Toward a Model for Incremental Grounding in Dialogue Systems

Thomas Visser
thomas.visser@gmail.com

University of Twente, EEMCS, HMI
Enschede, The Netherlands

Prof. dr. D.K.J. Heylen
Dr. ir. H.J.A. op den Akker
Dr. M. Theune

USC Institute for Creative Technologies
Playa del Rey, CA, USA

Dr. D.R. Traum

So

Yes

what

Go on

actually

Hm-hmm

is

Ah

incremental

Okay

grounding?

Well?

Contents

Acknowledgments

The title of this section is not to be confused with the kind of acknowledgments that play a major role in the rest of this thesis. This is not a preamble containing the definition of that important concept. This is where I say thanks.

First and foremost, I would like to thank my supervisors David, Dirk, Rieks and Mariët for their academic guidance, valuable feedback and moral support. Thanks to Dirk for introducing me to David and allowing me to pursue a direction of research that I set out for with my bachelor's thesis. Thanks to Rieks for introducing me to the topic of grounding back when you supervised my bachelor's thesis. Thank you, Mariët, for supervising my first research project in natural language processing even before that. Thanks to David for inviting me to the Institute for Creative Technologies and making me experience life in academics and Los Angeles. I am also very thankful for the great support that David DeVault has offered me, the pleasant conversation and his help with getting the right input for the implementation of my model.

I have a lot of people to thank for my pleasant stay in Los Angeles and at the Institute for Creative Technologies. Thank you Stefan, Derya, Jill, Alesia, Elnaz, Giotta, Antionne, Abe, Shannon, Rudy, Coen, Trace and the others.

I am also indebted to my friends and family for taking an interest and supporting my work. Thank you Jasper, Jorrit, Johan, Sanne, Marja, Johan, Marthe and Hannah.

If you are still looking for a definition of the kind of acknowledgments that support grounding in discourse, please read along.

Thomas Visser
December 3, 2012
Nijmegen

Abstract

Spoken dialogue systems have until recently upheld the simplifying assumption that the conversation between the user and the system occurs in a strict turn-by-turn fashion. In order to have more human-like, fluent conversations with computers, a new generation of spoken dialogue systems has arisen that is capable of processing the user's speech in an incremental way. As the user is speaking, the automated speech recognizer

We have studied the AMI Meeting Corpus in order to identify ways of grounding in human-human dialogue that a system would be able to pick up using incremental processing. These incremental grounding behaviors include overlapping feedback, the completion of unfinished utterances that were initiated by the other party and responding to an utterance before it is completed.

We have developed an incremental grounding model that supports those incremental grounding behaviors. The input of the model consists of incremental hypotheses of the explicit and predicted content of the utterance, i.e. what has been uttered so far and what is likely to be the full utterance meaning respectively. We have defined how grounding acts can be identified incrementally and how the grounding state, i.e. the collected contents and progress of the common ground units (CGUs), is updated accordingly. We defined new types of acknowledgments and how they affect the content of the CGU they ground, e.g. answering an unfinished question also grounds the part of the question that was not uttered. We implemented our model in the SASO4 dialogue system as a proof-of-concept of our approach, showcasing an up-to-date grounding state through the execution of a simple overlapping feedback policy.

Chapter 1

Introduction

On the many layers of human conversation, interlocutors continuously interact to facilitate a fluent and effective communication process. They share information, convey intentions through nonverbal behavior, finish each other's sentences, request the turn, interrupt the speaker and signal understanding. The latter is of particular interest for this thesis.

Understanding plays a key role in effective communication. Typically, the speaker will continuously introduce new information for the hearer to understand so eventually the conversation's goal can be met. But understanding on its own is not enough, the hearer also needs to signal understanding by acknowledging the new information. Then the speaker knows that the new information is now shared and the next topic can be discussed. This process is called *grounding* and the shared information is called the *common ground* [1, 2].

Signaling understanding can be done in different ways, ranging from not doing anything at all, i.e. not signaling misunderstanding, to taking the turn and elaborating on the new information[1]. The first conveys less evidence of understanding than the second, but in some situations that might just be enough [3]. Also, the second, more elaborate, way of showing understanding has an impact on the efficiency of the dialogue as it involves a change in who is the speaker. Therefore, listeners will often choose to signal understanding without interrupting the speaker, using backchannels, such as a head nod, "yeah" or "ok", overlapping the other person's speech without taking the turn, while still providing sufficient evidence of understanding.

Since the rise of the computer, in the 1940s and 1950s, scientists have been interested in recreating the human language capabilities in artificial systems [4]. Out of that interest, the field of *natural language processing* was born. Natural language processing is an interdisciplinary research area that operates on the intersection of computer science, psychology and linguistics. The goal of this field is to equip computers with capabilities that will allow them to perform tasks involving natural, human language.[1] In our everyday life, we encounter applications that have their roots in natural language processing research. For example, Google Translate [5],

[1]Natural language is the contrary of artificial constructed, such as programming languages.

Figure 1.1: The twins talking with a group of children in the Museum of Science, Boston.

Apple's Siri [6], Wolfram Alpha [7], computerized customer service help desk agents and voice controlled on board computers in cars.

Some of the aforementioned natural language processing applications involve having a conversation with the system. These systems are called *dialogue systems*. A dialogue system is designed to partake in a conversation and act and respond like a human would. Those systems are often represented as an *agent*, having a name, some sort of personality and, in case of a *spoken dialogue system*, a voice. These traits are realized by a complex system that involves speech recognition, language understanding, discourse reasoning, response planning, language generation and speech synthesis.

A state of the art spoken dialogue system is capable of having a conversation with the user on a *finite-domain* of recognized inputs and possible answers. For example, the Higgins dialogue system [8] can help the user find his way through a virtual village, Gunslinger [9] emerges the user in a movie-like scenario in which he can become the hero of the virtual inhabitants of a Wild West town by getting rid of a vicious outlaw and twins Ada and Grace [10] act as virtual museum guides that can answer visitor's questions (see Figure 1.1).

What all the mentioned systems and any typical state of the art dialogue system have in common is that they are still far from being sophisticated enough to engage in real human-like conversations. They are capable of having a conversation when the user obeys to a rigid dialogue structure, speaking clearly and in full sentences, but, until recently, systems would fail at participating in real fluent human-like conversations with frequent overlapping behaviors.

Recent efforts in the field of natural language processing have resulted in spoken dialogue systems that can engage in more human-like fluent conversations than described above, using incremental interpretation of user's speech [11, 12] and a more comprehensive listening behavior [13, 14, 15, 16], in order to move away from the rigid structure of typical human-computer conversations.

1.1 Incremental processing

In a typical non-incremental dialogue system, the system processes the user's utterances in a single piece. When the user is done talking, the ASR automatic speech recognition component (ASR) will transcribe it and pass it on to the rest of the system's components. In a dialogue system that is capable of incremental processing, the user's utterance is being processed while it is still in process. With short time intervals, the ASR will attempt to transcribe the utterance so far and pass that partial transcription on to the rest of the system. The other components can use that partial transcription for their own processing, e.g. the natural language understanding component (NLU) will try to find the meaning of the utterance so far. These incremental updates can be used to make decisions while the user is still talking. The system can decide to perform a backchannel, interrupt or help the user if he has trouble finding the right words.

Using incremental processing, a dialogue system can display an affective listening behavior towards the user, building rapport[2]. The Virtual Rapport [14, 15] monitors the pitch, loudness and fluency of the user's speech and tracks his posture shifts, gaze and head movements. The agent can nod, shift its posture or gaze and can mimic the user's body language, while the user is speaking to the agent. A user study showed that this increases the perceived naturalness and efficiency of the conversation.

In [17], the authors present a system that can finish the user's utterances. The system processes the incremental updates from the NLU to determine when it has reached the point of maximum understanding of an ongoing utterance. When used strategically, this ability can complete utterances in situations when this would positively contribute to the dialogue.

In [13], a system is presented that is capable of even more comprehensive incremental listening behavior. It uses incremental updates from the NLU to show, e.g. by nodding or frowning, whether the system is understanding what the user is saying.

1.2 Motivation

To date, the incremental dialogue systems, including those mentioned above, have not closely linked the incremental listening behaviors to the grounding model. Those listening behaviors however, convey information on whether the system is understanding and thus influence the grounding of the content that is being discussed in the conversation. Instead of being directly based off of prosody and NLU results, they should be initiated by a grounding model. A grounding model keeps track of the content that is being presented over the course of the dialogue and knows what behaviors to execute in order to ground pieces of that content. The incremental listening behaviors then become meaningful tools that contribute to efficient conversation. This does require a *incremental* grounding model, which does not exist yet.

[2]Rapport is a relationship between two conversational partners in which they understand each other's ideas and feelings and communicate well.

1.3 Problem Statement

Coming up with an incremental grounding model poses two main challenges. The first challenge originates from the incremental context, the second has its root in the shortcomings that existing non-incremental grounding models may have .

The incremental results that are generated in an incremental dialogue system during the user utterances are less reliable than the results from non-incremental processing. Many small pieces of data are processed instead of a few large chunks. The components of the dialogue system have less information to work with. Also, the separation of the user's speech into those small chunks is arbitrary, e.g. by using a 200ms window. It will happen that a chunk boundary occurs within a single word. The ASR might transcribe a chunk of speech as 'four', until the next chunk comes in and it turned out to be 'forty'. These errors propagate to the other components of the dialogue system that rely on the transcription. An incremental grounding model should be robust to those errors.

Existing grounding models have never been tested in incremental dialogue systems. While the theoretical groundwork of those models typically uses *processing units* of undefined length, the dialogue systems in which they have been implemented have always been non-incremental. Therefore, the existing implementations can rely on the traditional simplifying assumption that the conversation between the user and the system occurs in a strict turn by turn fashion. The system will not speak while the user is speaking and it is assumed that this also holds the other way around. In an incremental system, utterances are processed while they are being uttered and responses can overlap. An incremental dialogue system will be able to pick up a range of new overlapping behaviors that humans use for grounding purposes. The incremental grounding model needs to be able to support those behaviors. It is not clear if an existing grounding model would work or how it can be adapted to do so.

1.4 Research Goals

The main goal of our work is to come up with a model of incremental grounding and implement it in an incremental spoken dialogue system. Our point of reference is a typical grounding model in a non-incremental spoken dialogue system. To get an understanding of how to go from the point of reference to our research goal, we defined two sub-goals: 1) finding the differences between non-incremental and incremental dialogue processing and 2) finding the new grounding behaviors that an incremental grounding model should support in addition to the non-incremental acts.

1.5 Methodology

The first sub-goal, in which we investigate the differences between non-incremental and incremental dialogue systems, is achieved through a study of the literature in the field of natural language processing. This study includes the works that present new incremental approaches to specific components of the dialogue system or the system as a whole.

The second sub-goal, in which we set out to find new overlapping behaviors that an incremental model for grounding should support, is achieved by a study of the AMI Meeting Corpus [18]. The analysis of several interesting dialogue excerpts is conducted using theoretical frameworks from computational linguistic literature.

The main goal, i.e. creating a model of incremental grounding, calls for the combination of the results of both sub-goals. The theoretical model is developed to benefit from the strengths of incremental processing, while being robust to its negative characteristics. The model is implemented in an existing incremental spoken dialogue system to validate the design.

1.6 Contributions

The work presented in this thesis consists of the theoretical treatise of the relevant concepts mentioned above, a corpus study of overlapping grounding behavior, the development of a theoretical model for incremental grounding and the implementation of the model in an existing dialogue system.

1.7 Previous Work

Parts of this thesis are based on work already published in a paper that was written by the author together with David Traum and Rieks op den Akker, both supervisors of this thesis, and David DeVault.

T. J. Visser, D. R. Traum, D. DeVault, and R. op den Akker, "Toward a model for incremental grounding in spoken dialogue systems," in *12th IVA Workshop on Real-time Conversations with Virtual Agents*, Santa Cruz, 2012

1.8 Outline of Thesis

Chapter 2 discusses the two central concepts of this thesis: grounding and dialogue systems. It provides the user with a primer on both topics, discussing existing grounding theories and explaining the typical components of a dialogue system.

Chapter 3 starts with a discussion of incremental processing in spoken dialogue systems. In this section, existing approaches to the system architecture and individual components are

reviewed. This is followed by a study of the AMI Meeting Corpus, looking for examples of incremental grounding behavior in human-human dialogue. The remaining part of the chapter presents the new theoretical model for incremental grounding.

Chapter 4 describes the specific dialogue system that was used for this thesis, called SASO4. A discussion of the implementation of the theoretical model presented in the previous chapter follows. An important part of the implementation is the response policy, which determines when overlapping responses by the system are desired.

Chapter 5 provides a final discussion and summary of the contributions made in this thesis. The chapter is concluded by a discussion on how the model and implementation presented can be further developed.

Chapter 2

On Grounding and Dialogue Systems

2.1 Overview

All contributions to a dialogue are built on *common ground* [2]. The common ground contains knowledge that can be required to properly process new contributions. When a speaker presents a new contribution and it is accepted by the listener, its content will update the common ground, enabling every next contribution to build upon the previous. The process of updating the common ground is called *grounding*.

Common ground exists between two (or more) people and consists of their *common knowledge*, *mutual knowledge* or *belief*, which all are different notions used in literature to describe roughly the same. Something is said to be in the common ground if it is known to both parties and is known to be known. This can be expressed more formally, and perhaps more clearly, for two persons A and B as

$$p \text{ is CG} \iff bel(A,p) \wedge bel(B,p) \wedge bel(A, bel(B,p)) \wedge bel(B, bel(A,p)) \tag{2.1}$$

where $bel(P,q)$ models the belief of proposition q by person P.

In conversation, new content is added to the common ground, to a specific part of it which Clark calls the *personal common ground* [2]. This type of common ground is created and updated through personal experiences with each other. The opposite of the personal common ground is the *communal common ground*. The content of the communal common ground is defined by shared culture, nationality, profession, education, hobbies or religion. As soon as you find out you have something *in common* with the other person, e.g. you both love classical music, the common ground suddenly expands to incorporate topics of the newfound similarity.

At the most fundamental level, the communal common ground allows people to assume an agreement on the meaning of the words that make up the language that they speak, but it also allows for cultural references and professional jargon. The culture or jargon does not change over the course of the dialogue, but the extent of the communal common ground does as

commonalities with the conversational partner are discovered.[1] The personal common ground on the other hand continuously changes during the conversation, reflecting the latest conversational topics. In an influential model of grounding by Clark and Schaefer [1], the changes to the common ground are modeled as *contributions* to discourse.

Clark and Schaefer distinguish two phases in the grounding of a contribution: the *presentation* phase and the *acceptance* phase [1]. In the presentation phase, the speaker presents a piece of new content for the listener to consider. The speaker assumes that if the listener provides evidence of at least a certain strength, he can believe that the listener understands what he meant. In the acceptance phase, the listener accepts the new content by giving evidence of understanding, assuming that this evidence will make the speaker believe that he understands. The acceptance itself is also considered a contribution, which in turn needs to be accepted. Consider the following example:

(1) A: Maybe even pre-programmed sound modes, like um
B*: Okay
A: the user could determine a series of sound modes.
C*: Mm-hmm

This dialogue excerpt contains four contributions, one for each utterance. The first contribution is presented with "Maybe (…) modes" and accepted by "Okay". B's acceptance is also a presentation of a new contribution, which is then accepted by A's next utterance and so on.

2.2 Evidence of Understanding

The evidence of understanding that a listener can give to show his acceptance of the contribution can, according to Clark and Schaefer, be one of the five types listed in Table 2.1. The types of evidence are ordered in increasing strength. However, it can be argued that evidence 4 might be stronger than evidence 5, since it shows that the listener has processed the contribution on a deeper level [20]. In dialogue 1 above, B's "Okay" and C's "Mm-hmm" are acknowledgments and A's second utterance initiates the relevant next contribution at the same level as the last one.

The type of evidence that a listener gives is based on a trade-off between the effort it takes to provide such evidence and the strength of the evidence, providing at least enough evidence to sufficiently ground the contribution for the current purposes or, in other words, meet the *grounding criterion* [21].

The grounding criterion of A's contributions is higher than that of the acknowledgments by B and C, i.e. A's contributions require stronger evidence in order to be accepted. Acknowledgments are easy to understand and accept and therefore require less evidence. That is why A's contributions seem to require acknowledgment evidence and B's contribution is accepted by merely initiating the relevant next contribution.

