user speech with BCs produced as reaction to such cues. However, there is a danger in decoupling BCs from actual grounding state. The reactive layer and the main dialogue manager can get out of sync, as illustrated in (1) (constructed; system is B), a situation where the reactive layer produced BCs and hence signalled at least some form of understanding, which however isn't backed up by the main dialogue manager, which requests clarification. This shows that some form of synchronisation is needed in such an architecture, if behaviours produced by the reactive layer commit the system publicly to a certain discourse state.

(1)
A: Take the green block
B: uhu
A: and place i:t in the
B: yeah?
A: middle of the board.
B: OK.
B: I'm sorry, what did you say?

Interruptions, Turn-Taking From the perspective of a system with a reactive layer, these two phenomena are flip-sides of being able to deal with BCs and HES, respectively: if those phenomena are classified correctly, appropriate reactions to these phenomena can be made as well. For INT, that could be to stop talking (but again presumably losing information about what did get said), for TT this would be to start talking, or, if nothing is prepared, to produce a turn-initial non-committal signal like "erm".

Relevant Non-Linguistic Actions This seems to be an area where just adding a reactive layer cannot help. There must be a way to compute the relevance of such an action to what was already said, and for this, it seems, proper context updates have to be performed on such partial inputs. This is what the kind of architecture we turn to next promises to be able to do.

3.2 Incrementalising the Dialogue Manager

With some reflection, it should be clear that a system with an incremental dialogue manager (perhaps fed with more than just output of what is typically the previous module, NLU) can do at least that what was described above for the reactive-layer approach, as it is a proper superset. As such an approach has, to our knowledge, not been described in the literature, the more interesting ques-

tion is whether something like this is actually practically possible. (There is important prior work: (Allen et al., 2001) describe a general architecture for dialogue systems that somewhat falls under this heading; however, the focus there is more on architecture and no general DM strategy is described. Similarly, (Skantze and Schlangen, 2009) describe an implemented system that to some extent realises incremental DM, but again, the DM strategy is not the focus of that paper.)

We hence will not go through the list of phenomena again but instead devote the remaining space to sketching what a plausible incremental dialogue management approach could look like. Before we turn to this, we should note that a precondition of such an approach is that the modules feeding into the dialogue manager also work incrementally; recent work suggests that this precondition can be met (incremental ASR (Baumann et al., 2009); incremental NLU (Atterer and Schlangen, 2009; Schlangen et al., 2009; Atterer et al., 2009); generation (Kilger and Finkler, 1995; Otsuka and Purver, 2003)).

4 Sketch of an Incremental Dialogue Manager

For concreteness, we set ourselves the task here to model BC and RNLA behaviour with an incremental dialogue manager. What is needed to achieve this? First, a context representation that can be updated with partial information and that tracks grounding state (and how it is influenced by performing BCs and also RNLAs), and second, rules that compute when to perform these behaviours.

Perhaps surprisingly, it turns out that not very many deep conceptual changes are needed to get this from extant dialogue modelling paradigms. (2) shows the structure of the plan of an information state update-based system to provide ticket price information (from (Larsson, 2002, p.52)).

(2)
ISSUE: $?x.\textbf{price}(x)$
PLAN: $\langle$
findout($?x.\textbf{means\_of\_transport}(x)$),
findout($?x.\textbf{dest\_city}(x)$),
findout($?x.\textbf{depart\_city}(x)$),
findout($?x.\textbf{depart\_month}(x)$),
findout($?x.\textbf{depart\_day}(x)$),
findout($?x.\textbf{depart\_class}(x)$),
consultDB($?x.\textbf{price}(x)$)
$\rangle$

The items in this plan are questions that the system must get answered in order to handle the issue of providing price information. Raising them in the form of a sequence of questions, however, is what leads to the typical slightly rigid structure characteristic of dialogues with enquiry-based systems ("how do you want to travel?", "Where do you want to go?", "When?", etc. etc.). The assignment of each bit of necessary information to a separate question, and hence a separate user reply, appears somewhat unnatural; and this is indeed is reflected by the often made observation that human users tend to react to such restricted questions with what in the field of dialogue system design is called *overanswering*, that is by providing more information than the question taken on its own asks for (McTear, 2004).

If we remove the direct connection between the questions in the plan and expected utterances, and allow the user to address more than one of those question within a single utterance, and without them having been raised explicitly, we have made a first step towards an incremental dialogue manager. We then also note that the utterance bits answering individual questions (e.g. "*I want to fly...*", "*...from Amsterdam...*", "*...on Monday*") seem like good candidates for chunks that can be acknowledged by BCs.

