



University of Twente

Faculty of Electrical Engineering, Mathematics and Computer Science

P.O. Box 217, 7500 AE Enschede
The Netherlands

Politeness and Alignment in the Virtual Guide

Master's Thesis

Human Media Interaction

M.A. de Jong

08-13-2007

Exam Committee

Dr. M. Theune

Dr. Ir. H.J.A. op den Akker

Ir. D.H.W. Hofs

Get around in ENGLISH - How to be Polite

Wrong



Right



Preface

In the period of October 2006 to August 2007 I have been working on my final project at the Universiteit of Twente, at the Human Media Interaction department.

The subject of talking to an computer program that can understand you and reacts on to you, always interested me. That's why I chose to work on the Virtual Guide. Although a complete understanding and reasoning agent is still far beyond our reach, working on the project has been challenging, but very interesting and fun as well. So much even, that I regret I could not work on it for a bit longer.

Now that my final project is complete, I would like to thank the following persons:

First of all, my parents for making this all possible and supporting me.

Secondly, Mariët, Rieks en Dennis for their excellent guidance. Their constructive comments, knowledge, tips and technical skills helped me a great deal during the course of this project. The relaxed atmosphere at Human Media Interaction made working there enjoyable.

Thirdly, I would like to thank all my fellow students working in the room with me, some of which have already have gotten their diploma. They know who they are, I really enjoyed working together with them and wish them a lot of success in their future endeavors.

Finally, I would like to thank my flat mates, friends and other persons who participated in the user tests and questionnaire, which really helped me out.

Markus de Jong

Samenvatting

The Virtuele Gids is een conversationele agent die in het begin niet veel begreep van wat de gebruiker zei en zelf ook niet veel had te zeggen. Dit afstudeer project was er op gericht om hier verandering in te brengen door het vergroten van de variatie van de taal van de gids door het aan laten passen van het taalgebruik van de Gids aan dat van de gebruiker, zoals ook een echt persoon zou doen.

Zodoende was het primaire doel van dit project het geloofwaardiger maken van de Gids. Voordat dit gerealiseerd kon worden, moesten eerst de basiskenmerken van de Gids worden bestudeerd. Hierdoor is een uitgebreide documentatie van de Virtuele Gids ontstaan. Hierna moesten deze verbeterd worden.

Dit is bewerkstelligd door het uitbreiden van het aantal uitingen van de gebruiker dat de Gids kan begrijpen. Ondersteuning voor bijzinnen, meerdere zindelen met betekenis per uiting en nieuwe manieren voor de Gids om zich te uiten zijn toegevoegd. Andere verbeteringen zijn onder meer een betere dekking en duidelijkere antwoorden op vragen die buiten het domein van de Gids vallen. Ook is er ondersteuning toegevoegd voor korte expressies die uit meerdere woorden bestaan.

Om de geloofwaardigheid van de Gids te verbeteren is er een taal-aanpassingsmodel in de Gids geïntroduceerd, waardoor de taal van de Gids op de lijn van dat van de gebruiker komt te liggen. Uitingen van gebruikers worden gescand voor formaliteit van taal- en beleefdheidsindicatoren, waarmee de aanpassingsstatus van de Gids wordt aangepast. De Gids geeft antwoord door het genereren van natuurlijke taal met variabele formaliteit en beleefdheidstactieken, gebaseerd op deze status.

Beperkte gebruikerstest gaven aan dat de verbeteringen aan de Gids werkten. Het aanpassingsmodel vergrootte de variatie van uitingen van de Gids. Een globale vragenlijst betreffende het aanpassingsmodel gaf aan dat de beleefdheidstactieken die door de Gids werden gebruikt

zo beleefd als bedoeld werden beoordeeld, en dat het aanpassen van de taal merkbaar was. De resultaten voor variaties in formaliteit van de uitingen zijn hoopvol, maar dit onderdeel heeft echter nog meer werk nodig. Op basis van de onderzoeksresultaten zijn veranderingen gemaakt in de implementatie van de Gids. Het wordt aangeraden om meer data te verzamelen voor de formaliteit van taal en dat er meer uitgebreide (gebruikers)tests plaats zullen vinden om het model beter te valideren.

Abstract

The Virtual Guide is a conversational agent that did not understand much of what the user said and did not have much to say at first. This project focused on changing this by increasing the variation of her language and by letting her align her language to that of the user, much as a real person would.