[1] There is a lot we share through communal common ground, even with people we've never met. In contrast, think about how you would explain something simple as a bus stop to a Martian.

1.	*Continued attention*. B shows that he is continuing to attend and therefore remains satisfied with A's presentation.
2.	*Initiation of the relevant next contribution*. B starts in on the next contribution that would be relevant at a level as high as the current one.
3.	*Acknowledgment*. B nods or says "uh huh", "yeah", or the like.
4.	*Demonstration*. B demonstrates all or part of what he has understood A to mean.
5.	*Display*. B displays verbatim all or parts of A's presentation.

Table 2.1: Types of Evidence of Understanding, from Clark and Schaefer [1, p. 267]

2.3 Levels of Action

Up until now, we have been talking about grounding in the sense of understanding what the speaker is saying and providing evidence of that understanding. In this section, we go into more detail about what actions are required before what we have been calling 'understanding' can be reached.

Consider the act of asking a question. For most questions, the ultimate communicative goal that the speaker has, is to get an answer. This requires a listener who is aware of the convention that when a question is asked you are supposed to give an answer. This requires that the listener understands the intended meaning of the sentence, i.e. it being a question. In turn, this requires from the listener that he recognizes that the sound coming from the speaker's mouth form words. And at the most fundamental level, this requires that the listener attends to the sound coming from the speaker. Analog to these actions by the listener, there is a same set of actions that the speaker performs by asking a question: he makes the sound that corresponds to the words he wants to say, he presents the words, he asks a question and he proposes to the listener to answer the question.

Clark introduced the notion of *action ladders* to capture the levels of action in communication [22]. Table 2.2 contains the combined ladder of the actions that are performed by the speaker and the listener, resulting in the four levels of *joint action*. Continuing the example from before, we can now say that at the conversation level the speaker proposes and the listener considers the project of asking and answering a question, at the intention level the speaker is signaling and the listener is recognizing a question, at the signal level the speaker presents and the listener identifies the verbal utterance and at the channel level the speaker executes and the listener attends to the sounds that make up the utterance.

Since each level is built on top of the level below, a communication error at a lower level will make it impossible to succeed at the levels above. For example, if the speaker fails to recognize the utterance as a question, he will certainly not consider answering it. This is what Clark calls *upward causality*.

This also works the other way around. If there is evidence that the action on a higher level

Level	Speaker S	Listener L
4. Conversation	S is proposing activity w	L is considering the proposal of w
3. Intention	S is signaling that p	L is recognizing that p
2. Signal	S is presenting signal s	L is identifying signal s
1. Channel	S is executing behavior t	L is attending to behavior t

Table 2.2: The four action levels, from Clark [22, p. 152]

succeeded, the levels below must also have succeeded. For example, if the listener provides a fitting answer to the question posed by the speaker, he must have heard and understood the speaker correctly. This is what Clark calls *downward evidence*.

To ensure that errors are detected and recovered from, coordination between the conversational partners is required on four levels. This is achieved by grounding the mutual understanding on all levels. In this light, the commonsensical meaning of 'understanding' is just a special case of succesful coordination. What we have been calling 'understanding' can now be defined more formally as understanding on level 3 and up.

Now we know that grounding occurs on all four action levels, we can also talk about how evidence of understanding (see Section 2.2) relates to this. Depending on the type of evidence, it can only be used to infer understanding up to a certain level. For example, the weakest evidence type that Clark and Schaefer describe, Continued attention, only requires that the listener attends to the speaker's behavior, thus only provides evidence of understanding on level 1. In some cases it is sufficient to merely provide evidence of understanding up to a lower level, e.g. when the conversational partners have rapport or if the grounding criterion is low.

2.4 A Theory of Computational Grounding

Because of the fundamental role that grounding plays in human-human conversation, it is also an essential part of human-computer interaction. Clark's work describes a formal model of language use, including grounding, but his primary audience is the cognitive psychologists and psycholinguists, not computer scientists and computational linguists. Before his theory could be applied in the context of dialogue systems, it needed to be adapted for computational use. This task was taken up by Traum, who came up with a computational theory of grounding [20, 23] which has since become an influential approach to solving the grounding problem for dialogue systems.

Traum addresses several aspects of Clark and Schaefer's model that are not well suited for computational use [24]. In the two-phase structure, it is required to specify how much acceptance the second phase needs. The grounding status of a contribution therefore depends on its own acceptance phase, the acceptance of its acceptance phase, etc. Also, with just the two phase concepts at hand, the description of the grounding process is too coarse to be able to tell the

grounding state of the current contribution after each utterance. Often, larger parts of the conversation are needed before the effect of a single utterance can be seen. This is not of much use for a dialogue agent that, in the middle of a conversation, needs to decide what its next action will be.

Instead of the two phases of presentation and acceptation, Traum defines seven *grounding acts* that perform a specific function towards the grounding of a piece of content. Clark and Schaefer called this piece of content, i.e. the unit of grounding, a 'contribution', while Traum calls it a 'discourse unit' (later renamed to *Common Ground Unit* or CGU). 'Contribution' and 'CGU' are similar concepts, but the CGU is more closely related to surface structure of the dialogue [25], and therefore better suited for analysis from the perspective of a dialogue system, while the dialogue is taking place.

Label	Description
initiate	Begins a new CGU
continue	Adds new content to an open CGU
acknowledge	Provides evidence of understanding of the CGU
repair	Removes, adds or replaces content from the CGU
request repair	Signals lack of understanding
request ack	Signals the need for evidence of understanding
cancel	Ends the work on a CGU, leaving it ungrounded

Table 2.3: The seven grounding acts from Traum's model, adapted from [24, p. 127]

The seven grounding acts are presented in Table 2.3. Each CGU begins with an *initiate* act, in which the speaker presents new content to the conversation. The speaker that initiates a CGU is assigned the *Initiator* (I) role for that CGU. The initiator can *continue* in the following utterances, which adds more content to the current CGU or *repair* to revise the content of the CGU. If the listener, who is also the *Responder* (R), of the CGU does not understand what the initiator means, he can signal his lack of understanding by performing a *request repair*. When the responder understands, he can *acknowledge* the CGU. If the responder fails to acknowledge the CGU and the initiator is uncertain about the understanding of his partner, he can solicit evidence of understanding by performing a *request acknowledgment*. The grounding of a CGU can be abandoned by a *cancel* act, which leaves the CGU ungrounded and ungroundable.

The sequence of actions described in the paragraph above is just one of the many possible courses grounding can take. Table 2.4 contains the transition diagram of the finite automaton that models the grounding of a CGU. For all grounding acts, it describes the effect it has on the grounding state of a CGU, given the previous state of the CGU. Most CGUs will reach the final state F, meaning that the content has become common ground.

For example, consider dialogue excerpt 2[2], in which two persons try to manage a rail road freight system. The dialogue is from the TRAINS project [26], the annotations are adapted from

[2]We annotate the dialogues using the following notation: $\text{GroundingAct}_{\text{Role}}\text{CGU\#}$

	In State						
Next Act	S	1	2	3	4	F	D
Initiate^{I}	1						
Continue^{I}		1			4		
Continue^{R}			2	3			
Repair^{I}		1	1	1	4	1	
Repair^{R}		3	2	3	3	3	
ReqRepair^{I}			4	4	4	4	
ReqRepair^{R}		2	2	2	2	2	
Ack^{I}				F	1*	F	
Ack^{R}		F	F*			F	
ReqAck^{I}		1				1	
ReqAck^{R}				3		3	
Cancel^{I}		D	D	D	D	D	
Cancel^{R}			1	1		D	

*repair request is ignored

Table 2.4: Traum's CGU transition diagram. A CGU is said to be in the common ground when it reaches state F [20, p. 41].

[20, p. 66]. The first CGU is initiated by A as he proposes to move engine E from Avon. The initiate grounding act is *conveyed* by A's utterance, the content of the CGU is the function in the conversation, intentional meaning, etc. of that utterance (see Clark's four action levels above). Before A has finished the utterance, B corrects a mistake that A made, which is a repair act. A immediately acknowledges the repair, which grounds CGU 1. Note that Traum's 'acknowledge' is different from Clark and Schaefer's 'Acknowledgement' in that the first is meant to cover all types of evidence of understanding while the latter is just one way to convey understanding. In this example, A's "okay" is coincidentally both. B follows up with another acknowledgment, which does not affect the state of the CGU. A starts to utter his proposal from the start of the dialogue excerpt for the second time, now incorporating the repair from B, which initiates a new CGU, but does not complete it. Note that this is a new CGU, even though its content is very similar to the first CGU. The creation of new CGU's is primarily based on the grounding process and not on the CGU's content. A however appears to change his mind halfway in the sentence. A hesitates and decides to cancel the proposal, abandoning CGU 2 and leaving it ungrounded. A continues with a new proposal, initiating CGU 3.

(2) A: [so we should move the engine at Avon engine E to]$^{\text{Initiate}_I 1}$
B*: [engine E1]$^{\text{Repair}_R 1}$
A: [E1]$^{\text{Acknowledge}_I 1}$
B: [okay]$^{\text{Acknowledge}_R 1}$
A: [engine E1 to Bath to]$^{\text{Initiate}_I 2}$ [or]$^{\text{Cancel}_I 2}$
A: [we could actually move it to Dansville]$^{\text{Initiate}_I 3}$

2.5 Spoken Dialogue Systems

In this section, we describe a typical academic spoken dialogue system in order to provide an introduction to the various parts it is composed of. This will clarify the subsequent discussion on grounding in such a system and the work of this thesis as a whole. A more detailed description of a spoken dialogue system is provided in Chapter 4, where the SASO4 dialogue system is discussed.

A well-known approach to building a dialogue system is to use a pipeline architecture, in which a chain of components takes a user utterance as input and comes up with a system response as output [27, 28]. Figure 2.1 gives an overview of such an approach. In a typical dialogue system, the components process a whole user utterance at a time. After the user is done talking, indicated by releasing the push-to-talk button or assumed after a certain amount of speech inactivity, the ASR will take the audio signal and come up with a hypothesis of the complete utterance. The NLU will then work with the ASR hypothesis and determine the meaning of the utterance. The Dialogue Manager performs additional analysis and consults the internal state to decide on the type of response. The Natural Language Generator transforms the Dialogue Manager's result into natural language, which is then converted into speech by the Text-to-speech Synthesizer.

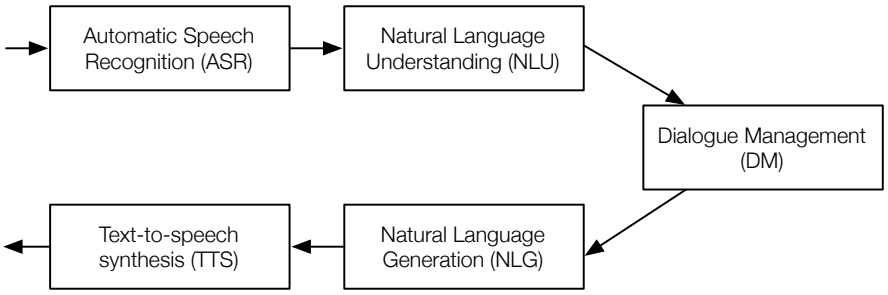

Figure 2.1: An overview of the architecture of a typical spoken dialogue system.

2.5.1 Automatic Speech Recognition

The Automatic Speech Recognition (ASR) component turns the audio signal from the user's microphone into a written transcription of what the user is saying. A typical ASR uses an *acoustic model*, which describes the probability of a word sequence given the observed audio, and a *language model*, which describes the probability of certain word n-grams (usually bigrams and/or trigrams). With those two models combined, the ASR can calculate the most probable transcription for the observed audio signal.

2.5.2 Natural Language Understanding

The Natural Language Understanding (NLU) component uses the ASR output to determine the meaning of the user's utterance. An example representation, often called a *frame*, can be found in Figure 2.2. This frame represents the meaning of "Utah, do you want to be the sheriff?", but also of "Would you agree to becoming the sheriff, Utah?" The NLU will collapse all the ways of saying the same thing into a single frame. The frames are constructed using a vocabulary of concepts that the Dialogue Manager will be able to process.

```
s.addressee            utah
s.mod                  interrogative
s.sem.speechact.type   info-req
s.sem.type             question
s.sem.q-slot           polarity
s.sem.prop.agent       you
s.prop.type            event
s.prop.event           accept
s.prop.theme           sheriff-job
```

Figure 2.2: An example NLU frame

Approaches to the NLU problem include the use of a Context Free Grammar (e.g. [29]), keyword or keyphrase spotting and data-driven statistical language modeling [30]. The latter has been found to have an increased robustness, i.e. in dealing with unseen utterances and ASR errors, compared to the other two approaches [28].

2.5.3 Dialogue Management

The Dialogue Manager (DM) is responsible for three aspects:

* Contextual interpretation: interpret pragmatic meaning
* Domain reasoning: reason about world and update internal state
* Action selection: decide what to do next

The DM performs the final processing of the input and initiates processes that will lead to system response. It is the core component of the dialogue system.

The representation provided by the NLU denotes the context-free meaning of the user's utterance. It is the DM's task to interpret this within the current context to figure out the pragmatic meaning. This includes the resolution of named entities, referring expressions (e.g. "I", "here" and "that") and recognizing the dialogue acts that have been performed by the utterance. Among the possible dialogue acts that an utterance can perform are the seven grounding acts from Traum's theory.

Based on the fully interpreted user utterance, the DM can reason about how it relates to its model of the world, i.e. which states, events and objects are mentioned, and its stance towards the utterance content. The grounding state of the relevant CGUs is updated based on the recognized grounding acts and, if applicable, new content is added to the common ground.

Finally, the DM will select the communicative act that is to be performed by the system. This act should convey the dialogue acts that the DM wants to communicate based on the outcome of the contextual interpretation and domain reasoning. If the DM encountered an ambiguous referring expression that prevented full understanding of the utterance, a clarification request, i.e. a request repair in terms of grounding, might be the appropriate response in order to reach understanding. If the DM did understand the utterance, it should pick up its role in the proposed project, providing evidence of understanding by doing so.

For example, if the user says "Utah, do you want to be the sheriff?" and the NLU returns the corresponding frame as shown in Figure 2.2, the system could show its understanding by taking up on the project of answering the question. The answer would *demonstrate* (i.e. one of Clark and Schaefer's types of evidence of understanding, see Table 2.1) understanding of the question and successfully ground it.

2.5.4 Natural Language Generation

The Natural Language Generation (NLG) component uses the semantic representation of the communicative act from the DM and generates a corresponding textual representation, the *surface text*, that is to be synthesized by the text-to-speech component. The least complex approach to NLG is to have one or more surface texts ready for every possible communicative act and simply select one for the act at hand. More complex approaches exist that aim at increasing the flexibility of the output, e.g. by parsing the semantic representation with a grammar of known representation chunks linked with pieces of surface text [31].

The ability to have a flexible choice between various surface texts for a communicative act, including the ability to use anaphoric expressions, elliptical constructions and make conceptual pacts with the user will make the conversation more natural and efficient [32].

Spoken	Utah, do you want to be the sheriff?
Recognized	utah do you want the we the sheriff utah do you want the sheriff you town you want we the sheriff

Table 2.5: Example ASR errors

2.5.5 Text-to-speech Synthesis

The simplest approach to generating audio output for a dialogue system is to have all possible utterances prerecorded. While the output quality will exceed any of the alternatives, this approach is not applicable to a system beyond the most limited ones. A more flexible solution is text-to-speech synthesis (TTS). The task of a TTS is twofold: 1) calculation of the pronunciation of the utterance and 2) generation of an audio signal of the pronunciation.

2.6 Errors in Spoken Dialogue Systems

Listening and speaking does not come as natural to dialogue systems as it does to humans. A system's speech recognition and language understanding capabilities are not nearly as sophisticated as that of humans. Therefore, a dialogue system can never be certain of what the user is saying, it can only *hypothesize*. It is not clear if that really differs from our human capabilities, but it seems that we are doing fine nonetheless. However, dialogue systems will have to learn to deal with frequent errors.

In the typical pipeline architecture of dialogue systems, the result of a single component influences all downstream components. For example, an incorrect ASR hypothesis might result in the wrong frame being selected by the NLU, which consequently moves the DM to select a communicative act that the user will not understand. Such propagating errors are not unlikely, since even a state-of-the-art ASR will frequently misinterpret the user's speech and return incorrect words in the transcription. Table 2.5 contains three examples of ASR hypotheses with errors.