Figure 2 shows an example of the information state format used in a system we are currently building.[2] The domain of the system is constructing a puzzle, where the user controls the computer to select and move around pieces, getting immediate visual feedback. The structure shown is our equivalent of a Question-Under-Discussion stack (Ginzburg, 1996), and is to be read as follows. In angle brackets, all information is grouped together that the system needs to collect to perform one domain action. Here, we have the actions take and delete; the curly brackets indicate that they are alternatives. I.e., the system needs to collect information about which action to perform, and about which tile to perform it on. (Here, both actions have the same parameter, but that is just a coincidence). After the first semicolon, we specify what an appropriate RNLA is when the information chunk specified in this line has been provided. E.g., once we know that the action to perform is

[2] We should note that we are in the early stages of the realisation of the system. While first experiments indicate that the concepts sketched here should work, the devil will, as always, be in the details of the implementation.

```
{< a ( 1 action=A=take; 2 prepare(A)   ; 3 U),
     ( 4 tile=T       ; 5 highlight(T) ; 6 U),
     ( 7              ; 8 execute(A,T) ; 9 U) >

 < b (10 action=A=del ;11 prepare(A)   ;12 U),
     (13 tile=T       ;14 highlight(T) ;15 U),
     (16              ;17 execute(A,T) ;18 U) >}
```

Figure 2: Example Information State

take, we can prepare for this action. The last column in each row records the resolution/grounding state: has this question / bit of information been resolved / provided? Has the provided value been grounded with the user? Etc. The last line finally records what to do when all bits of information have been collected.

The idea now is that users can provide this information (initially) unprompted and within one (or more) utterance(s), and that it is "struck out" immediately once provided. Additionally, both BC and RNLA feedback can be given during the utterance. In the appendix, we give two worked examples that illustrate some nice consequences of this setup: replies to RNLAs mid-utterance (e.g., after highlighting a piece, a "right" and then a continuation of the utterance), and delivery in installments with trial intonation (Clark, 1996) can be modelled.

5 Conclusion

We have surveyed what we call "sub-utterance phenomena", that is, phenomena that require processing in a dialogue system of units smaller than full utterances. We have discussed two possible approaches to such processing, namely either providing a parallel structure that takes care of some reactive behaviours, or else making the system as a whole more responsive and able to process small units of user input. We should stress that we do not necessarily favour one approach over the other. If the emphasis is on keeping a legacy dialogue model, then adding a reactive layer is a good way to increase reactivity. If however the emphasis is on full semantic modelling, we think that an incremental dialogue manager may have some advantages—despite being a paradigm that clearly needs more work to be fully understood.

Acknowledgements
This work was funded by DFG grant SCHL845/3-1, Emmy Noether Programme.

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K. Tanenhaus. 2007. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proceedings of Decalog (Semdial 2007)*, Trento, Italy.

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *Proceedings of the conference on intelligent user interfaces*, Santa Fe, USA, June.

Jennifer E. Arnold, Carla L. Hudson Kam, and Michael K. Tanenhaus. 2007. If you say *Thee uh* you are describing something hard: The on-line attribution of disfluency during reference comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 33(5):914–930.

Michaela Atterer and David Schlangen. 2009. RUBISC – a robust unification-based incremental semantic chunker. In *Proceedings of SRSL 2009*, Athens, Greece, March.

Michaela Atterer, Timo Baumann, and David Schlangen. 2009. No sooner said than done? testing incrementality of semantic interpretations of spontaneous speech. In *Proceedings of Interspeech 2009*, Brighton, UK, September.

Karl G D Bailey and Fernanda Ferreira. 2007. The processing of filled pause disfluencies in the visual world. In R. P. G. van Gompel, M. H. Fischer, W. S. Murray, and R. l. Hill, editors, *Eye Movements: A Window on Mind and Brain*, pages 485–500. Elsevier.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009. Assessing and improving the performance of speech recognition for incremental systems. In *Proceedings of NAACL-HLT 2009*, Boulder, Colorado, USA, May.

G. W. Beattie and P. J. Barnard. 1979. The temporal structure of natural telephone conversations. *Linguistics*, 17:213–229.

Jonas Beskow, Rolf Carlson, Jens Edlund, Björn Granström, Mattias Heldner, Anna Hjalmarsson, and Gabriel Skantze. 2009. Multimodal interaction control. In Alexander Waibel and Rainer Stiefelhagen, editors, *Computers in the Human Interaction Loop*, pages 143–158. Springer, Berlin, Germany.

