NL Generation for Virtual Humans in a Complex Social Environment

David Traum
University of Southern California
Institute for Creative Technologies
traum@ict.usc.edu

Michael Fleischman and **Eduard Hovy**
University of Southern California
Information Sciences Institute
{fleisch,hovy}@isi.edu

1 Introduction

Natural language generation is a broad field, given the wide variety of different applications for text generation. Perhaps one of the most challenging of these applications is natural language generation for spoken dialogue systems. In spoken dialogue systems, real-time throughput is required, which constrains the processing to less than a second if the system is to seem natural, especially given other processing of input and output. Thus text generation approaches which involve selecting from among many possible alternatives or involve complex calculations to determine preferences (Langkilde & Knight 1998) is not appropriate. Generation in dialogue is also somewhere in between single-shot sentence generation and generation of extended discourse. On the one hand, single short utterances must be generated because one can not predict a priori exactly how the other dialogue participant(s) will react, and subsequent generation may depend more on the input that is newly provided than any previously available information. On the other hand, dialogues generally have a coherent structure, depending on the goals and overall structure of the task that is being discussed as well as the immediately previous utterance (Grosz & Sidner 1986). Thus text-planning notions are still relevant, even if one can not count on being able to produce paragraph-level or longer utterances as pre-planned due to the interactive nature of dialogue.

Another large issue for generation systems is the nature of the input to the system. While the output is generally well defined as coherent texts (of whatever size) in the appropriate language, there is no generally agreed-upon format for the input to a generation system. Even if one were to define a standard language for this input, this would not necessarily alleviate the problem, since the conversion from existing representations to this standard language might be just as difficult as the process of generation to natural language itself. For generation within a dialogue system, there is also the question of where the "dialogue component" ends and generation begins. In dialogue systems a large part of the issue is not just *how* to convey a given meaning in natural language, but *what* meaning should be conveyed, as well as *when* this utterance should be spoken. This is to be contrasted with generation for machine translation, in which the content is already specified, and the utterances are generated one by one into the target language as the inputs are provided in the source language (e.g., (Dorr 1989)). Likewise, for story generation or instruction manual generation, the generator may decide how much to express explicitly and how to order the presentation of content, but the content itself is largely fixed by the input, and interactive issues, such as whether the next planned text could be produced or how long to wait before producing the next text are largely absent. Generation for dialogue systems must also be concerned with dialogue issues such as *turn-taking*, *grounding*, *initiative*, and collaborative notions such as obligations and joint goals. There is also an issue of "point of view" of the speaker, which may be absent from text generation systems.

In this paper, we describe a generation system for virtual humans (Rickel *et al.* 2002), in a story-based multi-character Virtual-reality training system (Swartout *et al.* 2001). Generation for these characters puts additional constraints beyond those of most dialogue systems. The language produced must be appropriate for the character's role in the interactive experience, expressing emotions as well as beliefs and goals. The characters must also be able to speak to multiple addressees, tailoring language for each. The agents also need to be able to express content using both speech and visual modalities.

In the next section, we describe the virtual world and virtual humans that use the dialogue and generation systems we have developed. In Section 3 we describe the architecture of the agent system, including how dialogue and generation processing fits in. In Section 4, we describe the dialogue representations that are used as motivation and inputs for the generation system. In Section 5 we provide more detail on aspects of the generation process, from motivation to speak through the agent speaking english text. Section 6 has some final remarks.

2 MRE

The test bed for our dialogue model is the Mission Rehearsal Exercise project at the University of Southern California's Institute for Creative Technologies. The project is exploring the integration of high-end virtual reality with Hollywood storytelling techniques to create engaging, memorable training experiences. The setting for the project is a virtual real-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2003		**00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
NL Generation for Virtual Humans in a Complex Social Environment	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California,Institute for Creative Technologies,13274 Fiji Way,Marina del Rey,CA,90292**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**8**	

ity theatre, including a visual scene projected onto an 8 foot tall screen that wraps around the viewer in a 150 degree arc (12 foot radius). Immersive audio software provides multiple tracks of spatialized sounds, played through ten speakers located around the user and two subwoofers. Within this setting, a virtual environment has been constructed representing a small village in Bosnia, complete with buildings, vehicles, and virtual characters. This environment provides an opportunity for Army personnel to gain experience in handling peacekeeping situations.