The performance of an ASR can be measured with the *word error rate* (WER). It is defined as the number of incorrect words in the ASR hypothesis divided by the number of words in the user's utterance. Existing literature reports relatively high WER values, e.g. 23.6% ([28, p. 138]), 39% ([33]) and 54% ([34]). The performance of an ASR can be improved by training it with the same equipment, e.g. microphone and sound card, that is being used in production, with the same user that will be operating the system and on example utterances that fall inside the domain of the system. Remaining errors can however be corrected by the components downstream.

Depending on the type of NLU, it might be able to select the correct frame despite of errors in the ASR hypothesis. [30] describes a statistical approach that classifies every incoming utter-

ances as one of the NLU frames from a finite set. It does require a large corpus of utterance-frame pairs to train the classifier. But if this corpus consists of real ASR transcriptions, including real ASR errors, the classifier will learn to map possibly erroneous ASR hypotheses of an utterance to the correct frame representing the actual user utterance.

The performance of a classifier can be measured using f-score. [30] reports an f-score of 74.46% with a WER of 35.6%, and [34] reports the same f-score with a WER of 54%, showing that a robust NLU can compensate for errors made by the ASR.

If the NLU sends the incorrect frame to the dialogue manager, it will impede all three tasks that the DM performs. The DM might draw incorrect conclusions when resolving contextual references, corrupt the internal state with incorrect information and select an incorrect conversational act to respond with. This can be resolved by having the NLU provide more information about its analysis. It could provide an *n-best list* of frames instead of a single frame. The DM could then re-rank the frames based on contextual information, i.e. how likely it is that a frame occurs given the current state of the dialogue, and select a frame that is more likely to be correct. The NLU could also provide *confidence metrics* that indicate the confidence the NLU has in its prediction [33]. If the confidence is below a certain threshold, the DM can conclude that the NLU did not understand and make the system show its misunderstanding.

2.6.1 Grounding in Spoken Dialogue Systems

Because of the apparent risk of misunderstanding, grounding plays an important role in dialogue systems. By keeping track of grounding, the DM can determine which actions are required to reach understanding.

The DM can assess its degree of understanding based on the confidence of the ASR, NLU and the confidence in its own contextual interpretation. The DM then can decide whether to accept or reject the hypothesis. A good decision strategy minimizes the *false acceptances* and *false rejections* and maximizes *true acceptances* and *true rejections* (see Table 2.6).

	System Accepts	System Rejects
Correct Hypothesis	True Acceptances	False Rejections
Incorrect Hypothesis	False Acceptances	True Rejections

Table 2.6: The four outcomes of accepting/rejecting a hypothesis.

If the DM accepts the hypothesis, it will have to provide evidence of understanding to the user in order to make the content grounded. The type of evidence to give is, as discussed previously, a trade-off between effort of execution and strength of evidence, while making sure that the grounding criterion is met. Because of the system's susceptibility to errors, the system might want to err on the side of putting too much effort into providing evidence of understanding.

Using one of the stronger two evidence types, i.e. demonstration and display, the system can provide implicit verification of the content. Consider the following example from [35]:

(3) U: I want to go to Swalmen
S: When do you want to go to Swalmen?

The system's (S) response is intended as an acknowledgment grounding act, but by explicitly repeating the content that is to be grounded, the user (U) is also given the opportunity to repair. If the user continues by answering the question, he passes on the opportunity to perform a repair and thereby provides evidence that the initial request to go to Swalmen is now grounded. If the user had requested to go to Almen instead of Swalmen, the system's response would not be an acknowledgment, as intended. This would become clear to the system as soon as the user performs a repair.

If the system would not be confident about understanding the user's utterance, it could perform a request repair by explicitly verifying the content. The following example is also from [35]:

(4) U: I want to go to Swalmen
S: Do you want to go to Swalmen?

The user would have to respond to the request repair, increasing the combined effort it takes to ground the user's initial utterance. This has a negative impact on the efficiency and fluency of the conversation, if the system's understanding would turn out to be correct to begin with. On the other hand, if the system would have performed an acknowledgment of an incorrect hypothesis (i.e. the case where in Dialogue 3 the user actually said Almen instead of Swalmen), the user would have to put effort into correcting the mistake instead of answering the question about travel time.

In Dialogue 4, the system correctly processed the user's utterance, but had low confidence in the result. More often, there will be errors on one of the lower action levels that either the ASR or NLU is concerned with that prevent the system from sufficient understanding in order to generate a verification. Consider the following example:

(5) U: I want to go to Swalmen
a) S: I couldn't hear what you were saying
b) S: I don't understand what you mean

The system could use a) to indicate an error with the ASR, it is a request repair on the signal level. This could make the user reposition the microphone or talk louder. Alternative b) indicates a problem with the NLU, which deals on the intention level. The intended meaning of the user's utterance is not clear and the user should try to rephrase it (see e.g. [36] for a system that deals with those misunderstandings in a similar way).

A dialogue system can also provide evidence of understanding on all four action levels. Usually, the system's response will convey acceptance of the project that the user is proposing, thus ground on all four levels at once. While the project might only be clear at the end of the utterance, on the lower levels there are finer grained concepts that could be grounded before the end of the utterance. In human-human conversation this happens a lot through vocal and non-verbal backchannels (e.g. "okay", "yeah" or head nods). This contributes to the efficiency and fluency of the conversation, preventing the speaker from over-elaborating [37]. The type of dialogue system that we have been discussing so far would not be able to do this, since its components operate sequentially on the complete user utterances. That is why, recently, researchers have started working on spoken dialogue systems that can incrementally process the user's input, increasing the responsiveness and getting rid of the rigid dialogue structure that most dialogue systems suffer from [38].

Chapter 3

A Model of Incremental Grounding

3.1 Overview

A dialogue system is said to be capable of incremental processing if it starts processing before the user utterance is complete. In such systems, each component will start processing after receiving a minimal amount of its characteristic input [39]. For the ASR, this *incremental unit* (IU) is a short fragment of audio signal, for the NLU, this is any change to the ASR's hypothesis and for the DM, this is any change to the NLU output.

Because the utterance is still in the progress of being uttered at the time of processing, the components generate output based on incomplete information. As new information comes in, and the information becomes more complete, the components might need to revise their hypothesis [40]. Consider the sequence of hypotheses in Table 3.1 that the ASR produces as the user utters "Utah, do you wanna be the sheriff?" At $t = 1$, the ASR hypothesizes that the word "New" has been spoken. This hypothesis was probably generated right after the user uttered the "U" of "Utah". Another revision occurs at $t = 9$ as the ASR goes "meet you sure" to "be the sheriff". The hypothesis at $t = 8$ was probably generated right after the user said "sher". The 2-gram "meet you" probably has a high probability in the language model, but with the addition of "sheriff", the 3-gram "be the sheriff" has a stronger preference.

The other components that rely on the ASR's output will have to deal with those revisions and probably in turn revise their own output as well. It does not have to be the case however that a component outputs the same amount of IUs as it receives as input. Some components might accumulate multiple IUs before producing output or the other way around. An example of the former is the NLU. The frame elements of the NLU frame in Figure 2.2 (p. 19) have no 1-to-1 mapping with the words in the corresponding ASR hypotheses in Table 3.1. The NLU accumulates the changes from several incremental hypotheses before a new frame element is added to the NLU output or produces multiple new frame elements after a single incoming IU. Table 3.2 contains a possible mapping between the ASR hypotheses from Table 3.1 and the

t	ASR Hypothesis
1	New
2	Utah
3	Utah it
4	Utah do
5	Utah do you why
6	Utah do you wanna be
7	Utah do you wanna be the
8	Utah do you wanna meet you sure
9	Utah do you wanna be the sheriff

Table 3.1: The ASR hypothesis is revised as more information becomes available.

frame elements of the corresponding frame.

t	NLU frame	
2	`s.addressee`	`utah`
4	`s.mod`	`interrogative`
5	`s.sem.speechact.type`	`info-req`
5	`s.sem.type`	`question`
5	`s.sem.q-slot`	`polarity`
5	`s.sem.prop.agent`	`you`
9	`s.prop.type`	`event`
9	`s.prop.event`	`accept`
9	`s.prop.theme`	`sheriff-job`

Table 3.2: An NLU frame with for each element the time-step (from Table 3.1) when it is added to the NLU output.

3.1.1 Predictive vs. Non-predictive

The approach to incremental processing described above can provide a basis for feedback behaviors such as head nods, shakes, gaze shifts and backchannels. Based on the growing ASR and NLU hypotheses, the DM can provide early grounding on the lower levels of Clark's action ladder. For some other responsive behaviors, the strictly incremental, *non-predictive*, interpretation is not sufficient and a prediction of the interpretation of the full utterance is required. Behaviors such as timing a reply to have little or no gap, grounding by saying the same thing at the same time or performing collaborative completions require this [17].

Sagae et al. and DeVault et al. describe an alternative approach to incremental processing [34, 17, 41, 11, 42, 33]. In these works, a *predictive* incremental NLU component is presented. Based the ASR hypothesis of a partial utterance, it will attempt to predict the full utterance meaning. The component also provides several confidence metrics related to the prediction, which the DM should take into account when using the NLU's result.

Both predictive and non-predictive incremental processing are valuable in dialogue systems. The two approaches combined even more, because then a distinction can be made between what has been said so far and what is likely to follow. Such a hybrid approach was presented in [27], which describes the SASO4 dialogue system that is also used for this thesis and will be discussed in more detail in Section 4.

3.1.2 Incremental language generation

So far, we have talked about incremental language understanding. A fully incremental dialogue system will however also have to be able to incrementally generate its output. This will allow the system to plan, realize and monitor its output while simultaneously processing the input from the user [43, 44]. By monitoring its own output, the system knows the explicit content of the utterance at any moment while it is in progress. This allows the system to more accurately interpret overlapping feedback from the user by relating it to the content uttered up to the moment of the feedback.

This thesis however - much like most work in grounding in spoken dialogue systems -, is primarily concerned with the state of understanding of the system. We focus on how to deal with system non-understanding and misunderstanding and how to provide feedback to the user of the system's understanding or the lack thereof. The system's capabilities are however only one side of the coin, as the user will also convey his understanding with overlapping behaviors.

While the theoretical model that is presented in this chapter does not make assumptions about how the role of Initiator and Responder are divided among the user and the system, the implementation that is presented in Chapter 4 primarily discusses how the system should provide feedback, i.e. the *feedback policy*, as it listens to the user.

3.2 Incremental Grounding Behavior in Human Dialogue

We have studied human-human dialogues for examples of incremental grounding behavior. These examples should provide insight into the grounding mechanics that support efficient communication.

The examples that are discussed in this section were taken from the AMI Meeting Corpus [18]. The corpus consists of 100 hours of meetings captured using recording devices of various modalities. In the meetings, a team of four subjects is given the task of designing a new remote control. Each team takes a design from start to prototype in a series of four meetings.

We used the mixed audio signal from the headsets that the subjects wore, the transcription of that signal and occasionally the videos if more information was deemed necessary. One general observation we made during the study was that the interesting interactions were more likely to occur in the last two meetings that a team had. We suspect that the team members by then had become more familiar with each other and rapport had been built.

This section continues with the discussion of several excerpts from the meetings. In those discussions, we take the perspective of the third-party impartial bystander. This differs from the perspective that a computational model of grounding operates from. We cannot peek inside the heads of the interlocutors to figure out the intentions behind their actions. The model on the other hand can only go by its intentions and has to observe the other person's response to figure out the *true* effect of its actions.

3.2.1 Granularity

Humans process language incrementally. This enables us to continuously provide feedback to the speaker or react on that feedback. This feedback is usually given by means of *backchannels*. Backchannels are verbal or non-verbal behaviors, such as head nods, frowns, "Okay", "Huh" or "Yeah", that can be performed in overlap with someone else's turn. The speaker remains in control of the *main channel*, while the listener uses the *back channel* to provide feedback on how well it is going.

The listener's feedback aids the speaker in his performance. The speaker can decide to elaborate on a certain concept if the listener is not understanding [37] or fade out an utterance when it appears the rest is not needed [45].

Frequent overlapping feedback will divide the content of a single turn, or utterance, over multiple smaller CGU's. An overlapping acknowledgment grounds the current open CGU and new content that the speaker continues to introduce after the acknowledgment becomes part of a new CGU, which lasts until the next acknowledgment. This mechanism can be observed in the following example:

(6) A1: [The LCD panel just displays um functionally what you're doing.]$^{\text{Initiate}_I 1}$
A2: [If you're using an advanced function right, like um brightness, contrast, whatever]$^{\text{Initiate}_I 2}$
C1*: [Right]$^{\text{Acknowledge}_R 1}$
C2*: [Okay]$^{\text{Acknowledge}_R 2}$

With C1, C acknowledges A's first utterance. When A continues, he initiates a second CGU, because C's acknowledgment grounded and closed the first CGU. If C would not have uttered C1, A2 would have been a continuation of CGU 1 instead. This would have resulted in one bigger open CGU at the end of A2, instead of two smaller CGUs. The frequent feedback reduces the amount of open content by grounding information early and often. Misunderstandings can also be handled more efficiently, since there is only a limited amount of content that is in the process of being grounded.

Because the size of CGUs is reduced, a single sentence is often represented by multiple CGUs. These smaller CGUs are related to each other, as the first part of a sentence creates expectations of the second part. This relation can be used in overlapping feedback to present evidence of understanding. But first, we need to further specify that relation and its properties.

3.2.2 Evidence of understanding by completion

Consider the following dialogue excerpt from an AMI meeting:

(7) C1: We could just go with um
D1*: Yeah
A1*: Normal coloured buttons

In the middle of C's sentence, C appears to struggle with how to continue his utterance, uttering a verbal hesitation "um". A then utters "Normal coloured buttons" as a completion of C's partial utterance. The dialogue continues without correction by C, so it is reasonable to assume that this was indeed what C intended to communicate (or was close enough). Meanwhile, D gives a simultaneous backchannel acknowledgment of C's utterance.

In a way, A is not making up the words he is saying. It is not satisfactory to say that A presents some new content, or according to Traum's theory, initiates a new CGU. His intention is to utter what C was going to say and while C did not say "normal coloured buttons", A did receive enough evidence leading to this completion. A appears to be able to predict the intended meaning (or perhaps even surface form) of the full utterance based on the partial utterance. We call the content of the partial utterance *explicit* and the content of the utterance completion *predicted*. A provides evidence of understanding by completing the utterance, grounding both the explicit and predicted content. We add *completion* to Clark and Schaefers list of types of evidence of understanding. A1 is thus a acknowledgment grounding act providing evidence by completion. The relevant utterances from dialogue 7 can be annotated with grounding acts as follows:

(8) C1: [We could just go with um]$^{\text{Initiate}_I 1}$
A1*: [Normal coloured buttons]$^{\text{Acknowledge}_R 1}$

It may seem like the listener is making up new content for the open CGU by proposing a completion. The completion is based on a hypothesis of what the speaker was going to say. This is not fundamentally different from a regular response, which is based on the hypothesis of what the speaker said. It is not clear why these two cases should be handled differently.

Attempted utterance completions do not always match a speaker's intended content or surface form, as in dialogue excerpt (9).

(9) B1: That would probably not be in keeping with the um the
C1*: *laugh* Technology
B2: fashion statement and such, yeah.
C2*: Yeah.

In this dialogue, B and C are reflecting on the features and design of the remote control they created. When B shows hesitation ("…with the um"), C decides to help and offers "Technology"

as a completion of B's utterance.[1] B however continues his utterance by saying "fashion statement and such", revealing perhaps more precisely what he intended to say. C then issues an overlapping acknowledgment of B's continuation with "fashion statement", by saying "Yeah". When B finally adds "yeah" to his own continuation, he also shows his agreement with C's continuation, i.e. the fact that the technology is indeed also not kept up with. Based on this analysis, the dialogue excerpt can be annotated with grounding acts as follows:

(10) B1: [That would probably not be in keeping with]$^{\text{Initiate}_I 1}$ the um the
C1*: *laugh* [Technology]$^{\text{Acknowledge}_R 1,\ \text{Initiate}_I 2}$
B2: [fashion statement and such]$^{\text{Initiate}_I 3}$, [yeah.]$^{\text{Acknowledge}_R 2}$
C2*: [Yeah.]$^{\text{Acknowledge}_R 3}$

C's predicted content "Technology" apparently does not exactly match B's original intention. However, it does provide some evidence of understanding of the explicit content of B's partial utterance and grounds CGU1 containing that content. It is a fitting completion to the unfinished utterance, because it is a syntactical and conceptual continuation. The strength of evidence of understanding that a completion conveys depends on the relation between the unfinished utterance and the continuation, being either syntactical, conceptual or both. A *syntactical* continuation provides weak evidence, because it only relates to the surface form. Following Clark's action levels, we can say that syntactical completions provide evidence up to the Signal level (level 2). A *conceptual* continuation operates on the same conceptual level as the unfinished utterance it completes. In the example above, both technology and fashion statement are similar concepts, which are both not satisfied in the remote control design. This type of completions operate on the Intention and Conversation (level 3 and 4) of Clark's action ladder and thus provide stronger evidence compared to completions with a mere syntactical relation to its antecedent.

