A System Architecture for Simulating Time-Constrained Language Production

Jean Carletta, Richard Caley, and Stephen Isard
Human Communication Research Centre
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW
Scotland
Contact email: {J.Carletta,R.Caley}@edinburgh.ac.uk
Contact phone: +44-31-650-4438
Contact fax: +44-31-650-4587

Abstract

The goal of our research is to simulate the human production of language under time constraints. In this paper, we briefly discuss two behaviours arising from time pressure, *hesitation* and *spontaneous self-repair*, which we have identified from a corpus of human dialogues. We then go on to describe a system architecture, adapted from work in real-time systems, which we intend to use in building the simulator.

When people speak under time pressure, they tend to be less fluent then they are in unpressured situations. However, much of human language is produced under the time pressure which arises naturally out of interactive situations; in dialogue, failure to begin speaking can be taken to indicate forfeiture of one's turn. There are some widespread strategies which human speakers seem to use when they encounter time pressure, but their effects, and the implications for natural language research, have largely been ignored. Human speakers often *hesitate* until they have constructed an utterance which better matches their intentions by, for instance, using terms like "erm" and "you know". They also often commit early to a particular utterance which they aren't sure accurately conveys the intended meaning, only to *spontaneously self-repair* afterwards (e.g., "Go left, sorry, I mean right at that place with all the churches — Holy Corner"). These behaviours are so common in every day dialogue that hearers usually fail even to notice that they have occurred. Hesitation and spontaneous self-repair are clearly important for research into speech recognition, since any system which is to be able to understand natural human speech must be able to recognise and interpret utterances containing them. These behaviours also provide us with some insight into the organisation of the processes underlying human language production (see especially Levelt [16] for the implications of spontaneous self-repair), since they tell us something about where these processes can be subdivided and interrupted. Moreover, we would argue that these strategies could also profitably be used to handle the time pressure which machines face in human-computer interaction, both in systems which communicate using natural language and those which use other techniques. We first give a brief analysis of hesitation and spontaneous self-repair by reference to a corpus of spoken human dialogues; we have reported more fully on this analysis elsewhere (Carletta, Caley, and Isard [6]). We then describe a system architecture adapted from the anytime algorithm work of Boddy and Dean [12, 2] which will naturally produce hesitations and spontaneous self-repairs when it is placed under time pressure. We intend to build a generator which conforms to the given architecture in order to simulate the behaviours found in the corpus of human dialogues; this work is currently in progress.

1 The Corpus Analysis

The corpus analysis uses a map navigation task first described in Brown et al. [4]. In these dialogues, two participants who are separated by a partition have slightly different versions of a simple map with approximately fifteen landmarks on it. The maps may have different landmarks or have some of the landmarks in different locations. In addition, one agent has a route drawn on the map. The task is for the second agent to duplicate the route. The HCRC Dialogue Database [1] contains 128 such dialogues involving 15 hours of speech. In half of the trials, there is a full partition, while in the others, the partition allows eye contact between the participants but no view of the partner's map. We have analysed eleven of the dialogues. We have noted information about each occasion (approximately 700 in all) when a speaker performed a spontaneous self-repair or a hesitation, excluding those examples where the partner began speaking before the beginning of the repair, since in these cases it was felt that control of the floor rather than time constraints may have caused the difficulty. Information is noted down using an adaptation of Levelt's [16] framework for describing repairs and includes information about where the speech interruption occurred, which words that occurred before the interruption were replaced by different words after the interruption, which words were resaid in order to show the hearer how to fit the two halves of the utterance together, and how the interruption was signaled to the hearer (e.g., by unusually long silence or explicit editing terms such as "uh" or "sorry"). In addition, we have noted how the intonation of the material preceding the interruption related to that following it and characterised the kinds of surface and semantic elements which differ between the material before the interruption and the final completed utterance.

1.1 Hesitation

Our corpus analysis suggests that there is a variety of techniques which agents use to take or keep the floor when they have not finished deciding what to say next. These include:

Pausing and Stretching. Agents quite often pause for a considerable time in mid-utterance but indicate intonationally that they intend to continue. In some cases, it seems quite likely that they need the extra time to construct more of the utterance. Agents can either use long pauses between clauses, or deliver words one or two at a time while they are thinking. Pausing is often combined with stretching the pronunciation of words so that they take more time.