The first prototype implementation of a training scenario within this environment was completed in September 2000 (Swartout *et al.* 2001). To guide the development, a Hollywood writer, in consultation with Army training experts, created a script providing an overall story line and representative interactions between a human user (Army lieutenant) and the virtual characters. In the scenario, the lieutenant finds himself in the passenger seat of a simulated Army vehicle speeding towards the Bosnian village to help a platoon in trouble. Suddenly, he rounds a corner to find that one of his platoon's vehicles has crashed into a civilian vehicle, injuring a local boy. The boy's mother and an Army medic are hunched over him, and a sergeant approaches the lieutenant to brief him on the situation. Urgent radio calls from the other platoon, as well as occasional explosions and weapons fire from that direction, suggest that the lieutenant send his troops to help them. Emotional pleas from the boy's mother, as well as a grim assessment by the medic that the boy needs a medevac immediately, suggest that the lieutenant instead use his troops to secure a landing zone for the medevac helicopter.

While the script is fine for a canned demo, we need to go beyond it in a number of ways, allowing for variation both depending on divergent behavior of the Lieutenant trainee, as well as perhaps a director or simulation Observer/Controller. Agents must communicate in ways faithful to their emotional and intellectual assessment of the situation yet still strive to maintain the immersiveness of a well-told story.

We currently use full NL generation for two of the characters, the sergeant, and the medic, while using a variety of templates and fixed prompts for generating the lines of the other characters. Figure 1 gives an example recorded dialogue fragment from this domain, illustrating a number of characters involved in multiple conversations across multiple modalities (LT, Sgt and Medic discussing the situation face to face, LT on radio to base and eagle1-6, Sergeant shouting orders to squad leaders). The Lt utterances were spoken by one of our researchers. The Sgt and Medic utterances were generated and synthesized spontaneously by the agents playing those roles. The other characters (Base, 3rd squad leader, and platoon Eagle 2-6) were controlled by simple algorithms and used pre-recorded prompts that were triggered either by keyword utterances or signals from the simulator illustrating a number of characters involved in multiple conversations across multiple modalities (LT, Sgt and Medic discussing the situation, LT on radio to base and eagle1-6, Sergeant with squad leaders). The utterances are labelled with `conversation.turn.utterance` (if a turn consists of only a single utterance, the ",1" is omitted). Not shown in the fragment are many non-verbal behaviors which are coordinated with and part of the communication between agents. For instance, the agents look at a speaker who is part of their conversation, while otherwise they might attend to other relevant tasks. They avert gaze to keep the turn when planning speech before speaking. Speech is also accompanied by head and arm gestures. After turns 3.1 and 5.1, troops move into position as ordered, signalling understanding and acceptance of the orders.

3 Overview of Agent Architecture

The animated agents controlling the Sergeant and Medic characters are implemented in the SOAR programming language (Laird, Newell, & Rosenbloom 1987), and built on top of the STEVE agent architecture (Rickel & Johnson 1999). SOAR provides a declaratively accessible *information state*, as advocated in the Trindi project (Larsson & Traum 2000). Most processing is done by production rules that examine aspects of the information state and produce changes to it.[1] In general, all rules fire (once) whenever their left side makes a new match against aspects of the information state, and rules will fire in parallel. There are *elaboration cycles* of finding matching rules and applying their results. Part of the information state contains the current *operators*, which act as a kind of focussing mechanism, so that an agent serially attends to different functions. Rules can have their left hand side refer to aspects of the operator so that they apply only when this operator is active. There is also a higher level *decision cycle*, in which new operators are chosen. The agent can focus on sub-goals by introducing new operators as subordinate to the main goal rather than replacing the main operator.