A special category of incorrect completions that is worth mentioning - but will not be pursued further in this thesis - contains a variety of completions that purposefully deviate from the speaker's intended meaning. Those kinds are used as a joke or as an opportunity to convey your own meaning at the cost of the original speaker. Consider the following hypothetical conversation about dinner between a mother and her child:

(11) Mom1: Tonight we're having
Child1*: pizza, yay!
Mom2: *laugh* You wish, it's green peas for you mister.

The child needs to be aware of what his mother intended to say in order to be able to make a wrong continuation. So by joking "pizza, yay!", the child acknowledges that dinner is going to be green peas, or something healthy at least. Humor identification is a natural language understanding problem (see e.g. [46]) of its own and not yet directly relevant to dialogue systems.

[1] In this thesis, we are ignoring the evidence of understanding that laughter can convey.

3.2.3 Implicit verification of predicted content

In Section 2.6.1, we distinguished between implicit and explicit verification to characterize various grounding strategies. We can use the same distinction to identify different ways of verifying predicted content. The completions as discussed in the previous section are a way to explicitly verify the grounded content. The following dialogue contains an example of implicit verification of predicted content:

(12) B1: [power-wise, have we got]$^{\text{Intiate}_I 1}$
A1*: [The battery.]$^{\text{Acknowledge}_R 1,\ \text{Initiate}_I 2}$
B2: [battery]$^{\text{Acknowledge}_R 2}$ [Do we have kinetic as well?]$^{\text{Acknowledge}_R 2,\ \text{Initiate}_I 3}$
A2*: [No.]$^{\text{Acknowledge}_R 3,\ \text{Initiate}_I 4}$
B3*: [No.]$^{\text{Acknowledge}_R 4}$
B4: [Okay, just battery]$^{\text{Acknowledge}_R 4}$

Before B has finished uttering his question (B1), A goes ahead and answers it. An answer conveys evidence of understanding of the question it answers, if it fits the question, i.e. if it actually answers the question that was posed. A fitting answer to an unfinished question shows understanding of the complete question, because it requires understanding of the full question in order to recognize what answer would fit. In this dialogue, the full question would have been something like, "power-wise, have we got what?" A's "The battery" implicitly acknowledges his understanding of the full question, because it answers that question. The dialogue continues without correction by B, so it can be assumed that A was right in his prediction of B's question. The result is efficient and fluent interaction, achieved through good coordination between A and B.

If such coordination, or rapport, is not present between the interlocutors, this type of interaction is not at their disposal. The following dialogue excerpt displays a failed attempt:

(13) C1: Are these the colours of production, or is this just what we had available?
D1*: Well I'm
D2: We're gonna have again the sort of foggy yellow from last time.

D attempts to answer C's question, but C refuses to give up the turn and continues uttering the remaining part of the question. It might be that C did not believe that D had already sufficiently understood his question. While implicit verification might be efficient for certain cases of grounding, it conveys less evidence of understanding than explicit verification. It will only be effective in cases that require a lower amount of evidence, e.g. based on the grounding criterion and rapport.

3.2.4 Conclusions

In this section, we have studied several displays of incremental grounding behavior in human-human conversations. We have seen that because of frequent overlapping feedback, the rate

at which CGUs are grounded increases and the size, i.e. the amount of content, of CGUs decreases compared to situations where incremental grounding is not used. Single utterances are often represented by multiple small CGUs. On this new scale of sub-utterance CGUs, the CGUs, through their content, are more related to each other. Incremental grounding acts not only acknowledge that what was already said, i.e. the explicit content of an utterance in progress, but may also concern what the speaker is about to say, i.e. the content of the remaining part of the utterance as predicted by the listener.

3.3 A Model of Incremental Grounding

Based on the empirical evidence gathered from the AMI Corpus, we have developed a grounding model for the purposes of more fine-grained incremental processing. The model is adapted from Traum's grounding acts model. It will need to be able to interpret the incremental updates that come in while an utterance is in progress and update the grounding state accordingly. The original model by Traum does not pose any limitations on the size of the processing units, but the typical unit size in existing implementation corresponds to full utterances. This makes sense, because the components of a dialogue system, especially the NLU and DM, would only be able to process full utterances. Now that that is changing, a grounding model with incremental processing capabilities is becoming relevant.

An important requirement of the model is that it should be robust to errors and revisions by upstream components. In Section 3.1, we described several cases in which the ASR or NLU would revise their incremental hypotheses. The ASR could, for example, revoke the last word from the previous hypothesis, or replace it with a different word. In turn, the grounding model has to update its hypothesis, i.e. the grounding state, which could result in a change in CGU content or require reinterpretation of an utterance assuming it conveys a different grounding act all together.

As input to our model, we assume a component with incremental speech understanding capabilities that delivers a finite sequence of incremental hypotheses as an utterance progresses, henceforth referred to as *partials*. Each partial contains both the explicit and predicted content of the utterance at that point in time. (The implementation of this component is discussed in Chapter 4.) We will denote the sequence of partials for an utterance as

$$\mathcal{O} = \langle (E_1, P_1, C_1), ..., (E_N, P_N, C_N) \rangle, \tag{3.1}$$

where E_i is the explicit content and P_i is the predicted content for the i^{th} partial. At each point in time, we assume further that the incremental understanding component is able to assign a confidence level C_i that describes the reliability of its estimates E_i and P_i.

Consider the following example:

(14) D1: So basically the only new thing is the LCD on the remote now.
B1*: Being manipulated by the joystick, yeah.
D2: Oh and the joystick , yeah.

At the moment when B decides to interrupt, let us call this time t, the output by the incremental understander could be: E_t ="So basically the only new thing is the LCD", P_t = "being manipulated by the joystick" and $C_t = 0.7$ [2]. For the sake of this example, we have represented the contents of E and P by surface texts. For a real system, it would be reasonable to assume that the content is similar to elements from an NLU frame (see e.g. Table 3.2). At this moment however, we will not make any assumptions on this matter other than that the content of E and P is a set of zero or more unique elements representing the pragmatic meaning of the utterance.

The task of the grounding model when processing an utterance consists of two steps. The model has to *identify* the grounding acts that are being conveyed by the utterance and the CGUs they relate to, and it has to *update* the affected CGUs accordingly. Updating of the CGU will change its grounding state and may change its contents. The ultimate goal is to have an up-to-date representation of both aspects, so the system knows what is going on and what the effect of overlapping grounding acts at any point during an utterance, e.g. at time t in the example above, would be.

Some grounding acts will only affect the grounding state, such as a request repair, acknowledgment, request acknowledgment and cancel, and others will also change its content, e.g. initiate, continue and repair. We will call the latter category *authorial* grounding acts, as they make the uttering party co-author of the CGU. By becoming an author, the *burden of evidence* shifts to the other interlocutor.

For example, if the responder repairs a CGU, the initiator is required to provide evidence of understanding for that CGU to be grounded. If the initiator however decides, in his turn, to repair the CGU again, he becomes the most recent author and the burden of evidence shifts to the responder. In Traum's original model, these notions are implicitly contained in the four in-progress grounding states (see Table 2.4). In state 1 and 2, the initiator is the *most recent author* and the burden of evidence lies with the responder, an acknowledgment act by the responder from those states will move the CGU to the final state. In states 3 and 4, the situation is the other way around.

3.3.1 Two approaches

We have investigated two approaches to modelling grounding in an incremental dialogue system: a content first approach and function first approach.

In the content first approach, the difference between the explicit content of the partials is used as the main input for the grounding model. A partial is different from its predecessor if

[2]While C_t is made up, its value seems reasonable for this example. B is confident enough to try and propose a completion, but not confident enough to justify an implicit verification.

it contains content that its predecessor did not have or if it lacks content that the predecessor had. The effect that the difference has on the content or creation of CGUs then determines what grounding act the partial conveys. Consider the example in Table 3.3. We again use the surface text to represent the partial's explicit content. The first partial initiates a new CGU and adds the initial content "New". The second partial contains new content ("Utah"), but also lacks content from the previous partial ("New"), so it is repairing the CGU that was created with the previous partial. The CGU now contains "Utah". The third partial has new content in addition to the content from the previous partial. There is no need for a repair, the CGU content can be extended. Thus, the third partial is a continue act.

t	E_t	New content	Removed content	Grounding act
1	New	New		initiate
2	Utah	Utah	New	repair
3	Utah it	it		continue
4	Utah do	do	it	repair
5	Utah do you why	you why		continue
6	Utah do you wanna be	wanna be	why	repair
7	Utah do you wanna be the	the		continue
8	Utah do you wanna meet you sure	meet you sure	be the	repair
9	Utah do you wanna be the sheriff	meet you	be the sheriff	repair

Table 3.3: The first approach to incremental grounding: using the differences between each partial.

With this approach, the grounding model can not distinguish between what the user said and what the NLU understood. An NLU error or the recovery thereof is a repair, and not treated different than an actual repair, when the user actually fixes his previous statement. The lack of a distinction between these cases emphasizes their similarity and relieves the system from having to reliably identify either of the two individually in an utterance.

Compared to Traum's model, this approach models a single utterance as a sequence of grounding acts instead of a single grounding act that covers the function of the complete utterance. The initiation of a CGU is represented by an initiate act, followed by continue and/or repair actions. Repairing a CGU consists of a number of repair and continue acts. This works well for the authorial grounding acts, because they affect the content of the CGUs as they are being conveyed, which can now be processed incrementally. The remaining grounding acts, request repair, acknowledge and cancel do not progress in any way while the utterance is being uttered. The individual partials do not bring any content, but together make for one of the three aforementioned acts, the first partial no different than the last. Following this approach however, every single partial of a cancel conveying utterance is a cancel in itself. Does the second partial's cancel, cancel the first cancel then? Should all partials with the same act be grouped instead? Perhaps, but that would be inconsistent with how the authorial acts are handled. In the second approach that we present next, all grounding acts are processed in a similar and consistent manner.

The second approach favors function over content. It most importantly solves the problem encountered with the first approach, but also takes care of CGU content updates. As long as consecutive partials are part of the same grounding act, they are grouped. In the example from Table 3.3, the model would select the initiate act based on the first partial, and reconsider that hypothesis after each partial, to eventually conclude that all nine partials together form one initiate act. This is the approach that we pursued in our work and will now continue to present.

3.3.2 Identification of grounding acts

The first step in processing an utterance is to identify the grounding acts it conveys. This determines how the content of the utterance is processed. The explicit content and predicted content can be combined into the full utterance meaning: S_t for time t. S_t can be used to for pragmatic reasoning, which is typically a responsibility of the DM, to recognize backward looking speech acts, such as answers, confirmations or clarification requests. These acts can signify acknowledgments or request repairs respectively. We assume a dialogue system that is capable of understanding the grounding acts of Traum's model. What we add is more comprehensive and incremental identification of grounding acts, including a strategy to deal with a change of the hypothesized grounding act.

Following Traum's model, most utterances will only have a single grounding act. Not all combinations of grounding acts are suitable to be conveyed in a single utterance. The model will only support utterances that simultaneously convey an acknowledgment and initiate act. This combination was found most often during our study of the AMI corpus, see e.g. Dialogues 10 and 12.

Starting with S_1, the model will try to assign grounding acts to the utterance. At time t, when C_t is high enough and the model selected the grounding acts for that utterance, the partial will be processed according to the selected grounding acts. The subsequent partials will be processed in the same way (more details follow in the next section), until the end of the utterance or if the selected grounding acts are revised. If the selected grounding acts are revised, e.g. caused by an NLU error or if the user said something unexpected, all updates to the grounding state under the former grounding act selection are reversed, and the current and subsequent partials will be processed according to the new information.

Evidence of understanding

In our study of the AMI Corpus (see Section 3.2), we have encountered several ways to convey acknowledgments that are specific to incremental grounding. Listeners can provide overlapping feedback, continue an unfinished sentence or respond to an unfinished utterance. Only because of incremental processing, these kinds of behaviors can now be identified and used to update the grounding state. We define a set of rules that an utterance can be tested against for the occurrence of those behaviors. Let U be the current utterance, V the previous utterance and E_t^U, P_t^U and S_t^U

the explicit content, predicted content and full content of utterance U at time t respectively, then:

U conveys an acknowledgment if:

there is an open CGU of which the speaker is not the most recent author and at least one of the following is true:

1. S_t^U conveys a positive backchannel (e.g. "Yeah", "Uh-huh", etc.)
2. V is unfinished and E_t^U is a syntactical or conceptual continuation of E_u^V, where u is when U started (*Completion, see Section 3.2.2*)
3. S_t^U contains parts of S_u^V, where u is when U started
   (Explicit verification, see Section 3.2.2)
4. S_t^U is the next relevant contribution to S_u^V, where u is when U started
   (Implicit verification, see Section 3.2.3)

Note that the special case $u = N^V$, where u is the time when U started and N^V the number of partials of the previous utterance, is a regular non-overlapping response. For this special case, 3 is equal to Clark and Schaefer's 'Display' evidence type and 4 is the 'next relevant contribution' type. Non-incremental grounding is a special case of incremental grounding.

3.3.3 Updating the grounding state

Based on the grounding acts that were identified in the utterance, each incoming partial is processed in order to update the grounding state. If more than one grounding act is identified, i.e. in the case of an acknowledgment-initiate combination, each partial is processed for each grounding act separately.

Initiate

The first partial of an initiate act triggers the creation of a new CGU. The ungrounded explicit content of that partial becomes the initial content of the new CGU. Consecutive partials will replace the CGU content with their explicit content. If an overlapping backchannel or request for repair is detected, the initiate act ends. Let the overlapping behavior start at partial t of this utterance U, then content of the CGU will be E_{t-1}^U. In the case of a backchannel, the CGU is grounded, and if the speaker decides to continue uttering U, this will be a new initiate act. The initial content of the new CGU will be the ungrounded content of E_t^U, which is all elements that were not grounded with the previous CGU. A second overlapping backchannel in the same utterance is handled analogously. In the case of a request repair, the continuation of U is a repair.

Continue

The explicit content of continue partials is added to the corresponding open CGU. Changes in the explicit content between consecutive partials will only affect the CGU content that was added during this continue act. This means that the content that was present in the CGU before the continue will remain untouched.

Acknowledgment

The processing of an acknowledgment depends on the type of behavior that is used to convey the evidence of understanding. Let U be the utterance that conveys the acknowledgement, V be the utterance that U is a response to and t be the time of last partial received of V.

Backchannels ground the CGU of V with content E_t^V. The current authorial act in V is ended and if the speaker continues after/during the backchannel, this is a new grounding act. If the same backchannel is covered by more than one partial, the first partial is used for timing purposes.

Completions ground the CGU of V with content E_t^V. V is unfinished. Regardless of whether the completion was actually what the original speaker intended to say, the syntactical or conceptual relation between the unfinished utterance and the completion conveys evidence of understanding. If the completion follows the intentions of the original speaker, the completion is an explicit verification of S_t^U and is processed as described below.

Explicit verifications ground the CGU of V. If V is unfinished, the predicted content of V at the time t of the start of U that is also in the explicit content of U ($P_t^V \cup E_t^U$) is added to the CGU. If V is finished, the content that was in V and is explicit in U ($S_N^V \cup E_t^U$) is grounded as content of the CGU. The second case is the aforementioned non-incremental special case of the first one. Consecutive partials of the same completion (or response if $t = N^V$) will expand the CGU's content, as they will likely contain more explicit content.