Johan Boye, Beth Ann Hockey, and Manny Rayner. 2000. Asynchronous dialogue management: Two case-studies. In *Proceedings of the 4th Workshop on Semantics and Pragmatics of Dialogue (Götalog2000)*, pages 51–55, Gothenburg, Sweden.

Susan E. Brennan and Michael F. Schober. 2001. How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44:274–296.

Ronnie Cann, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language*. Elsevier, Amsterdam, The Netherlands.

Johanneke Caspers. 2003. Local speech melody as a limiting factor in the turn-taking system in dutch. *Journal of Phonetics*, 31:251–276.

Herbert H. Clark and Jean E. Fox Tree. 2002. Using *uh* and *um* in spontaneous speaking. *Cognition*, 84:73–111.

Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge.

David DeVault and Matthew Stone. 2003. Domain inference in incremental interpretation. In *Proceedings of ICOS*

4: Workshop on Inference in Computational Semantics, Nancy, France, September. INRIA Lorraine.

Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. 2008. Towards human-like spoken dialogue systems. *Speech Communication*, 50:630–645.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2002. Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody. In *Proceedings of ICSLP2002*, Denver, USA, September.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2003. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *Proceedings of ICASSP-2003*, Hong Kong, China.

Cecilia E. Ford and Sandra A. Thompson. 1996. Interactional units in conversation: Syntactic, intonational, and pragmatic resources for the management of turns. In E. Ochs, E.A. Schegloff, and S.A. Thompson, editors, *Interaction and Grammar*, pages 134–184. CUP, Cambridge, UK.

Jonathan Ginzburg. 1996. Interrogatives: Questions, facts and dialogue. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

Agustín Gravano and Julia Hirschberg. 2009. Backchannel-inviting cues in task-oriented dialogue. In *Proceedings of Interspeech 2009*, pages 1019–1022, Brighton, UK, September.

Joseph Jaffé and Stanley Feldstein. 1970. *Rhythms of Dialogue*. Academic Press, New York, NY, USA.

Anne Kilger and Wolfgang Finkler. 1995. Incremental generation for real-time applications. Technical Report RR-95-11, DFKI, Saarbrücken, Germany.

Hanae Koiso, Yasuo Horiuchi, Syun Tutiya, Akira Ichikawa, and Yasuharu Den. 1998. An analysis of turn-taking and backchannels based on prosodic and syntactic features in japanese map-task dialogs. *Language and Speech*, 41(3–4):295–321.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University, Göteborg, Sweden.

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. 2002. Multi-tasking and collaborative activities in dialogue systems. In *Proceedings of SIGDIAL 2002*, Philadelphia, USA, July.

Oliver Lemon, Lawrence Cavedon, and Barbara Kelly. 2003. Managing dialogue interaction: A multi-layered approach. In Alexander Rudnicky, editor, *Proceedings of SIGdial 2003*, pages 168–177, Sapporo, Japan, July.

Willem J.M. Levelt. 1989. *Speaking*. MIT Press, Cambridge, USA.

W. D. Marslen-Wilson and L. K. Tyler. 1981. Central processes in speech understanding. *Philosophical Transactions of the Royal Society London*, B295:317–332.

Michael F. McTear. 2004. *Spoken Dialogue Technology*. Springer Verlag, London, Berlin.

Neal R. Norrick. 2009. Interjections as pragmatic markers. *Journal of Pragmatics*, 41(5):866–891.

Masayuki Otsuka and Matthew Purver. 2003. Incremental generation by incremental parsing. In *Proceedings of the student conference "Computational Linguistics in the UK"*, Edinburgh, UK, January.

Massimo Poesio and Hannes Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1(1):1–89.

Massimo Poesio and David Traum. 1997. Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

Antoine Raux and Maxine Eskenazi. 2007. A multi-layer architecture for semi-synchronous event-driven dialogue management. In *Proceedings of ASRU 2007*, Kyoto, Japan.

Antoine Raux and Maxine Eskenazi. 2008. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *Proceedings of the 9th SIGdial Workshop in Discourse and Dialogue*, Columbus, Ohio, USA.

H. Sacks, E. A. Schegloff, and G. A. Jefferson. 1974. A simplest systematic for the organization of turn-taking in conversation. *Language*, 50:735–996.

Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. 2002. Learning decision trees to determine turn-taking by spoken dialogue systems. In *Proceedings of ICSLP-2002*, pages 861–864, Denver, USA, September.