The STEVE agents, as augmented for the MRE project, have multiple components, each involving one or more operators, sometimes also including parts that work in other operators as well. These include

- a *belief model* including knowledge of the participants and other relevant people, objects and locations of the domains, including social relationships and various relevant properties.
- a *task model*, consisting of knowledge of the events that can happen in the world as well as plans for sequencing these tasks into plans to achieve specific goal states.
- a *perception module* that receives messages from the world-simulator and other agents and updates the agent's internal state, mediated by notions of foveal attention.
- an *emotional model*, including basic emotional states, appraisal of emotional state, and coping mechanisms to influence behavior (including mental actions such as adopting goals) on the basis of emotion (Gratch & Marsella 2001; Marsella & Gratch 2002).
- a *body control* module that adjusts body position, gaze, and gesture as appropriate for engaging and monitoring tasks as well as involvement in face to face conversation.

[1] SOAR also has access to functions in the tcl programming language, when needed.

1.1	LT	what happened here?
1.2.1	SGT	there was an accident sir
1.2.2	SGT	this woman and her son came from the side street and our driver didnt see them
1.3	LT	who's hurt?
1.4	SGT	the boy and one of our drivers
1.5	LT	how bad is he hurt?
1.6	SGT	the boy or the driver
1.7	LT	the driver
1.8	SGT	the driver has minor injuries sir
1.9	LT	how is the boy?
1.10	SGT	Tucci?
1.11.1	MEDIC	sir he is losing consciousness
1.11.2	MEDIC	we need to get a MedEvac in here ASAP
1.12.1	LT	understood
1.12.2	LT	Sergeant where is the medevac
1.13	SGT	the MedEvac is at the base sir
2.1	LT	eagle base this is eagle two six over
2.2	BASE	Eagle two six this is eagle base over
2.3	LT	requesting medevac for injured civilian over
2.4.1	BASE	Standby.
2.4.2	BASE	Eagle two six this is eagle base
2.4.3	BASE	medevac launching from operating base alicia time now
2.4.4	BASE	eta your location zero three
2.4.5	BASE	over
2.5	LT	roger eagle base two six out
1.14	LT	sergeant secure a landing zone
1.15	SGT	sir first we should secure thee assembly area
1.16	LT	secure the assembly area
1.17	SGT	understood sir
3.1.1	SGT	Squad leaders listen up
3.1.2	SGT	give me three sixty degree security here
3.1.3	SGT	First Squad take twelve to four
3.1.4	SGT	Second Squad take four to eight
3.1.5	SGT	Third Squad take eight to twelve
3.1.6	SGT	Fourth Squad secure thee accident site
4.1.1	SGT	Johnson
4.1.2	SGT	send a fire team to the square to secure an LZ
4.2	3SLDR	Yes sergeant
5.1.1	3SLDR	Sergeant Duran
5.1.2	3SLDR	get your team up to the square and secure an lz
6.1.1	1-6	Eagle two six this is one six
6.1.2	1-6	whats your ETA over
6.2.1	LT	one six this is two six
6.2.2	LT	ETA 45 minutes over
6.3.1	1-6	two six it's urgent you get here right now
6.3.2	1-6	situation's getting critical
6.3.3	1-6	We're taking fire
6.3.4	1-6	over
6.4.1	LT	Roger 1-6
6.4.2	LT	2-6 out
1.18	LT	sergeant send two squads forward
1.19.1	SGT	sir that's a bad idea
1.19.2	SGT	we shouldn't split our forces
1.19.3	SGT	instead we should send one squad to reconn forward
1.20	LT	send fourth squad to recon forward

Figure 1: An MRE dialogue interaction fragment

- a *dialogue* module, maintaining the dialogue layers described in the previous section.
- a *generation* module, to produce natural language utterances for the agents to say.
- an *action selection* module that decides which operators should be invoked at which times.

In addition to these core agent functions implemented within SOAR, the agent also relies on external speech recognition and semantic parsing modules, which send messages to the agent (which are interpreted by the perception module), and speech synthesis and body rendering, that take the output body control and speech directives from the agent and produce the behaviors for human participants to see and hear.