Implicit verifications ground the CGU of V with the full utterance meaning S_t^V as its content. Consecutive partials can be used to monitor the progress of the utterance, but do not trigger any additional processing of this acknowledgment. Utterances that are implicit verifications however also present new content - hence the *implicit* - that will initiate a new CGU. Partials of this utterance will therefore also be processed according to the description above.

Request for repair

A request repair changes the state of V's CGU to indicate that a repair from the other party is requested. If the request for repair is overlapping, the CGU's content will be E_t^V and the current

authorial act in V is ended. If the speaker continues after/during the request repair, this is a new grounding act.

Repair

A repair will modify the content of a CGU. For each partial of this repair, the explicit content is added to the CGU. Existing content in the CGU that is not compatible with the content of the repair is removed. We assume that the dialogue manager can reason about this.

Cancel

A cancel act will abandon a CGU and leave it ungrounded. There is no special logic required to handle a cancel in incremental grounding.

3.3.4 Examples

Increased granularity

The following dialogue is annotated with four points in time. We will continue by discussing this dialogue to exemplify what goes on in the model during different moments in the dialogue.

(15)
B1: cause we could ① just sort of say, sorry ③ what did you say about that or \
C1*: ② Yeah.
B1: what do you think about that, ④ rather than having to email it, yeah.
C2*: Yeah.
C3*: Yeah.

At ①, the model has received the first few partials from the NLU. While the first partial may have had a confidence value that was too low to take into account, by now the NLU must have returned a hypothesis that was deemed reliable. E_1 is "cause we could" and P_1 may contain a provisional idea of what the rest of the sentence is going to be. Based on that hypothesis, the utterance is supposed to convey an initiate act. A new CGU is created and assigned E_1 as its content.

From ① to ②, each incoming partial reaffirms the model's identification of the initiate act. As long as this is the case, the content of the CGU is replaced with the partial's, growing, explicit content, until, at ②, the CGU contains E_2^{B1}: "cause we could just sorta say".

At ②, C decides to acknowledge B's utterance so far. The CGU with content E_2^{B1} is grounded.

C's overlapping acknowledgment ends the first initiate act in B1. C continues with the utterance, but this is considered a new initiate act. At ③, a new CGU is created with all ungrounded content from E_3^{B1}, which is "sorry".

From ③ to ④, the content of the second CGU is replaced with the explicit content of every partial that comes in, ignoring the parts that have already been grounded. To sum up, at ④, there is one grounded CGU containing "cause we could just sorta say" and a second CGU with "sorry what did you say about that or what do you think about that", that is about to be grounded by C2.

Grounding predicted content

For this example, we revisit a dialogue that we have discussed before (see 8 on page 29). In the reprint below, we have annotated several interesting points in time.

(16)
C1: We could just go with um ①
D1*: Yeah
A1*: Normal ② coloured buttons ③

Until ①, partials have come in that describe the progress of C1. At ①, the explicit content of the most recent partial is "we could just go with" and the predicted part "normal coloured buttons." Then, A begins to hesitate how to finish C1. We ignore D's "Yeah," because multi-party grounding lies outside the scope of this thesis.

As a response to C's hesitations, at ②, A has started to complete C1. Because A1 appears to be a conceptual continuation of C1, it acknowledges C1. It also adds "normal" to the acknowledged CGU, which is the explicit content of A1 at ② that was also in the predicted content of C1 at ①. More formally, the CGU contents now is $E_1^{C1} \cap (P_1^{C1} \cup E_2^{A1})$.

This continues up to ③, where A1 is finished and all its explicit content, which is equal to the predicted content of C1, is added to the CGU and grounded.

Chapter 4

Incremental Grounding in SASO4

4.1 Overview

We have developed a prototype implementation of the model presented in the previous chapter for the SASO4 dialogue system. SASO4 is being developed at the University of Southern California's Institute for Creative Technologies. It is part of the Virtual Humans Project, a multidisciplinary effort bringing together expertise on all facets of virtual human development.

4.1.1 Institute for Creative Technologies

The Institute for Creative Technologies (ICT) is part of the University of Southern California (USC) [47]. It was established in 1999, as a joint project between USC and the U.S. Army. ICT's mission is:

> "... to conduct basic and applied research and advanced technology development in immersive technologies to advance and maintain the state-of-the-art for human synthetic experiences that are so compelling the participants will react as if they are real."

The work at ICT has resulted in numerous applications that convey this mission: Star Wars-like 3D teleconferencing, virtual museum guides for the Museum of Science in Boston, a portable virtual reality system, rehabilitation therapy tools, Light Stage and virtual reality therapy for treatment of post traumatic stress.

The author of this thesis spent six months at ICT, from November 2011 to May 2012. During that time, he worked in the Natural Language Dialogue group and was supervised by group leader David Traum.

4.1.2 The Virtual Humans Project

In the Virtual Humans Project, a multidisciplinary team of researchers aims at developing autonomous agents that are capable of face-to-face interaction with humans according to a variety of scenario's and tasks [48, 49]. The agents are embedded in a virtual world and can perceive events in their world as well as the real world and the actions of the user(s) in particular.

At the core of the Virtual Humans Project is the Virtual Human Toolkit (VHToolkit), a collection of components, tools and libraries that enable efficient development of virtual humans [50]. The toolkit is freely available for the academic community. For internal projects, the VHToolkit is augmented with experimental components, depending on the focus of the current research. Those components may eventually become part of the VHToolkit.

There are a number of dialogue systems built on top of VHToolkit, such as Gunslinger [9], TacQ [51], SimCoach ([52]) and several generations of SASO [53]. In this thesis, the latest installment of the SASO systems was used. The first SASO system was the SASO-ST system, which is short for Stability and Support Operations - Stability and Simulation. It allowed the user to practice negotiation skills in a conversation with a virtual doctor, dr. Perez, while trying to convince him to move his field clinic to a safer location [54]. This scenario was elaborated on in SASO-EN - Extended Negotiation - by adding a village elder as a second virtual human. This allowed for multilateral negotiations between the user and the virtual humans, allowing the user to gain the trust of the elder and together convince the doctor to cooperate.

4.2 SASO4

The latest installment of the SASO dialogue system is SASO4. This time, two users are emerged in a new, more complex, scenario:

> "An American Old West town has been freed from a dangerous outlaw, defeated by a U.S. Ranger with the help of Utah, the local bartender. The Ranger and his Deputy must now leave town to pursue their mission elsewhere. But before leaving, they need to recruit a town sheriff, so they offer the job to Utah. He will need resources – e.g., money to buy guns and to hire men – guaranteed before considering the offer. As owner of the saloon, Harmony is an influential woman in town. She will be present in the discussions, pushing forward her own agenda of demands, part of which she cannot discuss in front of Utah and must be dealt with in private by one of the officers. The Ranger and the Deputy have very limited resources, so they must negotiate to reach an agreement by committing as little as possible." – from [55]

The two users take on the role of the Ranger and his Deputy. Utah and Harmony (see Figure 4.1) are virtual humans. This scenario includes new complexities such as: the possibility of a 4-party dialogue splitting into simultaneous 2-party dialogues; a greater number of possible

Figure 4.1: Utah and Harmony

solutions to the negotiation problem and the presence of a hidden agenda in a virtual human [55].

The following is an abridged version of a real conversation with SASO[1]. At the start of the conversation, the situation is as shown in Figure 4.1.

(17)

Ranger:	We want to keep this town safe.
Utah:	How may I help you?
Ranger:	What do you think about becoming the new sheriff?
Utah:	Well, that's your job. We don't need another sheriff.
Ranger:	Why don't you wanna be the new sheriff?
Utah:	Deputy, can Ranger give me 150 dollars to become sheriff?
Deputy:	Yeah, we can give you 150 dollars.
Harmony:	Utah, you should stay bartender.
Utah:	Harmony, do you think I could never be anything better than a bartender?
Harmony:	Argh, I can't deal with this.

Harmony walks away, the Deputy follows her and they continue to talk in private.

Deputy:	Why don't you want Utah to be the new sheriff?
Harmony:	I am afraid that Utah will die.
Deputy:	Why?

[1]The version of SASO used is from May, 2012

Harmony: There's a group of outlaws nearby and I am afraid they will kill Utah.
Deputy: We can take care of those outlaws tonight.
Harmony: Deputy, let's rejoin the others

Harmony and the Deputy rejoin Utah and the Ranger.

Harmony: Ranger, I want Utah to become sheriff.
Utah: Okay Ranger, I want to become sheriff.

SASO4 is a *multi-modal dialogue system*, meaning that besides being a spoken dialogue system, it can also perform and recognize non-verbal gestures such as head nods and posture changes. This also means that its utterances can contain a verbal and a non-verbal component and that it can recognize signals from the user that combine the two modalities [56].

4.2.1 Architecture

Most of the components of SASO4 are inherited from earlier SASO versions and extended to meet the new scenario's needs. Since 2009, the system has been adapted to support incremental processing. The following components are relevant to this thesis:

- An automated speech recognizer (ASR) that produces incremental results, configured to do so every 200ms.
- A natural language understanding component (NLU) that comes up with semantic representations based on the incremental ASR results, including the meaning of the utterance so far, a prediction of the full utterance meaning and a confidence score.
- A dialogue manager that can determine the pragmatic meaning of the utterance based on the incremental NLU results, perform domain reasoning and select the system's response act.
- A natural language generator (NLG) that computes a surface text for the selected response act.
- A non-verbal behavior gesture generator (NVBG) that determines behaviors given the functional specification from the dialogue manager.

The components communicate via a shared message bus that is provided through the Virtual Human Messaging System (VHMsg), also part of VHToolkit. System components can subscribe to certain message types and will be notified when the requested messages are sent. This makes the components loosely coupled, which enables the use of virtually any programming language and gives the freedom to run the system on multiple physical machines.

A more elaborate discussion of the current state of the system can be found in [27].

4.3 Implementation

We have created a prototype implementation of the incremental grounding model described in Section 3.3. In a typical dialogue system architecture, this would be part of the dialogue manager, and the current SASO dialogue manager does in fact manage the non-incremental grounding. To adapt the existing model for the incremental context, extensive knowledge of the dialogue manager and the Soar cognitive achitecture [57] that was used to build the dialogue manager would be required. We instead opted to build a separate component that takes over the responsibility from the dialogue manager to keep track of grounding. An overview of the system including our component is displayed in Figure 4.2.

In addition to modeling incremental grounding, the component also executes a simple overlapping behavior policy that showcases up-to-date knowledge of the grounding state. The component selects behaviors according to the policy and instructs the appropriate components to execute those behaviors. Our policy is a rudimentary variation on Wang et al.'s comprehensive listener feedback model [13].

Our component is a proof of concept, exploring the feasibility of incremental grounding in the SASO4 dialogue system. The implementation is not fully functional, but is capable of showcasing isolated examples of incremental grounding. To turn our proof of concept into a working dialogue system, significant effort would be required on our component as well as the dialogue manager. This remains future work. In this section, we describe the planned implementation. In the discussion of parts that have not been implemented, the remaining required effort will be described.

4.3.1 Input

In Section 3.3, we defined the input for our model as a sequence of (E_i, P_i, C_i), where E_i is the explicit content, P_i is the predicted content for the i^{th} partial and C_i a confidence metric of both values. This input is provided by a *hybrid* NLU component, that is both *predictive* in its hypothesis of the full utterance meaning and *non-predictive* in its estimation of the current explicit content[2]. The component was developed by DeVault at ICT, one of the motivations being the results from our study of the AMI Corpus (see Section 3.2) regarding the necessity of a distinction between explicit and predicted content. We created a training corpus to increase the performance of the NLU (see Appendix A).

The NLU component emits `vrNLU` messages for each incremental ASR result it receives. The following is an example `vrNLU` message, formatted for readability:

[2]For a discussion on predictive vs. non-predictive approaches to incremental processing, please refer to Section 3.1.1.

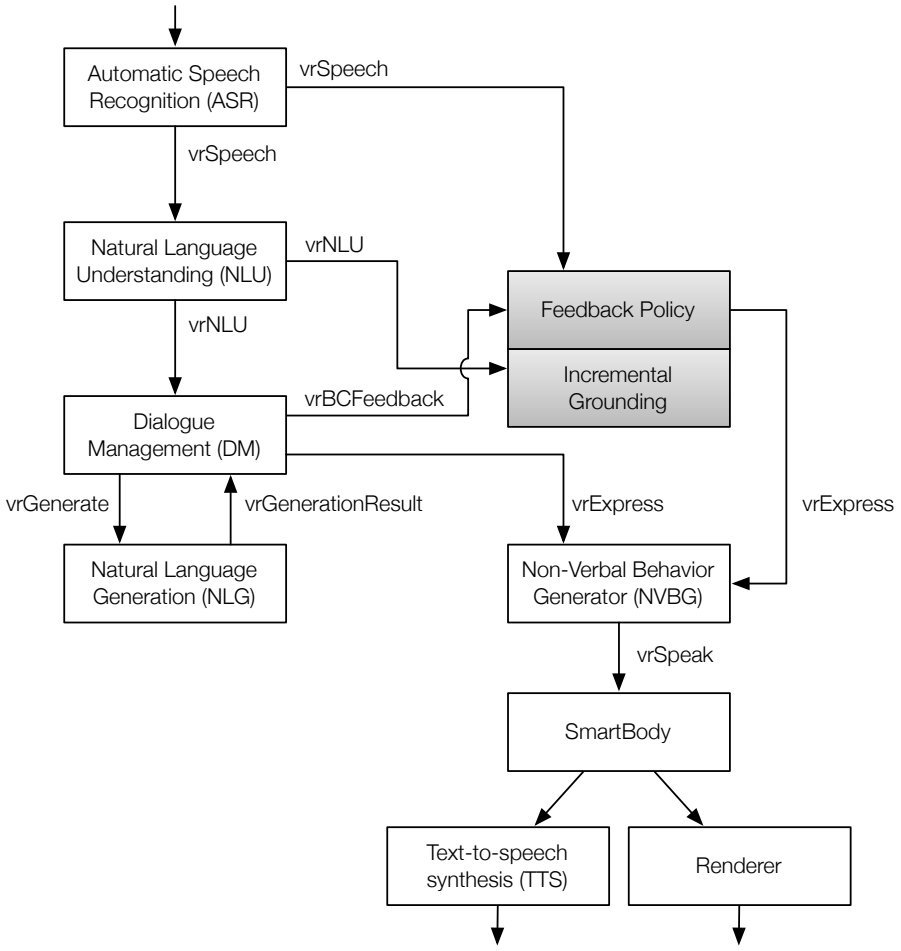

Figure 4.2: Overview of the SASO4 dialogue system. Our components are printed in grey. The annotated lines show the inter-componenent communications over the VHMsg system, the label is the message type.

(18) `vrNLU partial ranger0002 6`

s.mood	declarative
s.sem.agent	you
s.sem.event	providePublicServices
s.sem.modal.desire	want
s.sem.modal.holder	we
s.sem.speechact.type	statement
s.sem.theme	sheriff-job
s.sem.type	event
s.meta.nlu.name	NLUC
s.meta.nluc.subframe.included	true
s.meta.nluc.subframe.threshold	0.50001
s.meta.nluc.subframe.s.mood	declarative
s.meta.nluc.subframe.s.sem.type	event
s.meta.nluc.Incorrect	true
s.meta.nluc.Low	false
s.meta.nluc.High	false
s.meta.nluc.Correct	false
s.meta.nluc.MAXF	false
s.meta.nluc.WillBeIncorrect	false
s.meta.nluc.WillBeLow	false
s.meta.nluc.WillBeHigh	false
s.meta.nluc.WillBeCorrect	true
s.meta.nluc.PF1	true
s.meta.nluc.PF2	true
s.meta.nluc.PF3	true
s.meta.nluc.EF	0.3236286991036415

The message type identifier is followed by `partial`, which indicates that its content is based on a partial speech interpretation. The second argument, `ranger0002`, is the utterance identifier. When a user starts speaking, the ASR will assign a unique identifier to that utterance. The other components will adopt that identifier in their outputs related to that utterance. The third argument is the partial sequence number, in this case indicating that the NLU output is based on the sixth partial speech interpretation. The partial sequence number, together with the utterance identifier, can uniquely identify a partial within the scope of a single conversation.