Filler Terms and Stock Phrases. Agents can avoid long silences by filling them in with terms and phrases which do not contribute to the meaning of the utterance, such as "erm" and "you know what I mean". Fillers which occur in the middle of an utterance typically disrupt the intonation contour and often are spoken more softly than the surrounding words.

Repetition and Redundancy. Although there are also other uses of repetition (Walker [19]), often when an agent repeats parts of a previous utterance (either its own or the partner's), they seem to be signaling to the partner that they are still preparing to speak. This is especially true in cases where the speaker uses a flat intonation which does not appear to challenge the repeated utterance in any way. We would include in this category imperfect repetitions which involve minor wording changes.

1.2 Spontaneous Self-Repair

Besides producing hesitations which do not convey useful information to the partner, agents who do not have time to build their complete intended utterances can produce imprecise versions of them and then refine them later on. For instance, Clark and Wilkes-Gibb [9] report on *dummy noun phrases* (such as "thing" or "whats-it") which can be used as place holders in an otherwise complete utterance. As well as noun phrases with varying degrees of precision, we have also found variations in verb phrases and other syntactic structures:

- Yeah, the thing/ the cross that says start, yeah.
- ...until you're like level with the bottom of it/ of the actual wee drawing.
- Now go right to the other/ right-hand edge of the waterfall...
- OK right come round/ almost circle the fallen cairn...

Alternatively, agents quite often make hasty decisions in the language production process which they later decide are inappropriate for their current intentions. They can change their minds about many different levels of processing, from the entire goal of the current move down to the production of single phonemes:

- The start/ Have you got an extinct volcano?
- Have you got a fallen/ you've got a fallen cairn, haven't you?
- Go left and then down the left hand/ the RIGHT hand side of the Saxon barn...
- The outlaw's hideup/ hideout's at the top...
- Have you got cornfu-/ fields marked?

In these cases, the speakers seem to have made errors because time pressure has forced them to "jump to conclusions". It is important to note that sometimes decisions can invalidate previous choices which are usually considered to be higher level; for instance, trying to build a referring expression and realising that the partner may not share the referent can cause a change of dialogue move.

2 Anytime Algorithms for Generation

Our analysis suggests that there are two behaviours which time pressure causes for human speakers: speakers can stall for time by using non-contentful hesitations which serve to take or hold the floor, and they can jump to conclusions about the form of the utterance that they want to use by speaking before they have worked out the complete details, taking the risk that they will have to repair the utterance afterward. We intend to write a generator in the map task domain for which these behaviours arise naturally out of time pressure. The main characteristics of the generator will be that given enough time, it should be able to come up with a satisfactory utterance for the given goal and context, and that we should be able to interrupt the process of generation at any point and still make some utterance which attempts to satisfy the communicative goal; the more time is spent on generation, the better the match of utterance to intention should be. The *anytime algorithm* work of Boddy and Dean [3, 12] gives a framework within which such processes can be constructed, and we will be adapting their framework for our purposes.

2.1 A Brief Introduction to Anytime Algorithms

Boddy and Dean divide time-dependent planning into three phases: *prediction*, during which a state of the world which an agent must act to prevent is predicted, *deliberation*, during which the agent plans an action in response to what has been predicted, and *reaction*, during which the agent carries out that action. Although they characterise prediction in terms of preventing undesirable effects, it can also be characterised more positively in terms of causing desirable effects, since the absence of a desirable effect is undesirable. Boddy and Dean concentrate on what happens during deliberation, since in many of the domains in which they are interested, they can make the simplifying assumptions that prediction takes a fixed amount of time and reaction time is negligible. They define *anytime algorithms* to be ones

1. which can be suspended and resumed with negligible overhead, so that they are amenable to preemptive scheduling,
2. which can be terminated at any time and still return some answer, and
3. which return answers which improve as a function of time.

Then the main task of time-dependent planning is to schedule deliberation time for a set of competing anytime algorithms which handle subparts of the planning in a way which maximises the estimated utility of the entire solution. They then describe some simple classes of anytime algorithms, each of which requires a different way of maximising utility, and give results about how to construct schedules for them.