Dialogue related behavior is performed in three SOAR operators. *Understand-speech* is invoked whenever perception detects new utterances, including results from speech recognition and parsing. The understand-speech operator includes recognition rules, which use both the input as well as the information state to decide on the set of dialogue acts that have been performed in the utterance. The same operator is used regardless of whether the input speech came from a human participant (via speech recognition and parsing), the agent itself (via the output-speech operator – see below), or from other agents (via agent messages that may also include some partially interpreted dialogue acts). The *update-dialogue-state* operator applies updates to the information state that are associated with the recognized acts.

All of the generation functions from deciding what to say to producing the speech happen in the *output-speech* operator. *Proposal rules* for this operator function as goals to speak, given various configurations of the information state. Selection of one of these proposal rules (meaning there is nothing more urgent to do or say[2]) constitutes selection of a goal for generation. Once in the operator, there are several phases including also a couple of sub-operators. First is the *content selection* phase, in which the agent reasons about how best to achieve the output goal. Examples are which assertion to make to answer a pending question, or how to respond to a negotiation proposal. Once the content has been selected, next there is a *sentence planning* phase, deciding the best way to convey this content. This is followed by a *realization* phase, in which words and phrase structures are produced. Next, a *ranking* phases considers the possibly multiple ways of realizing the sentence, and selecting a best match. This final sentence is then augmented with communicative gestures including lip synch, gaze, and hand gestures, converted to XML, and sent to the synthesizer and rendering modules to produce the speech. Meanwhile, messages are sent to other agents, letting them know what the agent is saying. The output-speech operator continues until callbacks are received from the synthesizer, letting the agent know either that the speech was completed or has been interrupted (perhaps by someone else's speech). The last part of

[2]In SOAR, one can write preference rules for operator selection, which are used by the action selection module to decide on the current operator.

the operator prepares the content for the understand-speech operator, so that other dialogue acts, beyond those that were explicitly planned can be recognized. For example, an operator might be concerned with computing and generating an answer to a previously asked question. While the main goal is to provide the answer, this utterance will also involve speech acts relating to grounding the question, taking the turn, and making an assertion.

4 Dialogue Representation

While there is no generally accepted notion of what a dialogue model should contain, there is perhaps growing consensus about how such information should be represented. Following the Trindi project (Larsson & Traum 2000), many choose to represent an *information state* that can serve as a common reference for interpretation, generation, and dialogue updates.

Depending on the type of dialogue and theory of dialogue processing, many different views of the specifics of information state and dialogue moves are possible. A complex environment such as the MRE situation presented in the previous section obviously requires a fairly elaborate information state to achieve fairly general performance within such a domain. We try to manage this complexity by partitioning the information state and dialogue moves into a set of layers, each dealing with a coherent aspect of dialogue that is somewhat distinct from other aspects.

Each layer is defined by information state components, a set of relevant dialogue acts, and then several classes of rules relating the two and enabling dialogue performance:

recognition rules that decide when acts have been performed, given observations of language and non-linguistic behavior in combination with the current information state

update rules that modify the information state components with information from the recognized dialogue acts

selection rules that decide which dialogue acts the system should perform

realization rules that indicate how to perform the dialogue acts by some combination of linguistic expression (e.g., natural language generation), non-verbal communication, and other behavior.

- contact
- attention
- conversation
  - participants
  - turn
  - initiative
  - grounding
  - topic
  - rhetorical
- social commitments (obligations)
- negotiation