The remainder of the message is a flattened attribute-value matrix, or NLU frame, containing the predicted full-utterance meaning, the explicit sub-frame and some confidence metrics. The first eight lines are the predicted full-utterance meaning, or S_6 using the notation from the previous chapter. The next attribute, `s.meta.nlu.name`, contains the name of the NLU com-

P	Predicted full utterance
0.3	We can give you 200 dollars
0.2	We can capture the outlaws
0.1	We can give you 100 dollars
0.1	We can give you guns
0.05	We want you to be the sheriff
0.05	We want to keep this town safe
	…

P	Frame element
0.8	**we**
0.7	**can**
0.55	**you**
0.5	give
0.4	dollars
0.3	200
	…

Figure 4.3: **Left:** the n-best list of full utterance NLU frames (represented by their surface texts) for partial ASR result, "We can give." **Right:** Explicit sub-frame element candidates, ranked by the combined probability mass of the frames they occur in. Printed in bold are the elements that would be returned if the threshold is set at 0.50001.

ponent that produced the message. If the NLU was able to come up with a prediction of the explicit content of the utterance so far, `s.meta.nluc.subframe.included` will be `true`. The sub-frame, if present, consists of all the attributes that are prefixed by `s.meta.nluc.subframe`, except `.included` and `.threshold` (which will be explained later). The remaining attributes are are confidence metrics, which are combined by our implementation, the combination acting as C_6, to assess the reliability of the NLU result.

There is a small discrepancy between the input to our model as defined in the previous chapter and the actual content of the `vrNLU` message. We have defined the input as (E_i, P_i, C_i), i.e. the explicit content, the predicted content and a confidence metric. The message contains (E_i, S_i, C_i), i.e. the explicit content, the full utterance meaning and a confidence metric. In Section 3.3, we defined $S_i = E_i + P_i$. Consequently, we can get P_i from the NLU's output by subtracting the explicit content from the predicted full utterance meaning: $P_i = S_i - E_i$.

While the NLU component could have calculated P_i from S_i and E_i itself, the current output is characteristic to the approach applied in this component. It is an extended version of the strictly predictive NLU presented in [11, 17, 33, 34, 41, 42], that was developed to predict the full utterance meaning given a partial speech interpretation. The original implementation used maximum entropy classifiers to generate an n-best list of all the NLU frames[3] in the domain. The top NLU frame, i.e. the most probable NLU frame, would be returned as the result.

The hybrid NLU component computes the explicit sub-frame from the n-best list of frames. It selects individual frame elements that occur in any number of frames on the list that together meet a certain probability threshold. The NLU result in (18) reports that a threshold of 0.50001 was used, which means that the frame elements in the returned explicit sub-frame are more likely to be in the final frame than not. The threshold can be changed to alter the outcome: with a low threshold, the system will lean towards overestimation of the explicit content, while

[3]The SASO4 domain consists of 45 frames.

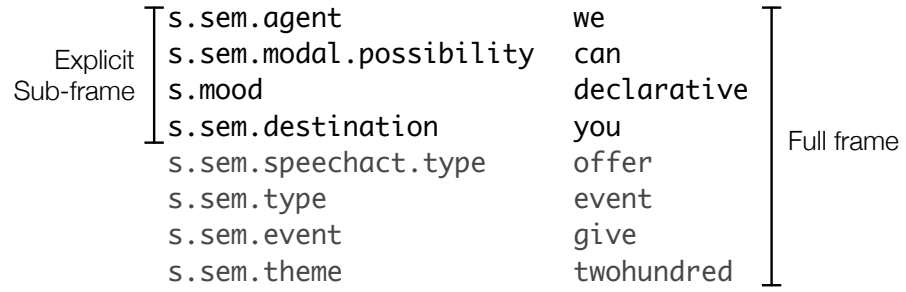

Figure 4.4: The NLU frame output from the example in Table 4.3 consisting of the predicted full frame and a sub-frame representing the explicit content, printed in black.

a higher threshold results in a more conservative estimation. Figure 4.3 shows how the explicit content frame elements are selected for a partial ASR result containing, "We can give". In this example, the predicted full utterance meaning would be "we can give you 200 dollars" and the explicit sub-frame "we can you". Note that we represent the frames using their surface text and the frame elements using similar words to enhance readability. In the actual system, the output would be as depicted in Figure 4.4.

As we take a closer look at inner workings of the NLU, we can also see some of its limitations. In the example discussed above, `s.sem.destination you` erroneously showed up in the explicit sub-frame. This is because utterances starting with "We can give" have a high probability of addressing the listener in the second person, i.e. "you." The estimation of the explicit sub-frame relies on the availability of multiple continuations to the partial utterance. If those are unavailable, the explicit sub-frame will be overestimated. For example, if the partial ASR result is "Utah, do you want to", there is only one possible continuation in the SASO4 domain: "be the new sheriff." Consequently, in the NLU result of the aforementioned partial, the predicted full frame and explicit sub-frame will both be the complete utterance meaning.

In Figure 4.2, it can be seen that our feedback policy also takes `vrSpeech` and `vrBCFeedback` as input. The latter provides addressee information that is taken into account when planning the behaviors, but lies outside the scope of this thesis and will not be elaborated on. The former is discussed below.

4.3.2 Model implementation

Identification of grounding acts

Our implementation in its current state only works on a per-utterance level. It will process every utterance by the user as if it is an initiate grounding act. Overlapping acknowledging behaviors by the system, as coordinated by the feedback policy, may split the utterance's content over multiple CGUs that have to be grounded individually.

The identification of grounding acts can be extended by incorporating the pragmatic reason-

ing that the dialogue manager (DM) performs. The SASO4 DM is capable of detecting initiate, continue, request repair, repair, acknowledge and cancel grounding acts using speech act analysis and keyword spotting. While our implementation is a separate component, as explained above, it ultimately should be integrated in the DM and thereby leverage its knowledge.

Updating the grounding state

In Section 3.3, we defined the input E and P as a set of unique elements representing the content of the utterance without making assumptions of what those elements are. Those elements become the content of CGUs. We have since defined the specific input format for our model in SASO4 (see Section 4.3.1) as NLU (sub-)frames. The content of CGUs in our implementation therefore consists of NLU frame elements, such as `s.sem.event providePublicServices`. We continue by providing an example of how an utterance is processed.

In this example, we show how the grounding model processes the "Utah we can give you two hundred dollars". In overlap with the utterance, the system makes Utah perform two head nods, one after "Utah" and one after "you." Those head nods ground the explicit content of the utterance up to the point of the head nod. The model input during the utterance is displayed in Table 4.1, together with the partial ASR transcriptions that the NLU had to work with.

The first partial is also the first partial with a non-empty explicit sub-frame. Our component creates a new CGU (CGU1) and adds `s.addressee utah` as its first content. The next two partials contain no new explicit content, so nothing happens. After the third partial, the system makes Utah perform a head nod. This is an acknowledgment grounding act, which grounds CGU1. In partial 5, the explicit sub-frame is extended with two new elements. Note that NLU incorrectly predicts the full utterance meaning to be "Utah we want you the be sheriff," however, the explicit sub-frame is correct. A new CGU (CGU2) is created and those two new elements are added to the CGU. After the sixth partial, which provided no new information, Utah performs the second head nod, grounding CGU2. At partial 7, the NLU switches its hypothesis on the full utterance meaning to the correct one. It also overestimates the explicit sub-frame to the full utterance. The six new explicit frame elements are added to a new, third CGU (CGU3). For the remaining partials of the utterance, the explicit sub-frame remains the same. Thus, at the end of the utterance, there is only the content in CGU3 left to be grounded by the system's response.

Note that the first explicit element, `s.addressee utah`, has disappeared from the final NLU hypothesis. There exists no frame in the SASO4 framebank that is about offering 200 dollars and has Utah as the addressee. CGU1 contained this element and was grounded with Utah's first nod. The system has grounded content that, in retrospect, was not correct.[4] There is not much the system can do to correct this, but by tweaking the explicit sub-frame threshold to have a more conservative result, the number of errors like this can be reduced.

[4]Utah is addressed in the utterance, so it does not seem as incorrect content. However, our model only has the NLU to go by, and the NLU does not have that content in its final hypothesis.

Partial	ASR transcription	NLU result	
1	THE	* s.addressee	utah
2	UTAH	* s.addressee	utah
3	UTAH	* s.addressee	utah
	…		
5	UTAH WE CAN GIVE YOU	* s.addressee	utah
		* s.mood	declarative
		* s.sem.type	event
		s.sem.agent	you
		s.sem.event	providePublicServices
		s.sem.modal.desire	want
		s.sem.modal.holder	we
		s.sem.speechact.type	statement
		s.sem.theme	sheriff-job
6	UTAH WE CAN GIVE YOU	*ditto*	
7	UTAH WE CAN GIVE YOU TWO	* s.mood	declarative
		* s.sem.type	event
		* s.sem.agent	we
		* s.sem.event	give
		* s.sem.destination	you
		* s.sem.modal.possibility	can
		* s.sem.speechact.type	offer
		* s.sem.theme	twohundred
	…		
9	UTAH WE CAN GIVE YOU TWO HUNDRED DOLLARS	*ditto*	

Table 4.1: Model input for the utterance "Utah we can give you two hundred dollars." Frame elements marked with an asterisk are part of the explicit sub-frame.

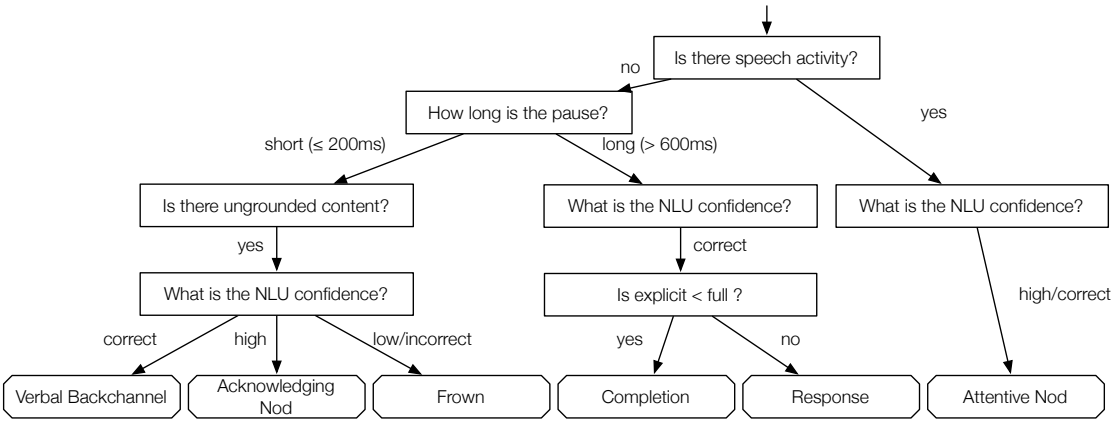

Figure 4.5: Overview of the feedback policy

4.3.3 Feedback policy

We have developed a simple feedback policy that defines various overlapping behaviors and the conditions for their execution. For the evaluation of those conditions, the incremental results from several components, including the incremental grounding model, are queried. Each behavior is specified by a functional and behavioral component. When selected, the behavioral component is executed and the functional component is processed by the incremental grounding model for grounding acts. An overview of the policy is displayed in Figure 4.5.

Conditions

Each node in Figure 4.5 is a condition that is evaluated after each partial is processed. During the evaluation, the ASR, NLU and incremental grounding model are consulted. We continue by describing each condition.

Speech activity The speech activity is evaluated using the incremental results emitted by the ASR. Every 200ms, the ASR sends out a `vrSpeech` message. For example:

(19) `vrSpeech partial ranger0001 9 1.0 normal DON'T OF UTAH`

This message is the ninth partial of utterance `ranger0001`. The fourth and fifth attributes (`1.0` and `normal`) are prosodic characteristics that are currently not in use. The remainder of the message is the partial transcription. We compare consecutive transcriptions to determine the speech activity. If the transcription changes, the user is speaking, if the transcription remains stable over several partials, the user is not speaking. Table 4.2 contains an example of the evaluation of the relevant conditions.

Humans use a pause by the speaker as an opportunity to provide feedback [58]. We adopt this strategy and therefore have this condition that tracks the pauses in the user's speech. We

Partial no.	Partial transcription	Activity	Short	Long
1	UTAH	yes	no	no
2	UTAH	no	yes	no
3	UTAH WE	yes	no	no
4	UTAH WE CAN	yes	no	no
	...			
7	UTAH WE CAN GIVE YOU TWO ARE	yes	no	no
8	UTAH WE CAN GIVE YOU TWO HUNDRED	yes	no	no
9	UTAH WE CAN GIVE YOU TWO HUNDRED	no	yes	no
10	UTAH WE CAN GIVE YOU TWO HUNDRED	no	yes	no
11	UTAH WE CAN GIVE YOU TWO HUNDRED	no	no	yes

Table 4.2: An example of speech activity evaluation based on ASR partial transcriptions

assume that the pauses signify some kind of boundary, separating one part - which might be the same as what Clark calls *installments* [22] - of the utterance from the next. At those boundaries in the utterance, we think the surface text can be mapped best to a set of NLU frame element. A mapping without *dangling* words that are not being represented by a frame element or the other way around, a frame element that describes more than that what has actually been said so far.

Ungrounded content The incremental grounding state is consulted to determine whether there is content available that can be grounded by a system acknowledgment. This is the case for open CGUs for which the system bears the burden of evidence.

NLU Confidence The state of the system's understanding of the user is evaluated using the confidence metrics provided by the NLU. The NLU component supplies us with thirteen metrics, of which we use four: `incorrect`, `low`, `high` and `correct`. These metrics are each backed by a classifier that has been trained to estimate the reliability of a prediction given the incremental ASR result. If `incorrect` is true, then the prediction is definitely incorrect. If `low` is true, the prediction is correct with a certainty of less than 50 %. If `high` is true, the prediction is correct with a certainty of at least 50%. Finally, if `correct` is true, the NLU is completely sure that its prediction is correct.

Utterance progress By comparing the predicted full utterance content with the explicit content, the system can keep track of the utterance progress. As long as the explicit content is a strict sub-set of the predicted full utterance content, the utterance is not complete and the user is expected to continue to talk. If the explicit content is equal to the full utterance content, the user has said everything the system expected him to say.

```
<act>
   <fml>
      <intention>
         <dialogue-act type="acknowledge">
            <attribute name="actor">utah</attribute>
            <attribute name="addressee">ranger</attribute>
         </dialogue-act>
      </intention>
   </fml>
   <bml>
      <speech id="bc1" type="application/ssml+xml">Yeah</speech>
   </bml>
</act>
```

Listing 4.1: Verbal backchannel specification

Figure 4.6: Utah performing an acknowledging nod, or wiggle.

Behaviors

The definition of each behavior consists of the functional specification, i.e. what it does, and a behavioral specification, i.e. what it looks like. Both aspects of the definition are shared with other components that aid in the realization or observe the selected behaviors. Conforming to existing SASO practice, the behavior definitions fit within the SAIBA framework [59]. The function of a behavior is defined using the Function Markup Language (FML) [60], and the realization is described in Behavior Markup Language (BML) [61].

Verbal backchannel When the user pauses for a short while and there is content that the system can ground and thinks it understands well, i.e. the NLU's `correct` confidence metric is true, the system will select a verbal backchannel to be executed. The backchannel, e.g. "Yeah" or "Okay," acknowledges the ungrounded content. The behavior specification is displayed in Listing 4.1. The content of the `fml` element describes the type of dialogue act, in this case an acknowledgment grounding act, it conveys, in this case from Utah to the Ranger. The content of the `bml` element describes that in order to execute the dialogue act, Utah needs to say "Yeah." An ensemble of components, i.e. Smartbody [62], NVBG [63], the renderer and the speech synthesizer, will take care of the realization of the behavior.

Acknowledging nod The acknowledging nod is similar to the verbal backchannel in that it also conveys evidence of understanding. The evidence is slightly weaker, as it is more subtle than its

```
<act>
   <fml>
      <intention>
         <dialogue-act type="acknowledge">
            <attribute name="actor">utah</attribute>
            <attribute name="addressee">ranger</attribute>
         </dialogue-act>
      </intention>
   </fml>
   <bml>
      <head amount="1.0" start="0" type="WIGGLE"/>
   </bml>
</act>
```

Listing 4.2: Acknowledging nod specification

verbal counterpart. The acknowledging nod is selected instead of the verbal back channel when the NLU has a `high` understanding, but it is not completely sure that it is `correct`. However, our grounding model does not take the strength of the evidence into account. The acknowledging nod and verbal backchannel are therefore processed in the same way.