2.2 Adapting Anytime Algorithms for Generation

For our purposes, prediction involves the adoption of a communicative goal. Deliberation involves planning an utterance which will fulfill the communicative goal, and reaction involves formulating the utterance. Prediction in our domain is as time-pressured as deliberation because the effects of time pressure on understanding the partner's utterances plays a role in time-constrained generation; for instance, speakers who are under time pressure might realise that they have misunderstood their partners' meanings after they have already begun realising responses. However, since in order to model these effects we would have to build a time-constrained understanding system as well as a generator, we will make Boddy and Dean's simplifying assumption that prediction time is fixed and use part of the JAM dialogue system [8, 7] to decide upon the next communicative goal. Deliberation in our domain will consist of a set of anytime algorithms making decisions in a system network which define the final utterance, corresponding roughly to *message construction* in Levelt's [16] terminology. We plan to organise the anytime algorithms around specific subtasks which involve related clusters of decisions such as task planning, choosing an appropriate dialogue move and speech act, and the planning of descriptions and referring expressions. We expect these anytime algorithms to have more complex estimated utilities than those which have previously been analysed and for which it will be necessary to build a heuristic deliberation scheduler [5]. Finally, reaction will correspond to a *formulation* process which will involves reading information about the utterance off of the system network and realising it as speech. Although we do not intend for our formulation process to take negligible time, we intend to follow de Smedt and Kempen [11] in assuming that message construction and formulation occur in parallel. Boddy and Dean's reason for assuming that reaction time is negligible is to avoid complications for the deliberation scheduler; our arrangement both avoids complications and allows aspects of time-pressured behaviour to be simulated which require formulation to be interrupted mid-stream.

The need for self-repair will arise out of this architecture when an utterance must be realised without sufficient deliberation time. Our system network will be set up so that at any time during deliberation, it can be used to formulate an utterance, although not necessarily a satisfactory one. We can do this by starting off all of the decisions in the network with defaults which at least lead to some general utterance. It is interesting to note that self-repairs can generally be divided into two classes: sometimes people use an utterance which is too ill-defined to convey the intended meaning (such as "Do something"), and sometimes they get things wrong ("Go to the left, uh, right of the swamp" or "do — how far do you want me to go?"). The first class of examples arises naturally from having the early stages of generation set up a skeletal utterance which is correct but which does not convey all of the desired information; the second class of examples arises from having defaults jump to conclusions which would have, with more time, been determined to be incorrect. What kinds of decisions are made for the utterance will depend on the expected utilities of the generation anytime algorithms and the heuristics used by the deliberation scheduler; by varying the heuristics, we should be able to generate a range of differently insufficient utterances. Boddy and Dean's framework only deals with planning initial actions and not with recovering when those actions turn out to be insufficient; therefore we must add to their framework a repair generator which, given time for further deliberation after the initial time pressure has passed, can construct the self-repairs.

Generating hesitations also involves adding to Boddy and Dean's approach. We hypothesise that hesitation is an alternate strategy which agents can use when they prefer having more deliberation time to risking self-repair by using an utterance which may be insufficient or incorrect. We plan to give agents a choice between hesitation and self-repair based on estimated utility of the utterance once deliberation time has run out. Hesitations should take negligible time to construct because they either involve standard fillers or simple repetition of a previous utterance, and so this decision can be made just prior to the originally allotted deliberation time running out.

3 A Possible System Architecture

The overall architecture which we expect to use for the generator is given in figure 1. When the agent has a communicative goal, a TIME CONTROLLER mediates between deliberation and reaction by first giving time to a DELIBERATION SCHEDULER and then signaling the FORMULATOR to formulate an utterance when time is up. We expect to provide a way of changing the constraints on