Figure 2: Multi-party, Multi-conversation Dialogue Layers

The layers used in the current system are summarized in Figure 2. The *contact* layer (Allwood, Nivre, & Ahlsen 1992; Clark 1996; Dillenbourg, Traum, & Schneider 1996) concerns whether and how other individuals can be accessible for communication. Modalities include visual, voice (shout, normal, whisper), and radio. The *attention* layer concerns the object or process that agents attend to (Novick 1988). Contact is a prerequisite for attention. The *Conversation* layer models the separate dialogue episodes that go on during an interaction. A conversation is a reified process entity, consisting of a number of sub-layers. Each of these layers may have a different information content for each different conversation happening at the same time. The *participants* may be active speakers, addressees, or overhearers (Clark 1996). The *turn* indicates the (active) participant with the right to communicate (using the primary channel) (Novick 1988; Traum & Hinkelman 1992). The *initiative* indicates the participant who is controlling the direction of the conversation (Walker & Whittaker 1990). The *grounding* component of a conversation tracks how information is added to the common ground of the participants (Traum 1994). The conversation structure also includes a *topic* that governs relevance, and *rhetorical* connections between individual content units. Once material is grounded, even as it still relates to the topic and rhetorical structure of an ongoing conversation, it is also added to the social fabric linking agents, which is not part of any individual conversation. This includes *social commitments* — both obligations to act or restrictions on action, as well as commitments to factual information (Traum & Allen 1994; Matheson, Poesio, & Traum 2000). There is also a *negotiation* layer, modeling how agents come to agree on these commitments (Baker 1994; Sidner 1994). More details on these layers, with a focus on how the acts can be realized using verbal and non-verbal means, can be found in (Traum & Rickel 2002).

The interface between generation and dialogue is still a difficult issue, given a lack of general agreement both on what constitutes the division of labor between those two areas, as well as no general agreement on internal representations of dialogue. Concerning the former point, in some systems, NL generation is seen as a sort of server in which meaning specifications are fed in, and NL strings are sent back, for the dialogue module to decide when to say (or stop saying). In other systems, e.g.(Allen, Ferguson, & Stent 2001; Blaylock, Allen, & Ferguson 2002), an *interaction manager* controls both generation, production, and feedback monitoring but not other dialogue functions. In starting the MRE project, we were unsure on exactly what the interface should be between these two modules, but we knew we wanted to be able to easily make more information available when it is needed and can be used. The simplest way to achieve this is to have the generation component be part of the agent itself, implemented using SOAR production rules, and having full access to the information state provided by the dialogue modules as well as task reasoning and emotion. It also allows for easy interleaving of processes, such as synchronizing gaze behavior and turn-taking with utterance planning.

5 Generation Phases

In this section, we look at each phase in the generation in a little more detail.

5.1 Operator Proposal and Selection

There are a number of output speech operator proposal rules. One basic one is to *ground* utterances by other speakers for which the agent is an addressee, giving evidence of understanding or lack of understanding. Another type of proposal rule concerns the obligation to address a request (including an information request regarding a question). Other rules involve trying to get the attention of another agent, making requests and orders, clarifying underspecified or ambiguous input, and performing repairs. There are also preference rules that arbitrate between multiple possible outputs. Preferences are given to addressing a request or question rather than merely acknowledging it, or for talking about a higher-level action rather than a sub-action. Preference is also given to reactive acts over initiative-taking acts.

5.2 Content Planning

Given a goal, there are often many ways to respond. An obligation to address a question can be met by answering the question, but also by refusing to answer, deferring the answer, or redirecting the question to another agent to answer. Even with a decision to answer, there are always multiple possible answers, including both true and false answers. Within the set of answers the agent believes to be true, there are also more or less informative answers available, depending on the assumed mental state of the interlocutor, but also based on perceptual evidence of accessible information and likely inferences or general interest. We also use the agent's emotion model to focus on which content to present given a choice of valid answers.

Likewise, for a proposal (issued as either an order or a request), the agent must decide how to respond, whether to accept, defer, reject, counterpropose, or perform other negotiation moves. For clarifications, one must decide which information to ask for.

5.3 Sentence Planning

Creation of sentence plans from content is currently a hybrid process. There is a fully general but simple sentence planner, which can produce simple sentence plans for any task or state that is in the agent's belief or task model. For more precise and non-standard realization, some sentence plans are selected rather than generated from scratch, given certain configurations of the content as well as other aspects of the information state. Finally, there is a short-cut procedure which also bypasses realization and moves directly to pre-selected prompts.