Listing 4.2 contains the specification of this behavior. The FML part is equal to the verbal backchannel. The BML defines a *wiggle* head animation to be executed immediately. A wiggle is two continuous head nods with decaying amplitude, see Figure 4.6. It is one of the nod types identified in [15].

Frown If there is groundable content that NLU is not confident of, i.e. `low` or `incorrect` is true, a frown is issued. The frown shows the system's lack of understanding and is therefore a request repair grounding act. The FML part of the specification describes a dialogue act of type request repair, which the grounding model will process accordingly. The BML specification consists of two `face` elements, that define a change in the facial expression using the Facial Action Coding System (FACS) [64]. The first `face` element lowers the brows (Action Unit 4) and the second tightens the eye lids (Action Unit 7). The result is shown in Figure 4.7.

Completion If the user pauses for more than 600ms in the middle of an utterance, the system will help by completing the utterance. It is a first step in the direction of dialogue systems that interrupt the user. While the system takes over the turn without the user explicitly releasing it first, this kind of interruptions is meant to help the user and not to enforce a dominant position [65].

The completion is a continuation of the user's unfinished utterance and contains the content that the user has not uttered yet. The system takes the predicted content, P_t, t being the most recent partial with activity, and instructs the natural language generator (NLG) to come up with a surface text for P_t. The completion provides evidence of understanding of S_t.

```
<act>
   <fml>
      <intention>
         <dialogue-act type="request-repair">
            <attribute name="actor">utah</attribute>
            <attribute name="addressee">ranger</attribute>
         </dialogue-act>
      </intention>
   </fml>
   <bml>
      <face amount="1.0" start="0.0" end="4.0" au="4" side="BOTH" type="facs" />
      <face amount="1.0" start="0.0" end="4.0" au="7" side="BOTH" type="facs" />
   </bml>
</act>
```

Listing 4.3: Frown specification

Figure 4.7: Harmony frowning

Figure 4.8: Utah performing two consecutive attentive nods.

```
<act>
   <fml>
      <listenerFeedback speaker="ranger" polarity="positive"
         agreement="neutral" uttid="ranger0004" />
   </fml>
   <bml>
      <head amount="0.4" start="0" type="NOD"/>
   </bml>
</act>
```

Listing 4.4: Attentive nod specification

A fully functional implementation of completion is not present in our prototype and will remain future work. Further development will require changes to the dialogue manager and NLG in order for the system to be able to come up with a surface text for partial frames. [17] explored an approach using the SASO-EN domain, which could be translated to the SASO4 domain with some additional work.

Response If the user pauses at the end of the utterance, the system can go ahead and give its response. The SASO4 dialogue system uses a push-to-talk speech interface, where the user presses and holds a button when he speaks. The time between the user being finished talking and him releasing the button is free for the system to use for planning and execution of its response. Without incremental processing, the system had no way of knowing whether the user is done talking. Now, the system can compare the estimated explicit content with the prediction of the full utterance meaning to identify a complete utterance without the user having to explicitly release the turn.

In our policy, the system will only select this behavior if the user has stopped talking for more than 600ms. While this does diminish the advantages of this, in theory, low-latency response technique, we feel that the lack of a comprehensive turn-taking model (see e.g. [66, 67]) and the NLU's tendency to overestimate the explicit sub-frame near the end of an utterance (as discussed in Section 4.3.1) prevent a more assertive approach.

Attentive nod If the user is talking and the system is understanding, it will signal this by performing an attentive nod. The system thereby shows that it is attending to the user's contribution, providing some evidence of understanding. In contrast to the other behaviors, which are selected

only once every time the conditions are satisfied, the attentive nod will go on as long as the conditions are met. As a result, the agent will nod slowly nod as long as the user speaks and the system understands (see Figure 4.8).

The evidence of understanding provided by this nod is the weakest that Clark and Schaefer define: continued attention (recall Table 2.1 on page 14). Our grounding model does not differentiate between the types of evidence and their varying effect on the *degree of grounding* of content. Because continued attention alone often does not sufficiently ground content, we do not interpret this behavior as an acknowledgment grounding act, but present it as an incentive for the user to keep talking. Consequently, the FML definition of the acknowledging nod contains no dialogue act. Instead, we use the `listenerFeedback` element from [13] (see Listing 4.4).

Chapter 5

Conclusions and Future Work

For this thesis, we have set out to develop a model of incremental grounding. We have conducted a literature study on incremental processing in spoken dialogue systems to get an insight in the new challenges and opportunities that incremental processing brings to the development of dialogue systems. We have studied human-human dialogues in the AMI Meeting corpus to find exhibits of incremental grounding behavior between humans that previously could not be supported by non-incremental dialogue systems.

In this chapter, we provide a summary of our work and make some concluding remarks. We end this chapter, and the main part of this thesis, by discussing several directions for future work.

5.1 Results and Conclusions

Spoken dialogue systems have until recently upheld the simplifying assumption that the conversation between the user and the system occurs in a strict turn-by-turn fashion. In order to have more human-like, fluent conversations with computers, a new generation of spoken dialogue systems has arisen that is capable of processing the user's speech in an incremental way. Incremental dialogue systems start processing while the user is still talking, which allows them to consider and execute overlapping behaviors such as backchannels and interruptions.

We have studied the AMI Meeting Corpus in order to identify ways of grounding in human-human dialogue that a non-incremental dialogue system would not be capable of. These incremental grounding behaviors[1] include overlapping feedback, completing unfinished utterances that were initiated by the other party and responding to an utterance before it is completed.

We have observed overlapping behaviors that acknowledge the part of the utterance that has been uttered so far. Each part is its own Common Ground Unit (CGU), which is the unit by

[1] The distinction between non-incremental and incremental grounding originates from the limitations of typical spoken dialogue systems and does not exist for humans. However, when developing spoken dialogue systems, it may be useful to look at human-human dialogue while taking those traditional limitations into account and distinguish between the two.

which content is grounded. In the incremental context, the CGUs are smaller, but at the same time appear to sometimes contain the content of the whole utterance, part of which has not been uttered yet. Based on the first part of an utterance, the listener may be able to tell what the remaining part will be. Through overlapping responses, halfway in the utterance, the listener can show his/her understanding of the complete utterance, including the part that has not yet been uttered. The listener can continue an unfinished utterance if the speaker has trouble finding the right words or decide to answer a question before it has been fully uttered.

To be able to mimic the human capability of predicting the meaning of an utterance before it has been fully uttered, a dialogue system needs to detect both the explicit and predicted content of an utterance incrementally. Based on that information, an up-to-date model of grounding can be maintained, charting the process by which interlocutors cooperate to add understood content, whether it has been uttered or predicted, to their common ground.

In the incremental context, new content comes in small bits, based on the incremental ASR hypotheses. The ASR hypotheses are based on small units of speech signal and are less reliable than full utterance transcriptions. Between updates, the ASR could add, remove or replace words in its hypothesis. The grounding model needs to reflect these changes, in order to be able to correctly process and generate overlapping grounding acts, such as acknowledgments (evidence of understanding) and request repairs (evidence of non-understanding).

We have developed a model for incremental grounding that takes incremental updates consisting of both the explicit and predicted content as input. We have defined how grounding acts can be identified incrementally and how the grounding state, the collected contents and progress of the CGUs, is updated accordingly. We defined new types of acknowledgments and how they affect the content of the CGU they ground, e.g. answering an unfinished question also grounds the part of the question that was not uttered. We implemented our model in the SASO4 dialogue system as a proof-of-concept of our approach, showcasing an up-to-date grounding state through the execution of a simple overlapping feedback policy.

5.2 Future Work

5.2.1 Implementation and evaluation

It should be clear from the discussion of our implementation in Chapter 4 that the result is not a fully functional dialogue system with incremental grounding. A tighter integration with the dialogue manager is required for the identification, and subsequent processing, of all grounding acts. Also, the 'Complete' and 'Respond' behaviors of the feedback policy have yet to be implemented. The 'Complete' behavior requires a natural language generation component that is capable of generating a surface text for a sub-frame, i.e. the predicted part of an utterance. Such a component can be trained using an aligned framebank, containing a hand annotated mapping between words in the surface text and the frame elements that represent those words (similar to

the frame in Table 3.2 on page 26). An effort to create such a framebank was initiated by the author, but has yet to be finished.

A finalized implementation of the incremental grounding model could be used in a dialogue system to gather a corpus of interactions with the system that can be used to evaluate the effect of our work on the efficiency and naturalness of the conversation.

5.2.2 Degrees of grounding

In Traum's model of grounding, a CGU can be in either of three states: ungrounded, in the process of being grounded and grounded. By providing evidence of understanding, the interlocutors ground content, but only if that evidence is strong enough. The type of content, the importance of it being fully understood, shared experiences between the participants, etc. together determine what evidence strength is enough, i.e. the grounding criterion. Evidence that is too weak will not ground the content and evidence that is strong enough will. We took Traum's model as a starting point for our work and therefore follow the same principle. As a result, our model is ignoring the evidence of understanding of the attentive nod from our feedback policy. The evidence is too weak for most cases and therefore we err on the side of not grounding, effectively throwing away the evidence the attentive nod conveys.

In [3, 68], Roque presents an extension to Traum's theory that adds degrees of grounding to the model. In the proposed extension, the state of a CGU depends on the type of evidence provided, registering all evidence types, weak and strong. In the initial phase of this thesis, Roque's work was considered as the basis for our work. However, we considered it is not specifically related to incremental processing and therefore was distracting us from the focus of our work. In a continuation of our work, Roque's adjustments to Traum's theory could be merged with our contribution to form a comprehensive grounding model.

5.2.3 Continuous processing

We have been talking about incremental processing as the early processing of parts of a whole, i.e. an utterance. This implies that that 'whole' has something more to offer than its individual parts. In SASO4, the end of an utterance offers the certainty of the final interpretation. At the end of an utterance, the ASR gives its final transcription and the Natural Language Understanding component (NLU) its final meaning representation. That final result is what the components stick with or, in terms of Schlangen and Skantze, *commit* to [40].

There are two issues that make this situation confusing. The first is that in SASO4, a single utterance always has to correspond to a single NLU frame. Nothing is committed until the end of the utterance. The second issue is that a commit at the end of an utterance implies that the final interpretation is different from the partial interpretations. Can it not be that the next utterance changes the way the utterance before has to be interpreted? It may be that both issues are actually one and the same and that together they point out the transitional nature of the

incremental dialogue processing movement. Incremental processing marks the middle between the rigid utterance-by-utterance processing and continuous processing.

In continuous processing, the input is a continuous audio signal, without the artificial source of certainty provided by the user releasing the push-to-talk button. Automatic Speech Recognizers (ASR) evolved from being able to detect individual words to being able to detect a sequence of words. An ASR treats each piece of audio signal as both an additional part of the previous word and the first part of the next word, resulting in many possible transcriptions. This is called the ASR *lattice*, from which the most probable outcome can be selected. This principle can also be applied to the NLU [69], i.e. treating each word as both an addition to the current frame and the first word of the next frame. From these elaborations, the NLU can select the most probable stream of frames for the continuous speech signal instead of a single one per utterance. This is an interesting direction to pursue in the near future.

Appendix A

Paraphrase Corpus

The primary input of our implemented model comes from the Natural Language Understanding (NLU) component and, indirectly, the Automatic Speech Recognizer (ASR). In contrast to the older SASO-EN domain, the new SASO4 domain that we have been using has not yet been used in experiments with real humans. As a consequence, there was no corpus of user interactions available to train the NLU and ASR. We have created the SASO4 Paraphrase Corpus for that purpose.

The corpus consists of 899 user utterances, each corresponding to one of the 45 frames in the SASO4 domain. The utterances were gathered with the help of five participants, including the author.

A.1 Method

Each participant took part in a recording session that was identical for all participants. During the session, the SASO4 frames and their gloss were presented one-by-one on a screen (see Figure A.1). The decision to display the gloss was made so that participants would not have to be experts on the frame format. The downside is however that the participants would paraphrase on the gloss rather than on the frame. For example, there is a frame that represents the promise from the Deputy to capture the outlaws. The gloss for that frame was "We can capture the outlaws tonight." The gloss mentions 'tonight', while this is not in the frame. As a result, participants less familiar with the frame format often mentioned 'tonight' in their paraphrases. This is the case for 64% of the paraphrases of this frame. This introduces a bias in the NLU's recognition, but not showing the gloss would have complicated the collection of this corpus.

For each frame, the participant was asked to record five paraphrases, i.e. different ways to *say the frame* (see Figure A.2). A total of 225 paraphrases were collected during each session, which on average lasted between 20-30 minutes.

After each session, the author would spend approximately one hour transcribing the para-

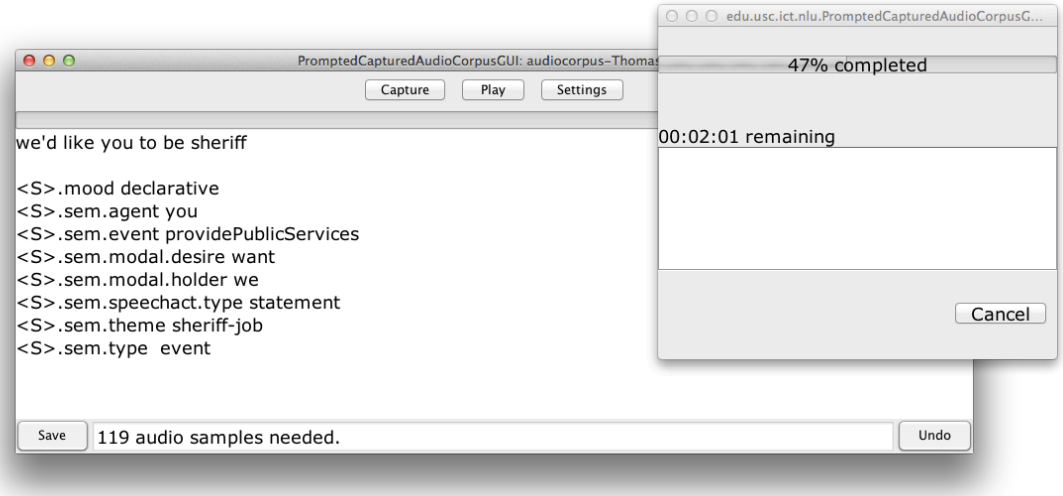

Figure A.1: A screenshot of the tool that was used to record the corpus.

phrases.

A.2 Statistics

Completeness

We noticed that participants would start repeating other participants or even themselves. We assumed that, eventually, after a certain number of participants, the contribution of the next participant, i.e. the number of new, unique, paraphrases, would drop under a level where it would not make sense to invite more people. We therefore kept track of the number of new unique paraphrases each participant added to the corpus, watching for signs that would indicate that we were reaching a plateau of completeness. We have included several statistics on the individual contribution by the participants as well as the cumulative body of paraphrases in Table A.1. To accommodate for small transcription inconsistencies, we also calculated the number of paraphrases that have a Levenshtein character distance of more than three, when compared to all other paraphrases. The statistics show that there is no significant decline in the number of new unique paraphrases and that additional participants are expected to add a significant contribution to the corpus.

Performance

A second measure that can help to decide whether or not to invite new participants is the impact of the growing corpus on the performance of the ASR and NLU. The performance of both components was measured after each participant by cross-validation. Table A.2 contains the ASR's

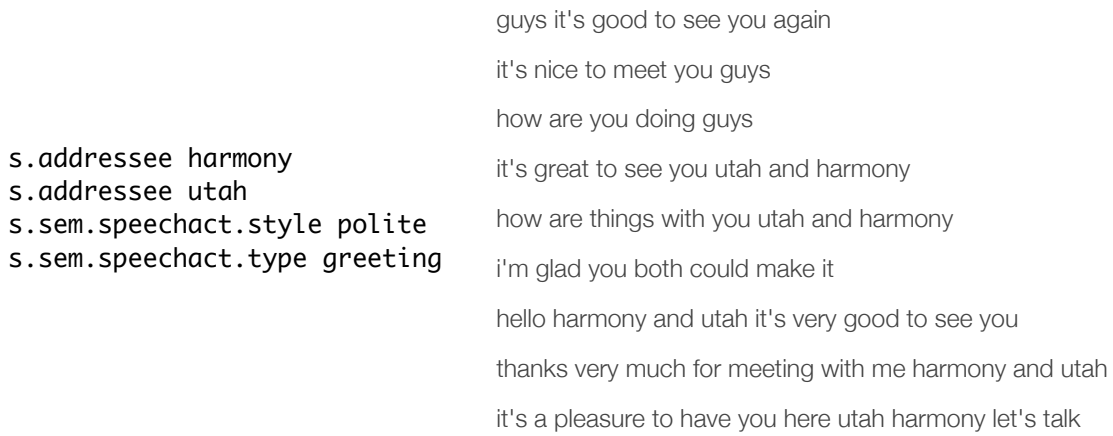

Figure A.2: An example from the corpus. On the left, a frame from the SASO4 domain is displayed. On the right, a sample of the paraphrases we recorded for that frame is shown.