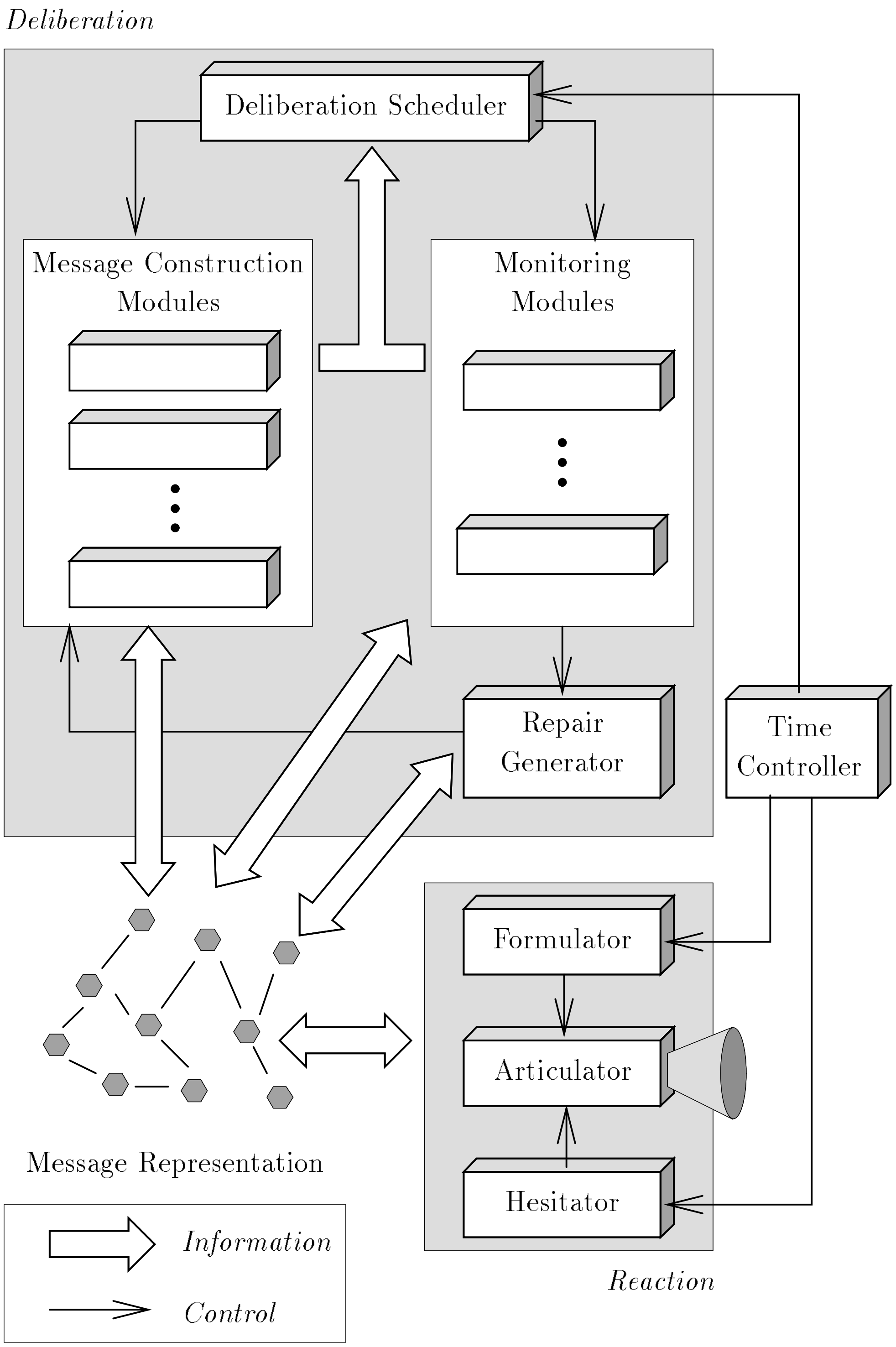

Figure 1: The System Architecture

the time controller unexpectedly so that it cuts off scheduled deliberation time, in order to simulate, for instance, moves in eye contact dialogues during which the speaker becomes pressured by seeing the partner prepare to speak before the predicted time. The deliberation scheduler, when it has been given a chunk of time, checks the estimated utilities of a set of anytime algorithm-based MESSAGE CONSTRUCTION MODULES covering different subparts of the generation process, and runs each of the different modules according to its beliefs about which will contribute the most to building a sufficient utterance. These modules work off a central MESSAGE REPRESENTATION which takes the form of a system network; different modules will make decisions encompassing all levels from deciding what kind of move to make and what the semantic content of the move should be down to lexical choice. When deliberation time runs out, the deliberation scheduler will return a value indicating whether to hesitate or to say part or all of the utterance encoded in the message representation; for instance, if the estimated utility goes up steeply with more deliberation time, then it might be best to hesitate, and if more time won't greatly improve the utterance, the system might decide to speak. If hesitation is chosen, then the time controller activates the HESITATOR, which inserts a hesitation into the speech stream (e.g., an "erm" or a repetition based on the message representation or possibly on some other working memory structure detailing past utterances, including the partner's). During the hesitation, the time controller gives the deliberation scheduler another chunk of time. If instead the system decides to speak, then the time controller sends a signal to the FORMULATOR, which then creates a surface form from the information in the MESSAGE REPRESENTATION and passes it to the ARTICULATOR. The formulator also marks which bits of the utterance have been realised in the message representation so that if a repair is needed, it can be coordinated with the faulty utterance.