As such, the main goal of this project was to increase the believability of the Virtual Guide. Before this was possible, the basic features of the Guide had to be studied. This yielded usable documentation for the Virtual Guide for future work on the Virtual Guide. Subsequently, these features were improved upon.

This was done by increasing the number of utterances the Guide can understand. Support for dependant clauses as well as multiple dialogue acts per user turn. New ways for the Guide to express itself were added. Other improvements include better coverage and more informing answers to questions that are out of the Guide's domain and support for short expressions of multiple words.

In order to increase the believability of the Guide, a language alignment model has been introduced. User utterances are scanned for formality and politeness indicators, after which the alignment state is adjusted. Based on the alignment state, system replies are created using natural language generation with varying politeness tactics and formality of language.

Limited usability tests indicated the proper functioning of the improvements made on the Guide's implementation. The alignment model increased the variation of utterances of the Guide. A questionnaire covering the alignment model indicated that the politeness tactics were received as was intended and that the alignment of language was noticable. The results on variation of formality of language of utterances was hopeful, but this part of the implementation still needs improvements. Based on these results, some changes in the implementation were made.

It is recommended that more data on language formality is collected and that more thorough (usability) tests are conducted to further validate the model.

CONTENTS

1	Introduction	1
2	The Virtual Guide	3
2.1	Introduction	3
2.1.1	The VMC world	3
2.1.2	The Virtual Guide	4
2.2	System Architecture	5
2.2.1	Multimodality	5
2.2.2	Facilitator	5
2.2.3	Exemplars	5
2.3	Process Description	7
2.3.1	Overview	7
2.3.2	Reference Resolver	7
2.3.3	World	9
2.3.4	History	9
2.3.5	Turn Taking	9
2.4	Text Parsing	11
2.4.1	Grammar and Lexicon	11
2.4.2	Feature Structures	12
2.4.3	SParses	12
2.5	Dialogue Act Interpretation	14
2.5.1	Dialogue Act Types	14
2.5.2	User Dialogue Act Selection	14
2.6	System Actions	17
2.6.1	Action Templates	17
2.6.2	Action Arguments	17
2.7	Example Dialogues	19
2.8	Issues	21
2.8.1	Language Recognition	21

2.8.2	Ontology	21
2.8.3	Out of domain	21
2.8.4	Miscellaneous Issues	22
3	Robustness	23
3.1	Introduction	23
3.2	Corpus Analysis	25
3.2.1	The Bussink Corpus	25
3.2.2	Results	26
3.3	Improving recognition	28
3.3.1	Lexicon	28
3.3.2	Action Templates	28
3.4	Dependent Clauses	30
3.4.1	Grammar	30
3.4.2	SParses	33
3.5	Multiple Dialogue Acts	34
3.5.1	Grammar	34
3.5.2	SParses	38
3.5.3	Turns	40
3.6	New User Dialogue Acts	43
3.6.1	Thanking	43
3.6.2	Positive and Negative Feedback	43
3.7	Miscellaneous Changes	45
3.7.1	Replies to Out of domain Questions	45
3.7.2	Multi-word Expressions	46
3.8	Usability Tests	48
3.8.1	Goals	48
3.8.2	Setup	49
3.8.3	Results	50
3.8.4	Conclusions	51
4	Alignment	53
4.1	Introduction	53
4.2	Alignment of Language	55
4.3	Politeness	56
4.3.1	Face	56
4.3.2	On-record strategy	56
4.3.3	Off-record strategy	60
4.3.4	Don't do FTA strategy	60
4.3.5	Politeness in Dutch	60
4.4	Formality of Language	65
4.4.1	Surface Formality and Deep Formality	65
4.4.2	Surface Formality Model	65
4.4.3	Fuzziness	66

4.5	Related work	67
4.5.1	Variation through Politeness	67
4.5.2	Alignment in Natural Language	69
4.6	Alignment Model	70
4.6.1	The Virtual Guide's Behaviour	70
4.6.2	The Alignment State	70
4.6.3	System Description	72
4.7	User Input Analysis	74
4.7.1	Formality Analysis	75
4.7.2	Politeness Analysis	77
4.8	Language Generation	81
4.8.1	Sentence Planning	81
4.8.2	Exemplars	81
4.8.3	Politeness Tactic Selection	83
4.8.4	Variation in Formality	86
4.8.5	Sentence Revisions	87
4.9	Virtual Guide User Interface	88
4.10	Example	89
4.11	Discussion	91
5	Evaluation	93
5.1	Introduction	93
5.2	Questionnaire	93
5.2.1	Goals	94
5.2.2	Setup	94
5.2.3	Formality Values	94
5.2.4	Politeness Tactics	96
5.2.5	Alignment Values	97
5.3	Results and discussion	103
5.3.1	Formality Values	103
5.3.2	Politeness Tactics	105
5.3.3	Alignment Values	109
5.4	Conclusions	113
6	Conclusions	115
6.1	Conclusions	115
6.2	Recommendations and Future Work	116
A	List of MWE's and their lexical representation	119
B	Action Templates	121
C	Supported Sentences	131
D	Questionnaire	133