Figure 3 shows an example content specification that is input to the sentence planner. These inputs contain minimal information about the object, state or event to be described, along with references to the actors and objects involved, and values representing the speaker's emotional attitude toward each object and event. A detailed account of the emotional aspects of the generation system can be found in (Fleischman & Hovy 2002). A set of SOAR production rules expands this information into an enriched case frame structure (Figure 4) that contains more detailed information about the events and objects in the input.

```
^time past
^speech-act assert
^event   :reference collision
         :attitude  -1
^agent   :reference driver
         :attitude  +4
^patient :reference mother
         :attitude  +1
```

Figure 3: Example input to sentence planner: content annotated with speaker's attitudes toward objects and events

```
(<utterance> ^type assertion
             ^content <event>)
(<event> ^type event ^time past
         ^name collision  ^agent <agent>
         ^patient <patient> ^attitude -1)
(<agent> ^type agent ^name driver
         ^definite true ^singular true
         ^attitude +4)
(<patient> ^type patient ^name mother
           ^definite true ^singular true
           ^attitude +1)
```

Figure 4: output of sentence planning

The task of expansion involves deciding which frame is to be chosen to represent each object in the input. For example, Figure 5 shows several possible frames that could be used to represent the agent `driver`. The decision is based on the emotional expressiveness, or shade, of each semantic option. A distance is calculated, using an Information Retrieval metric, between the shade of each semantic frame representing the driver and the emotional attitude of the speaker toward the driver. The frame with the minimum distance, i.e., the frame that most accurately expresses the agent's emotional attitudes, is chosen for expansion. This is done for each of the objects associated with the event or state. Once all objects have been assigned a frame, planning is complete, and realization begins.

5.4 Realization

Realization is a highly lexicalized procedure, so tree construction begins with the selection of main verbs. Each verb in the lexicon carries with it slots for its constituents (e.g., agent, patient), as well as values representing the emotional shade that the verb casts both on the event it depicts and the constituents involved in that event.

Once the verb is chosen, its constituents form branches in a base parse tree. Production rules then recursively expand the nodes in this tree until no more nodes can be expanded. As each production rule fires, the relevant portion of the semantic frame is propagated down into the expanded nodes.

Martinez

```
(<agent> ^type agent ^name martinez
         ^job driver ^proper true
         ^singular true ^shade +5)
```

The driver

```
(<agent> ^type agent ^name martinez
         ^job driver ^definite true
         ^singular true ^shade 0)
```

A private

```
(<agent> ^type agent ^name martinez
         ^rank private ^definite false
         ^singular true ^shade -2)
```

Figure 5: Subset of possible case frame expansions for object `driver`.

Thus, every node in the tree contains a pointer to the specific aspect of the semantic frame from which it was created.

For example, in Figure 6, the NP node of "the mother" contains in it a pointer to the frame `<patient>` from Figure 4. By keeping semantic content localized in the tree, we allow the gesture and speech synthesis modules convenient access to needed semantic information.

For any given state and event, there are a number of theoretically valid realizations available in the lexicon. Instead of attempting to decide which is most appropriate at any stage, we adopt a strategy similar to that introduced by (Knight & Hatzivassiloglou 1995), which puts off the decision until realization is complete. We realize all possible valid trees that correspond to a given semantic input, and store the fully constructed trees in a forest structure. After all such trees are constructed we move on to the final stage.

5.5 Ranking

In this stage we examine all the trees in the forest structure and decide which tree will be selected and sent to the speech synthesizer. Each tree is given a rank score based upon the tree's information content and emotional quality.

The emotional quality of each tree is calculated by computing the distance between the emotional attitudes of the speaker toward each object, and the emotional shade that the realization casts on each object. Realizations that cast emotional shades on objects that are more similar to the agent's attitudes toward those objects are given higher scores.

The information content of each tree is judged simply by how much of the semantic frame input is expressed by the realization. Thus, realizations that do not explicitly mention the agent (through passivization), for example, are given lower scores.

The score of each tree is calculated by recursively summing the scores of the nodes along the frontiers of the tree, and then percolating that sum up to the next layer. Summing and percolating proceeds until the root node is given a score that is equivalent to the sum of the scores for the individual nodes of that tree. The tree with the highest root node score is then selected and passed to the speech synthesis and gesture modules.