Once the system has spoken, the deliberation scheduler can be scheduled more time to work on determining whether or not the utterance which was produced sufficed. It uses a set of MONITORING MODULES for this which check for different expected problems in the utterance such as whether or not there is any information required for interpretation of the utterance which the system is not sure the hearer has or whether or not a referring expression which was generated turned out to be adequate. If any of the monitors come up with a problem which is recognised to be sufficiently serious to require repair, then that monitor calls the REPAIR GENERATOR. The repair generator marks out what changes are required in the message representation in order to come up with the corrected utterance according to whatever monitors can run in the time allotted, allowing for hesitation in the same way as for the first pass of generation. When time runs out and the time controller calls the formulator again, which will sometimes be before the formulator has completed the previous utterance, it compares the old representation for what was said to the new utterance requirements and constructs an appropriate repair. This process of scheduling time and repairing the utterance can continue indefinitely until the estimated utility levels out or the agent's processing effort is required elsewhere (e.g., for interpreting the partner's utterance or for generating a subsequent utterance).

In addition to these components, we could construct a SPEECH MONITOR which works on the basis of either inner or overt speech from the articulator. Levelt [16] suggests that most (but not necessarily all) human monitoring is done at this level. For the time being our architecture maintains his position that the intermediate products of formulation are not directly accessible for monitoring, and so without a speech monitor, we could not construct repairs on the basis of, for instance, incorrect lexical or phonetic choice. We are currently primarily interested in the kinds of repairs which result from time pressure on the deliberation in the message construction component of the architecture rather than from repairs emerging from the formulator. The implications of monitoring problems with message construction via speech rather than via the message representation are unclear to us at present and will require further consideration.

4 What Comes Next

To date, we have only designed the architecture of a system which should be able to simulate the time-pressured behaviours in the map task corpus. Over the next two years, we first intend to divide up the task of generation into different modules which are solvable by anytime algorithms which have the characteristic that when they are interrupted, they will simulate the insufficient utterances which precede self-repairs. Then we intend to construct utility estimates for the anytime algorithms and build a scheduler which will call the different modules. We will then turn our attention to the monitoring modules and the repair generator. When we have a working generator, we intend to vary the utility estimates provided by the different modules so that the profile of behaviours which we produce matches the profile of behaviours in the human corpus. We then intend to investigate how time-pressured behaviours relate to several parameters which vary in the corpus, such as whether or not the dialogue participants are friends or strangers and whether or not they have eye contact with each other. We also intend to vary whether the deliberation scheduler gives each component its entire time allocation at once or interleaves smaller time slices for separate components and to show how that affects what is produced when deliberation time is cut off unexpectedly.

We intend for the message representation used by the message construction modules to be a system network such as that used in Houghton and Isard [14], but also encoding information which allows decisions about dialogue moves to be made. Although we do not yet know how to divide up the message construction modules, we expect that there will be scope for building anytime algorithm versions of existing work. For instance, we will need a way of planning referring expressions, and this might involve combining the use of dummy noun phrases (Clark and Wilkes-Gibb [9]) and something like Reiter and Dale's algorithm [18] to come up with successively better solutions. Since the JAM system (Carletta [8]) already makes high level decisions for the map task domain from the choice of move to the semantic content of the utterance, we expect to build anytime versions of JAM's decision procedures for the new generator.

In order to simulate the time-constrained behaviours in the corpus, our message construction and formulation will need to run in parallel, so that formulation can be interrupted mid-stream. De Smedt and Kempen [11] have investigated the relationship between message construction and formulation in such an arrangement, and de Smedt's incremental parallel formulator IPF has the appropriate characteristics. We will also need a way of attaching a repair to its corresponding original utterance in a way which simulates the human data. Levelt [16] suggests that a repair must retrace enough of the original utterance to allow it to be unambiguously attached syntactically. Langer [15] and Hindle [13] both discuss how to allow a parser to handle self-repaired utterances and Neumann and Finkler [17] and de Smedt [10] discuss how tactical decisions for such utterances can be made.