*CONTENTS**CONTENTS*

E Questionnaire Results	141
Bibliography	145

CHAPTER 1

Introduction

The Virtual Guide is part of a virtual representation of the Muziekcentrum (Virtual Music Centre, or **VMC**) in Enschede. A picture of the application in action can be seen in figure 1.1.

The Virtual Guide is an *agent* that represents a guide. In short, an agent is a piece of software that assists the user. In our case it assist the user with navigation tasks. It is a *conversational agent*: the user is able to go into dialogue with it. It's also an *embodied agent*: the agent is represented in 3D by an avatar, as can be seen in figure 1.1.

This thesis focuses on the analysis and generation of language in the Virtual Guide, and finally resulting in the addition of alignment of language in order to enhance the user experience with the Virtual Guide.

The thesis starts with a *general description* of the Virtual Guide system, which includes a process description from start to end, a description of its parts and how they relate and communicate with each other to produce a working dialogue system, and finally a discussion of what the limitations of the original implementation are.

After introducing the system, a chapter is dedicated to the *robustness improvements* that were made in the implementation in user input recognition and general issues, concluded by usability tests.

In the next chapter, the expansion of the *language alignment model* based on formality of language and politeness is presented, starting with a discussion of relevant literature and related work, followed by a description of the alignment model and realisation of this model.

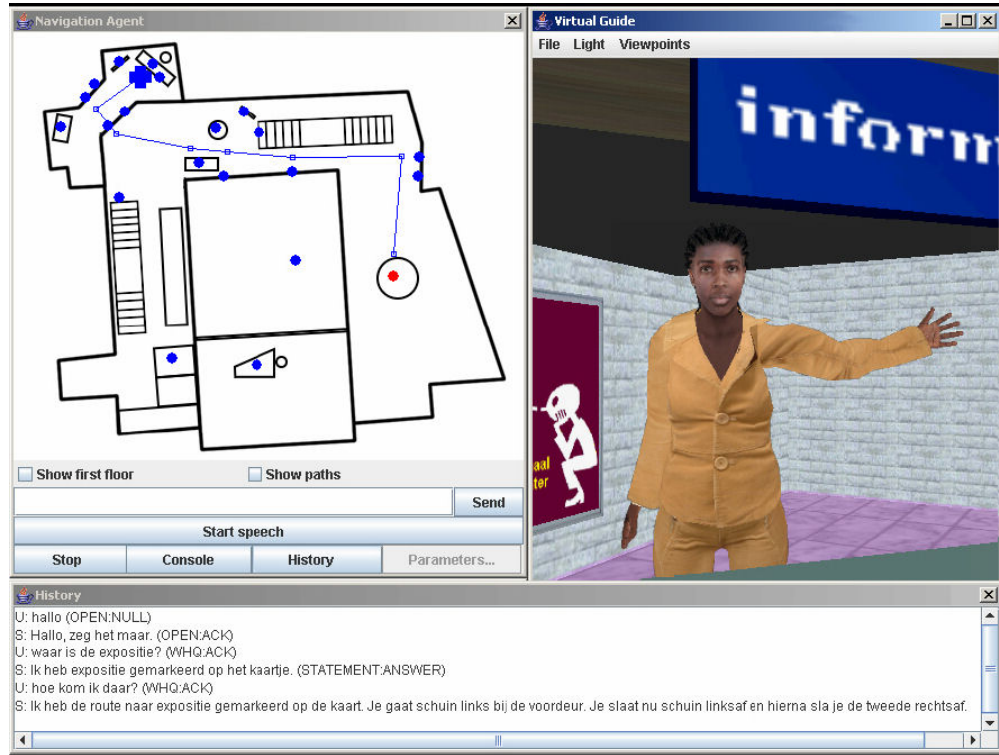


Figure 1.1: The Virtual Guide in the VMC world

After this, the evaluation of the alignment model is discussed, followed by conclusions and suggestions for future work.