5.6 Sequencing Issues

There are a number of issues relating to generation being part of the deliberate behavior of an agent engaged in task-oriented dialogue. The previous discussion in this section described the normal process of utterance generation, from the point at which a goal was proposed until speech was produced. There are, however several cases in which this cycle does not follow the straightforward path. First, some proposals may later be retracted. For instance, if a goal to address a request is selected in preference to a goal to acknowledge the request and fully realized, the goal to acknowledge will be dropped, since addressing will also count as (indirectly) acknowledging. Some communicative goals can not be immediately realized, for example communicating content to characters who are paying attention to the agent. In this case, one must first adopt and realize a goal to get the attention. Sometimes a goal must be dropped during the output-speech operator. For instance if the agent realizes that there is no need to say anything or doesn't know how to say what it wants to. In this case, the agent will produce a verbal disfluency (e.g., "uh") and continue on with a new realization goal. Finally, "barge-in" capability is provided by allowing the agent to back out of an existing output-speech operator in favor of attending to the speech of others. If the goal still remains after interpreting the interruption, the agent will readopt it and eventually produce the interrupted utterance. If on the other hand, the motivations for the goal no longer hold, the goal will be dropped.

Currently there is a limited facility for multi-utterance discourse plans. For certain content, such as a description of a charged event or a rejection of an order, the agent plans multiple utterances to give rationale, either assigning causality or giving explanations and counterproposals. In this case, single sentence generation is carried out, as normal, but strong motivations are set up for future utterances, which will directly follow, unless the agent is interrupted.

6 Summary

. We adopt a hybrid approach to NL generation for virtual characters in a complex interactive environment. For some simple characters that will have only a limited range of choices of what to say, we use pre-calculated prompts and simple templates. For more sophisticated characters, more complex generation techniques are needed to behave appropriately given the rich structure of social interaction and agent emotions that are being tracked. Within the agents also, multiple methods are used, including prompt generation for very specific situations, selected sentence plans for intermediate degrees of flexibility while still allowing complex utterance structure, and sentence planning for fully general coverage.

Generation covers simple cases of reactive feedback and turn management as well as complex representations of sequences of events, negotiation moves and emotional affect. At this point, we still feel that it is best to keep a fairly tight coupling between generation and dialogue functions, given

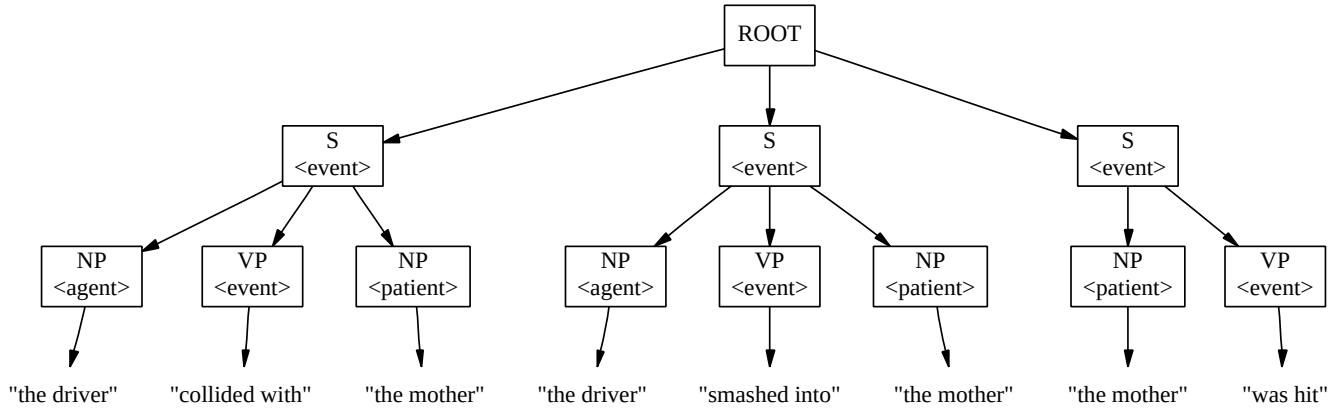

Figure 6: A subset of the forest output of realization.

the fairly broad range of fairly quickly changing information that can affect generation. The SOAR architecture is well suited for this, allowing declarative information to be made available for the use of either module as processing is continuing. In the future, we are planning a number of extensions, including information-structure based sentence planning, more elaborate discourse planning, and statistical sentence plan selection and ranking.