5 Conclusions

The system architecture described in this paper is designed to naturally bring about hesitation and self-repair when the system is placed under time pressure. Our work should be seen primarily as an attempt to explain why these dialogue phenomena occur. There is a clear need to understand what happens when a human agent hesitates or spontaneously self-repairs because these behaviours are so frequent in normal speech; speech input systems will be much more natural to use if they permit speakers to correct themselves in something like the normal way. However, an understanding of time-constrained behaviour is also useful for more mainstream generation work. The most obvious application is as part of a system, such as the one being developed at the Dundee Microcomputer Centre, which mediates between a vocally disabled person and a speech synthesiser.[1] In the Dundee system, the real difficulty is in getting the instructions from the user to the synthesiser in time to let the partner know that the user has something to say. Hesitation, partial utterances, and self-repair could be incorporated as ways of keeping the partner's attention. Analogous tactics are already in common use by programs which signal to the user that they are still operating if they notice that they have a long computation to do, using, for example, run bars or pictures of clocks. These signals serve exactly the same purpose as the hesitations from our study. One can imagine that the user might prefer to have more information involving analogues to dummy phrases which are subsequently better specified, and that in some cases the user might not mind occasionally receiving incorrect or abstract information which at least focuses the attention in the right direction. This is true of interfaces based on natural language as much as any others. Although it is clear that there are some applications for which one wouldn't want the system to generate anything but a complete and correct utterance, for the most part people don't seem to find hesitation and self-repair an obstacle to communication.

References

[1] A. H. Anderson, M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, and R. Weinert. The HCRC map task corpus. *Language and Speech*, 34(4):351–366, 1991.

[2] M. Boddy and T. Dean. Solving time-dependent planning problems. Technical Report CS-89-03, Brown University Department of Computer Science, February 1989.

[3] M. Boddy and T. Dean. Solving time-dependent planning problems. In *Proceedings of IJCAI-89*, pages 979–984, 1989.

[4] G. Brown, A. Anderson, R. C. Shillcock, and G. Yule. *Teaching Talk*. Cambridge University Press, 1984.

[5] R. Caley, J. Carletta, and S. Isard. Generating natural language under time constraints: A preliminary report. In *Proceedings of the IEEE Workshop on imprecise and approximate computation*, December 1992.

[6] J. Carletta, R. Caley, and S. Isard. The effects of time constraints on language production. *Submitted to the European Chapter of the Association of Computational Linguistics*, 1993.

[1] Alan Newell, Dundee Microcomputer Centre, personal communication.

[7] J. C. Carletta. Planning to fail, not failing to plan: Risk-taking and recovery in task-oriented dialogue. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*, volume 3, pages 896–900, July 1992.

[8] J. C. Carletta. *Risk-taking and Recovery in Task-Oriented Dialogue.* PhD thesis, Edinburgh University Department of Artificial Intelligence, 1992.

[9] H. H. Clark and D. Wilkes-Gibb. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

[10] K. de Smedt. Revisions during generation using non-destructive unification. In *Proceedings of the Third European Workshop on Natural Language Generation*, pages 63–70, 1991.

[11] K. de Smedt and G. Kempen. Incremental sentence production, self-correction and coordination. In G. Kempen, editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pages 365–376. Martinus Nijhoff, 1987.

[12] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of AAAI-88*, 1988.

[13] D. Hindle. Deterministic parsing of syntactic non-fluencies. In *Proceedings of the 21st Meeting of the Association for Computational Linguistics*, pages 123–128, 1983.

[14] G. Houghton and S. Isard. Why to speak, what to say and how to say it: Modelling language production in discourse. In *Modelling Cognition*, pages 249–267. John Wiley and Sons, 1987.

[15] H. Langer. Syntactic normalization of spontaneous speech. In *Proceedings of COLING-90*, volume 3, pages 180–183, 1990.

[16] W. J. M. Levelt. Monitoring and self-repair in speech. *Cognition*, 14:41–104, 1983.

[17] G. Neumann and W. Finkler. A head-driven approach to incremental and parallel generation of syntactic structures. Technical Report RR-91-07, DFKI, February 1991.

[18] E. Reiter and R. Dale. A fast algorithm for the generation of referring expressions. In *Proceedings of Fourteenth International Conference on Computational Linguistics (COLING-92)*, pages 232–238, 1992.

[19] M. A. Walker. A model of redundant information in dialogue: The role of resource bounds. Technical Report IRCS Report 92-95 (Dissertation Proposal), University of Pennsylvania, October 1992.