CHAPTER 2

The Virtual Guide

This chapter will give an overview of the original implementation of the Virtual Guide before any adjustments or additions were made. It can be used as a manual for future work on the Virtual Guide, since such a collection of documentations was not available before. First, the overall architecture will be discussed, followed by a description of the different parts and processes in the Virtual Guide system. Special attention will be given to the processes that make the Guide able to carry out a dialogue with a user. Finally, some example dialogues with the Guide and issues with the implementation of the system will be examined.

2.1 Introduction

This section will introduce the user to the Virtual Guide and the VMC.

2.1.1 The VMC world

The VMC represents a real life building located in the Dutch city of Enschede. Its creation originated in the HMI group's Parlevink project of the Computer Science Faculty at the University of Twente and is described in [NH00]. The VMC is a 2-storey building a user can navigate through. It has many locations and objects, such as an exhibit, coffee tables, a bathroom and an auditorium.

A map with all the objects and locations marked as dots is displayed in figure 2.1.

The Graphical User Interface of the Virtual Guide consists of a 3D-view of the world next to a console, see figure 1.1. The user enters his or her input into the console by text or talks

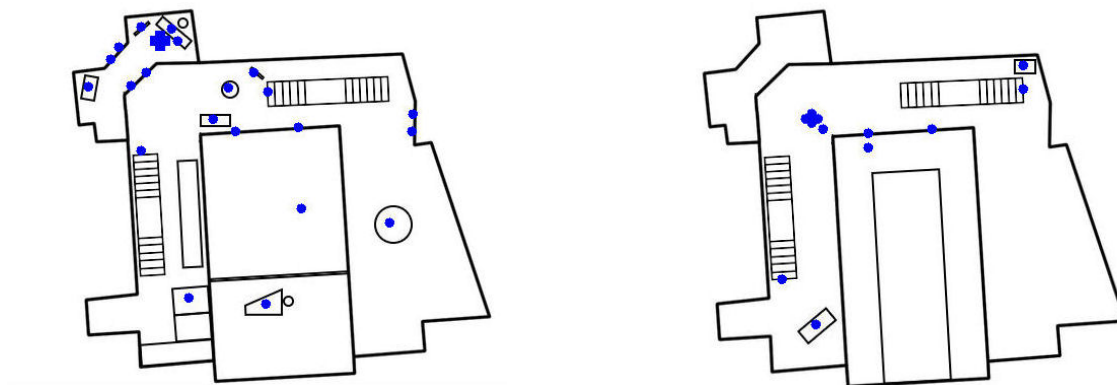


Figure 2.1: Mini-maps of both floors

to the Guide using speech recognition.

2.1.2 The Virtual Guide

The Guide is limited to answering questions related to the locations and routes to objects in the world. The Guide's avatar itself is stationary and located behind a desk. The user can ask queries while being anywhere in the VMC world, either via speech or textual input. The Guide cannot give information on for instance concerts and performances in the VMC or supply the user with step by step directions in real time as the user navigates through the building. Instead, the Guide is able to mark locations of objects on the map and can give a description of the route from the current user location to an object verbally, while at the same time displaying the route on the map. The avatar uses animated gestures that are associated with words in the route description, such as 'left' and 'right', after which it will point in that direction. Examples of possible user queries and subsequent dialogues can be found in section 2.7.

2.2 System Architecture

This section describes three general technical features of the Virtual Guide.

2.2.1 Multimodality

As said, the user can ask questions related to navigation tasks. This is made possible by using *multimodal input*: by speech or text and by pointing at the mini-map (*gesture input*). This multimodal input is processed by the Virtual Guide, or Guide for short. The Guide's output is also multimodal. It can reply in text and speech (the generated textual output is speech-synthesized), by marking locations on the mini-map and by making gestures by pointing either left or right. Another feature is the calculation and displaying of routing information on the map. See [BK04] for a description. It must be noted that, for simplicity, only textual and gesture input is used in this project. This simplifies user input analysis and is less prone to error and ambiguity.

2.2.2 Facilitator

The Virtual Guide architecture uses a *facilitator* agent. In short, other agents can register with the facilitator agent to request messages of certain types, comparable to a message subscription. Whenever the facilitator receives a message from an other agent, it will send this message to all agents that have a subscription to this message type. The agent is unaware who receives the output the agent creates and how this output is processed. A facilitator makes the architecture more flexible because inter-agent communication is done by a single agent, removing the need for agents to refer to other agents individually. A facilitator also increase modularity: it is possible to replace one agent with another as long as the communication protocol is the same. Also, the agents can be moved to different locations without problems. The agent only needs to know the location of the facilitator.