Acknowledgements

The work described in this paper was supported by the Department of the Army under contract number DAAD 19-99-D-0046. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Department of the Army. We would also like to thank the rest of the MRE team for providing a stimulating environment within which to carry out this research, particularly the others contributing to aspects of the system that are used here.

References

Allen, J. F.; Ferguson, G.; and Stent, A. 2001. An architecture for more realistic conversational systems. In *Intelligent User Interfaces*, 1–8.

Allwood, J.; Nivre, J.; and Ahlsen, E. 1992. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics* 9.

Baker, M. 1994. A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education* 5(2):199–254.

Blaylock, N.; Allen, J.; and Ferguson, G. 2002. Synchronization in an asynchronous agent-based architecture for dialogue systems. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, 1–10. Philadelphia: Association for Computational Linguistics.

Clark, H. H. 1996. *Using Language*. Cambridge, England: Cambridge University Press.

Dillenbourg, P.; Traum, D.; and Schneider, D. 1996. Grounding in multi-modal task-oriented collaboration. In *Proceedings of the European Conference on AI in Education*.

Dorr, B. J. 1989. Lexical Conceptual Structure and Generation in Machine Translation. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, 74–81.

Fleischman, M., and Hovy, E. 2002. Emotional variation in speech-based natural language generation. In *Proceedings of The Second International Natural Language Generation Conference (INLG'02)*.

Gratch, J., and Marsella, S. 2001. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the Fifth International Conference on Autonomous Agents*.

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

Knight, K., and Hatzivassiloglou, V. 1995. Two-Level, Many-Paths Generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, 252–260.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. SOAR: an architecture for general intelligence. *Artificial Intelligence* 33(1):1–64.

Langkilde, I., and Knight, K. 1998. Generation that Exploits Corpus-Based Statistical Knowledge. In *Proceedings of COLING-ACL '98*, 704–710.

Larsson, S., and Traum, D. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 6:323–340. Special Issue on Spoken Language Dialogue System Engineering.

Marsella, S., and Gratch, J. 2002. A step towards irrationality: using emotion to change belief. In *In proceedings of AAMAS 2002: First International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Matheson, C.; Poesio, M.; and Traum, D. 2000. Modelling grounding and discourse obligations using update rules. In

Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics.

Novick, D. 1988. *Control of Mixed-Initiative Discourse Through Meta-Locutionary Acts: A Computational Model*. Ph.D. Dissertation, University of Oregon. also available as U. Oregon Computer and Information Science Tech Report CIS-TR-88-18.

Rickel, J., and Johnson, W. L. 1999. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence* 13:343–382.

Rickel, J.; Marsella, S.; Gratch, J.; Hill, R.; Traum, D.; and Swartout, W. 2002. Toward a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems* 17.

Sidner, C. L. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of the Fourteenth National Conference of the American Association for Artificial Intelligence (AAAI-94)*, 814–819.

Swartout, W.; Hill, R.; Gratch, J.; Johnson, W.; Kyriakakis, C.; Labore, K.; Lindheim, R.; Marsella, S.; Miraglia, D.; Moore, B.; Morie, J.; Rickel, J.; Thiebaux, M.; Tuch, L.; Whitney, R.; and Douglas, J. 2001. Toward the holodeck: Integrating graphics, sound, character and story. In *Proceedings of 5th International Conference on Autonomous Agents*.

Traum, D. R., and Allen, J. F. 1994. Discourse obligations in dialogue processing. In *Proceedings of the 32^{nd} Annual Meeting of the Association for Computational Linguistics*, 1–8.

Traum, D. R., and Hinkelman, E. A. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence* 8(3):575–599. Special Issue on Non-literal language.

Traum, D. R., and Rickel, J. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first International Joint conference on Autonomous Agents and Multiagent systems*, 766–773.

Traum, D. R. 1994. *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. Dissertation, Department of Computer Science, University of Rochester. Also available as TR 545, Department of Computer Science, University of Rochester.

Walker, M. A., and Whittaker, S. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings ACL-90*, 70–78.