Emanuel A. Schegloff. 1982. Discourse as an interactional achievement: Some uses of 'uh huh' and other things that come between sentences. In Deborah Tannen, editor, *Analyzing Discourse: Text and Talk*, pages 71–93. Georgetown University Press, Washington, D.C., USA.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *Proceedings of SIGdial 2009*, London, UK, September.

David Schlangen. 2006. From reaction to prediction: Experiments with computational models of turn-taking. In *Proceedings of Interspeech 2006, Panel on Prosody of Dialogue Acts and Turn-Taking*, Pittsburgh, USA, September.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of EACL 2009*, pages 745–753, Athens, Greece, March.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathllen M. Eberhard, and Julie C. Sedivy. 1995. Intergration of visual and linguistic information in spoken language comprehension. *Science*, 268.

M. Theune. 2003. Natural language generation for dialogue: system survey. Technical Report TR-CTIT-03-22, Centre for Telematics and Information Technology, University of Twente, Enschede.

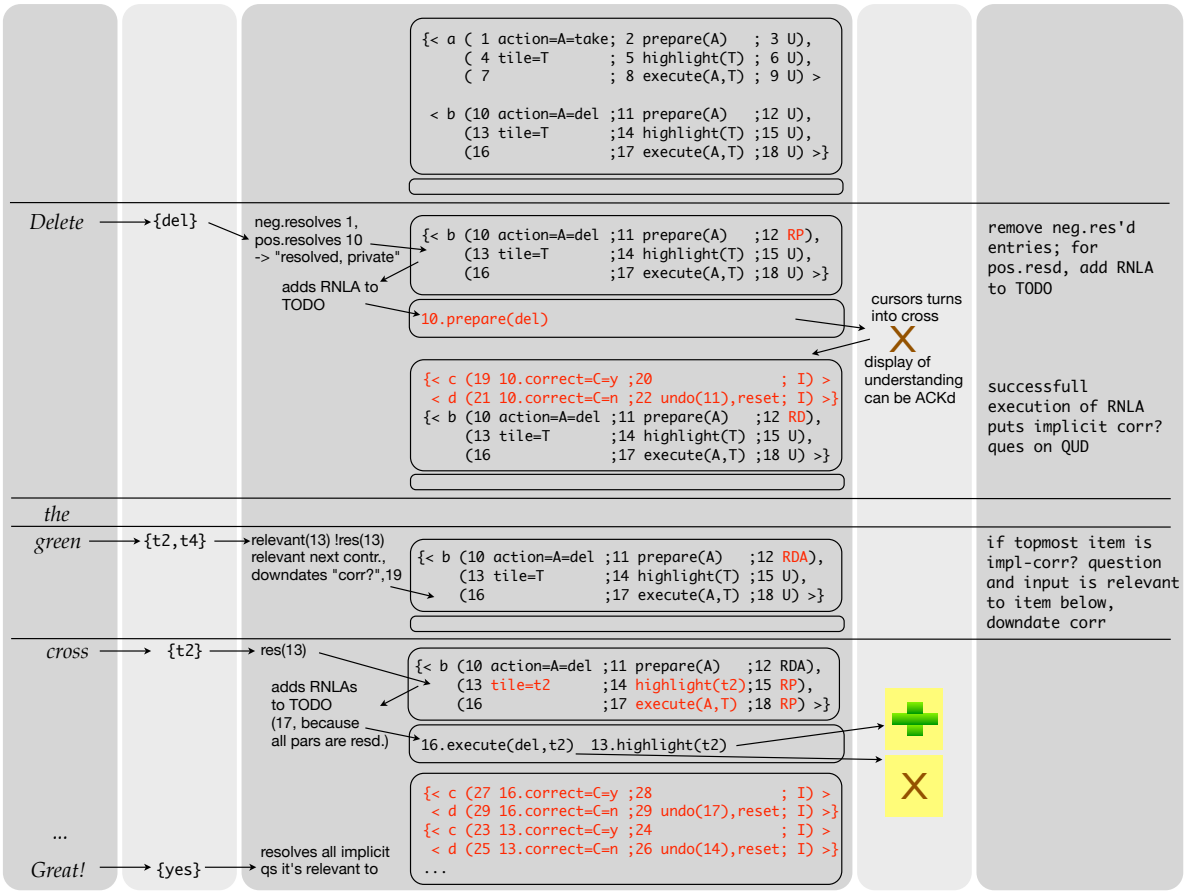
Kristinn R. Thórisson. 2002. Natural turn-taking needs no manual: computational theory and model, from perception to action. In Björn Granström, David House, and Inger Karlsson, editors, *Multimodality in Language and Speech Systems*, pages 173–207. Kluwer, Dordrecht, The Netherlands.