	Participant				
Paraphrases	1	2	3	4	5
Recorded (cumulative)	225	450	675	900	1125
Unique (cumulative)	212	386	564	726	899
Dist. > 3 (cumulative)	192	356	520	668	825
Recorded (individual)	225	225	225	225	225
Unique (individual)	212	174	178	162	173
Dist. > 3 (individual)	192	164	164	148	157

Table A.1: Some statistics of the individual contributions and cumulative body of (unique) paraphrases.

	Participant				
	1	2	3	4	5
ASR word error rate	0.30	0.21	0.30	0.33	0.31
NLU F-score	0.71	0.68	0.78	0.64	0.78

Table A.2: The performance of the ASR and NLU determined by cross-validation on the corpus.

word error rate and NLU's F-score. The data is however inconclusive and more participants would be needed to be able to say something about a good corpus size for optimal performance.

Ambiguity

Ambiguity can be introduced in the corpus if a single utterance has been recorded as a paraphrase of multiple different frames. We found that there are nine ambiguous utterances, all are linked to two frames. Vice versa, we found a total of 12 frames that have paraphrases that are also linked to a different frame. An overview of all ambiguous utterances can be found in Table A.3

Utterance	Frames	
much obliged	s.sem.speechact.type	thank
	s.sem.speechact.style	polite
	s.sem.speechact.type	greeting
okay	s.sem.speechact.type	accept
	s.sem.speechact.type	acknowledge
harmony how are you doing	s.addressee	harmony
	s.sem.speechact.type	greeting
	s.addressee	harmony
	s.sem.speechact.style	polite
	s.sem.speechact.type	greeting
hey harmony	s.addressee	harmony
	s.sem.speechact.type	greeting
	s.addressee	harmony
alright	s.sem.speechact.type	accept
	s.sem.speechact.type	acknowledge
say again	s.sem.speechact.type	no-ack
	s.addressee	harmony
	s.sem.speechact.type	no-ack
hi harmony	s.addressee	harmony
	s.sem.speechact.style	polite
	s.sem.speechact.type	greeting
	s.addressee	harmony
	s.sem.speechact.type	greeting
hey utah	s.addressee	utah
	s.sem.speechact.type	greeting
	s.addressee	utah
how are you doing	s.sem.speechact.type	greeting
	s.sem.speechact.style	polite
	s.sem.speechact.type	greeting

Table A.3: A list of all the ambiguous utterances and the frames they are linked to.

Appendix B

Origin of Dialogue Excerpts

1 from AMI meeting ES2002d, around 06:19

2 from [20], originally from the TRAINS project [26]

3 from [35]

4 from [35]

5 based on a dialogue from [35]

6 from AMI meeting ES2002d, around 2:00

7 from AMI meeting ES2002d, around 11:20

8 see **7**

9 from AMI meeting ES2003d, around 15:55

10 from AMI meeting ES2003d, around 15:55

11 an hypothetical dialogue, made up for the purpose of this thesis

12 from AMI meeting ES2002d, around 07:00

13 from AMI meeting ES2002d, around 03:52

14 from AMI meeting ES2003d, around 16:03

15 from AMI meeting ES2004d, around 28:18

16 see **7**

17 from an actual conversation between the author and the SASO4 dialogue system

Bibliography

[1] H. Clark and E. Schaefer, "Contributing to Discourse," *Cognitive Science*, vol. 13, no. 2, pp. 259–294, 1989.

[2] H. Clark and S. Brennan, "Grounding in Communication," *Perspectives on Socially Shared Cognition*, vol. 13, no. 1991, pp. 127–149, 1991.

[3] A. Roque, "Dialogue management in spoken dialogue systems with Degrees of Grounding," Ph.D. dissertation, University of Southern California, Los Angeles, 2009.

[4] D. Jurafsky, J. Martin, and A. Kehler, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. MIT Press, 2002, vol. 2.

[5] Google, "Google Translate," http://translate.google.nl/.

[6] "Apple - iOS 6 - Use your voice to do even more with Siri," http://www.apple.com/ios/siri/, 2012.

[7] Wolfram | Alpha LLC , "Wolfram|Alpha," http://www.wolframalpha.com.

[8] J. Edlund, G. Skantze, and R. Carlson, "Higgins – a spoken dialogue system for investigating error handling techniques," in *Proceedings of the International Conference on Spoken Language Processing, ICSLP*, vol. 4, 2004, pp. 229–231.

[9] A. Hartholt, J. Gratch, and L. Weiss, "At the virtual frontier: Introducing Gunslinger, a multi-character, mixed-reality, story-driven experience," in *Intelligent Virtual Agents*. Springer, 2009, pp. 500–501.

[10] W. Swartout, D. Traum, R. Artstein, D. Noren, P. Debevec, K. Bronnenkant, J. Williams, A. Leuski, S. Narayanan, D. Piepol *et al.*, "Ada and Grace: Toward realistic and engaging virtual museum guides," in *Intelligent Virtual Agents*. Springer, 2010, pp. 286–300.

[11] D. DeVault, K. Sagae, and D. Traum, "Incremental interpretation and prediction of utterance meaning for interactive dialogue," *Dialogue & Discourse*, vol. 2, no. 1, pp. 143–170, 2011.

[12] G. Skantze and D. Schlangen, "Incremental dialogue processing in a micro-domain," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009, pp. 745–753.

[13] Z. Wang, J. Lee, and S. Marsella, "Towards more comprehensive listening behavior: beyond the bobble head," in *Intelligent Virtual Agents*. Springer, 2011, pp. 216–227.

[14] J. Gratch, A. Okhmatovskaia, F. Lamothe, S. Marsella, M. Morales, R. van der Werf, and L. Morency, "Virtual rapport," in *Intelligent Virtual Agents*. Springer, 2006, pp. 14–27.

[15] L. Huang, L. Morency, and J. Gratch, "Virtual rapport 2.0," in *Intelligent Virtual Agents*. Springer, 2011, pp. 68–79.

[16] D. Heylen, H. op den Akker, M. ter Maat, P. Petta, S. Rank, D. Reidsma, and J. Zwiers, "On the nature of engineering social artificial companions," *Applied Artificial Intelligence*, vol. 25, no. 6, pp. 549–574, 2011, iSSN=0883-9514.

[17] D. DeVault, K. Sagae, and D. Traum, "Can I finish?: learning when to respond to incremental interpretation results in interactive dialogue," in *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009, pp. 11–20.

[18] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[19] T. J. Visser, D. R. Traum, D. DeVault, and R. op den Akker, "Toward a model for incremental grounding in spoken dialogue systems," in *12th IVA Workshop on Real-time Conversations with Virtual Agents*, Santa Cruz, 2012.

[20] D. R. Traum, "A computational theory of grounding in natural language conversation," Ph.D. dissertation, University of Rochester, Rochester, NY, 1994.

[21] H. Grice, "Logic and conversation," *Syntax and Semantics*, vol. 3, pp. 41–58, 1975.

[22] H. Clark, *Using language*. Cambridge University Press Cambridge, 1996, vol. 4.

[23] D. Traum and J. Allen, "A speech acts approach to grounding in conversation," in *2nd International Conference on Spoken Language Processing*, 1992, pp. 137–40.

[24] D. Traum, "Computational Models of Grounding in Collaborative Systems," in *Psychological Models of Communication in Collaborative Systems-Papers from the AAAI Fall Symposium*, 1999, pp. 124–131.

[25] C. Nakatani and D. Traum, "Coding discourse structure in dialogue (version 1.0)," University of Maryland, Tech. Rep. UMIACS-TR-99-03, 1999.

[26] F. James, K. Lenhart, G. Ferguson, P. Heeman, C. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio *et al.*, "The TRAINS project: A case study in building a conversational planning agent," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 7, no. 1, pp. 7–48, 1995.

[27] D. Traum, D. DeVault, J. Lee, Z. Wang, and S. Marsella, "Incremental dialogue understanding and feedback for multiparty, multimodal conversation," in *Intelligent Virtual Agents*. Springer, 2012.

[28] G. Skantze, "Error handling in spoken dialogue systems," Ph.D. dissertation, KTH Computer Science and Communication, 2007.

[29] G. Skantze and J. Edlund, "Robust interpretation in the Higgins spoken dialogue system," in *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, 2004.

[30] A. Leuski and D. Traum, "A statistical approach for text processing in virtual humans," in *26th Army Science Conference*, 2008.

[31] D. DeVault, D. Traum, and R. Artstein, "Practical grammar-based NLG from examples," in *Proceedings of the Fifth International Natural Language Generation Conference*. Association for Computational Linguistics, 2008, pp. 77–85.

[32] M. Pickering, S. Garrod *et al.*, "Toward a mechanistic psychology of dialogue," *Behavioral and Brain Sciences*, vol. 27, no. 2, pp. 169–189, 2004.

[33] D. DeVault, K. Sagae, and D. Traum, "Detecting the status of a predictive incremental speech understanding model for real-time decision-making in a spoken dialogue system," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[34] K. Sagae, G. Christian, D. DeVault, and D. Traum, "Towards natural language understanding of partial speech recognition results in dialogue systems," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, 2009, pp. 53–56.

[35] E. Krahmer, M. Swerts, M. Theune, and M. Weegels, "Problem spotting in human-machine interaction," in *Proc. Eurospeech*, vol. 99, 1999, pp. 1423–1426.

[36] D. Hofs, M. Theune, and R. op den Akker, "Natural interaction with a virtual guide in a virtual environment," *Journal on Multimodal User Interfaces*, vol. 3, no. 1, pp. 141–153, 2010.

[37] S. Oviatt and P. Cohen, "Discourse structure and performance efficiency in interactive and non-interactive spoken modalities," *Computer Speech & Language*, vol. 5, no. 4, pp. 297–326, 1991.

[38] G. Aist, J. Allen, E. Campana, C. Gallo, S. Stoness, M. Swift, and M. Tanenhaus, "Incremental dialogue system faster than and preferred to its nonincremental counterpart," in *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, 2007, pp. 761–766.

[39] W. Levelt, *Speaking: From intention to articulation*. MIT press, 1993.

[40] D. Schlangen and G. Skantze, "A general, abstract model of incremental dialogue processing," in *Proc. of the 12th Conference of the European Chapter of the ACL*, 2009.

[41] K. Sagae, D. DeVault, and D. Traum, "Interpretation of partial utterances in virtual human dialogue systems," in *Proceedings of the NAACL HLT 2010 Demonstration Session*. Association for Computational Linguistics, 2010, pp. 33–36.

[42] D. DeVault and D. Traum, "Incremental speech understanding in a multi-party virtual human dialogue system," *NAACL-HLT 2012*, p. 25, 2012.

[43] A. Kilger and W. Finkler, "Incremental generation for real-time applications," German Research Center for Artificial Intelligence, Tech. Rep. 95-11, 1995.

[44] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proceedings of the SIGDIAL 2010 Conference*. Tokyo, Japan: Association for Computational Linguistics, September 2010, pp. 1–8.

[45] M. Poesio and H. Rieser, "Completions, coordination, and alignment in dialogue," *Dialogue and Discourse*, vol. 1, no. 1, pp. 1–89, 2010.

[46] C. Kiddon and Y. Brun, "That's what she said: double entendre identification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 89–94.

[47] J. Wohlner, "Institute for Creative Technologies," http://ict.usc.edu/, 2012.

[48] A. Hartholt, T. Russ, D. Traum, E. Hovy, and S. Robinson, "A common ground for virtual humans: Using an ontology in a natural language oriented virtual human architecture," *Proceedings of LREC, Marrakech, Morocco, may*, 2008.

[49] P. Kenny, A. Hartholt, J. Gratch, W. Swartout, D. Traum, S. Marsella, and D. Piepol, "Building interactive virtual humans for training environments," in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, vol. 2007, no. -1. NTSA, 2007.

[50] J. Wohlner, "Virtual human toolkit," http://ict.usc.edu/prototypes/vhtoolkit/, 2012.

[51] D. R. Traum, A. Roque, A. Leuski, P. Georgiou, J. Gerten, B. Martinovski, S. Narayanan, S. Robinson, and A. Vaswani, "Hassan: A virtual human for tactical questioning," in *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, September 2007, p. 71–74.

[52] A. Rizzo, K. Sagae, E. Forbell, J. Kim, B. Lange, J. Buckwalter, J. Williams, T. Parsons, P. Kenny, D. Traum *et al.*, "Simcoach: an intelligent virtual human system for providing healthcare information and support," in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, vol. 2011. NTSA, 2011.

[53] D. Traum, W. Swartout, J. Gratch, and S. Marsella, "A virtual human dialogue model for non-team interaction," *Recent Trends in Discourse and Dialogue*, pp. 45–67, 2008.

[54] W. Swartout, J. Gratch, R. Hill Jr, E. Hovy, S. Marsella, J. Rickel, D. Traum *et al.*, "Toward virtual humans," *AI Magazine*, vol. 27, no. 2, pp. 96–109, 2006.

[55] B. Plüss, D. DeVault, and D. Traum, "Toward rapid development of multi-party virtual human negotiation scenarios," *Proceedings of SemDial*, pp. 63–72, 2011.

[56] S. Scherer, S. Marsella, G. Stratou, Y. Xu, F. Morbini, A. Egan, A. Rizzo, and L. Morency, "Perception markup language: Towards a standardized representation of perceived nonverbal behaviors," in *Intelligent Virtual Agents*. Springer, 2012, pp. 455–463.

[57] J. Laird, *The Soar Cognitive Architecture*. MIT Press (MA), 2012.

[58] L.-P. Morency, I. Kok, and J. Gratch, "A probabilistic multimodal approach for predicting listener backchannels," *Autonomous Agents and Multi-Agent Systems*, vol. 20, pp. 70–84, January 2010.

[59] S. Kopp, B. Krenn, S. Marsella, A. Marshall, C. Pelachaud, H. Pirker, K. Thórisson, and H. Vilhjálmsson, "Towards a common framework for multimodal generation: The behavior markup language," in *Intelligent Virtual Agents*. Springer, 2006, pp. 205–217.

[60] D. Heylen, S. Kopp, S. Marsella, C. Pelachaud, and H. Vilhjálmsson, "The next step towards a function markup language," in *Intelligent Virtual Agents*. Springer, 2008, pp. 270–280.

[61] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. Marshall, C. Pelachaud *et al.*, "The behavior markup language: Recent developments and challenges," in *Intelligent virtual agents*. Springer, 2007, pp. 99–111.

[62] M. Thiebaux, S. Marsella, A. Marshall, and M. Kallmann, "Smartbody: Behavior realization for embodied conversational agents," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 151–158.

[63] J. Lee and S. Marsella, "Nonverbal behavior generator for embodied conversational agents," in *Intelligent virtual agents*. Springer, 2006, pp. 243–255.

[64] P. Ekman and E. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1998.

[65] J. Goldberg, "Interrupting the discourse on interruptions:: An analysis in terms of relationally neutral, power-and rapport-oriented acts," *Journal of Pragmatics*, vol. 14, no. 6, pp. 883–903, 1990.

[66] K. Thórisson, "Natural turn-taking needs no manual: Computational theory and model, from perception to action," *Multimodality in language and speech systems*, vol. 19, 2002.

[67] R. Akker and M. Bruijnes, "Computational models of social and emotional turn-taking for embodied conversational agents: a review," University of Twente, Centre for Telematics and Information Technology (CTIT), Tech. Rep. TR-CTIT-12-13, 2012.

[68] A. Roque and D. Traum, "Degrees of grounding based on evidence of understanding," in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, 2008, pp. 54–63.

[69] H. op den Akker and C. Schulz, "Exploring features and classifiers for dialogue act segmentation," *Machine Learning for Multimodal Interaction*, pp. 196–207, 2008.