Nigel Ward and Wataru Tsukahara. 2000. Prosodic features which cue back-channel repsonses in english and japanese. *Journal of Pragmatics*, 32:1177–1207.

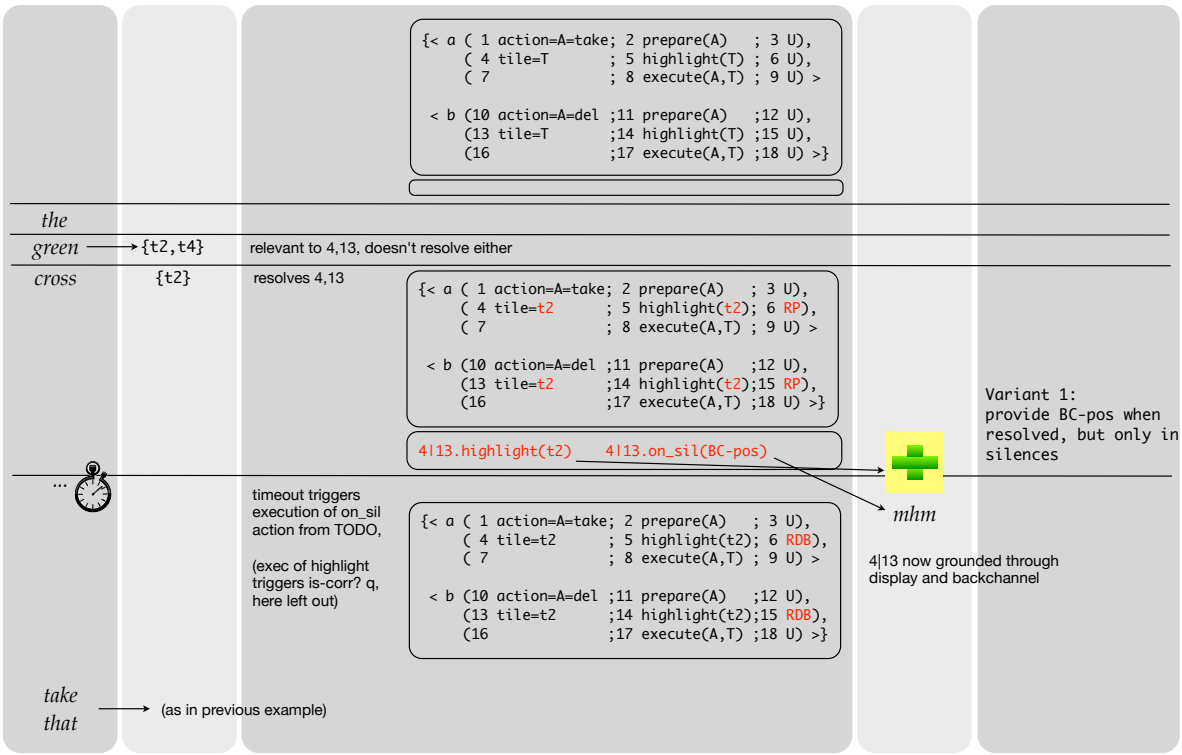
Nigel G. Ward, Anais G. Rivera, Karen Ward, and David G. Novick. 2005. Root causes of lost time and user stress in a simple dialog system. In *Proceedings of the 9th European Conference on Speech and Communication Technology (Interspeech2005)*, Lisbon, Portugal, September.

V. H. Yngve. 1970. On getting a worde in edgewise. In *Papers from the 6th Regional Meeting*, pages 567–578, Chicago, USA. Chicago Linguistics Society.

APPENDIX



Example 1. Columns are, from left to right: user utterance, semantics, updates and resulting information state (consisting of QUD and TO-DO field), system reaction, and comments. Utterance of *delete* eliminates other candidate (*take*) from QUD, triggers visible action (cursor turns into cross), which implicitly raises question "was this correct?". Question is answered by relevant continuation ("the green. . . "), and hence removed.



Example 2. Columns as above. "*the green cross*" is relevant to both action alternatives (*take* and *delete*). Is uttered with rising pitch (trial intonation), which leads to short timeout, at which a confirming BC is uttered.
Resolution / grounding states shown: U, unresolved; RP, resolved, but private (not grounded); I, implicitly raised; RD, resolved, understanding displayed; RDB, resolved, displayed and BC offered.

Figure 3: Worked Examples