2.2.3 Exemplars

The system architecture uses CoGenTex's Exemplars Framework, version 1.2 [exe07] as described in [WC98] to generate natural language.

Exemplars is a rule-based, object-oriented framework for dynamic generation of text, based on java objects. This method uses templates based on XML to generate language to allow logical functions to skip or include and randomize parts in the generated text. After compiling the exemplars, they are converted into java classes which are used by the Virtual Guide. An example exemplar is displayed below:

```
// Handle the routeData.
exemplar numberEdits(int i) extends Edits implements RandomlySelectable {
    boolean evalConstraints() { return i < 5; }
    boolean justSkip() { return i = 0; }
    double weight() { return 0.5; }

    void add() {
        i+=2;
    }
    java.util.Random r = new java.util.Random();
    void apply() {
        add();
        int j = i + r.nextInt();
        <<+
            Het nummer is <number type='NumberEdit'>{ j }</number>
            {{ StartCalculation(j) }}
        +>>
    }
}
```

This example shows the exemplar `numberEdits`. If the constraints are met (checked with `evalConstraints()`), the `apply()` method executes the code of the exemplar. Exemplars may use java objects such as `{j}`, but other exemplars as well, such as `StartCalculation(j)`. Exemplar code is displayed as `<<+...+>>`. This is what will generate the actual system utterance. As with java classes, exemplars can extend each other. In this example, `numberEdits()` extends `Edits`. Also, a random factor can be introduced by letting an exemplar class implement `RandomlySelectable`. `weight()` can be added to exemplars to influence the random selection process and shift the weight to a certain exemplar in a group of possible choices.

Another function of exemplars is the application of *edit rules* for text polishing. By tagging certain words with `<tag>word<\tag>`, the edit rules will look for this tag and make changes if necessary. For example, a numeral “1” might be replaced by the word “one” with the help of edit rules. After the edit rules have been executed, the tag is removed from the utterance.

After compiling the exemplar classes, java classes are formed which can be called by the system.

See [BK04] for a description of its usage for the generation of route descriptions in the Virtual Guide. Exemplars are reused in an expansion of the Virtual Guide for the alignment model (see section 4.8.2). The process of generating language is also described there.

2.3 Process Description

This section will describe the overall inner workings of the Virtual Guide. Elements important for this project will be discussed in greater detail in later sections. This process description will start with the user entering input, and will end with the system executing a system action, after which the process will start again for new user input.

2.3.1 Overview

See figure 2.2. Communication between agents in this figure happens via the facilitator (see section 2.2.2). The process starts with the user opening the dialogue by either speech or text. In case the user uses speech, it is processed by the **Speech Recognizer**. The speech recognition result is then passed on to the **NLParser**.

If the user enters text, this is done via the console or **TextInputPanel**. By entering text and pressing the send button, the text input is passed on to the **NLParser**. In this project, speech recognition is not used, so from now on only text input will be discussed. A third form of supported input are mouse clicks from the user on the mini-map which are registered by **World2dMap**. This will result in gesture input. The **NLParser** will parse the text input by using a lexicon and grammar. The parsed text input is sent to the **fusion agent**, which can merge different modalities of user input. It also connects noun phrases in the input to objects in the VMC. After merging the gesture input and parsed text input, its output is multimodal. More information on the processing of textual input is given in section 2.4.

Multimodal input is analysed by the **Dialogue Act Determiner**, which determines the *user dialogue act*. This will result in a user turn that will execute system actions for system replies. See section 2.5 for a more detailed description of the user dialogue act recognition process and section 2.6 for an overview of system actions.

The purpose of the **Dialogue Manager** is to sequentially process user input, to update the state of the dialogue and world and to generate system actions. It creates one or more dialogue acts based on multimodal input and the dialogue history.

The dialogue manager will take a resolved user turn and process it to generate a correct system turn. Besides the dialogue act determiner, it uses **Reference Resolver**, **World**, **Action Stack** and **Dialogue History** to generate system output. These features will be discussed next.

2.3.2 Reference Resolver

The noun phrases in the input are connected to world objects by the fusion agent, as well as connecting gesture input to objects in *deictic expressions*. Deictic expressions are references that are dependent and only have meaning in situation they are expressed, in this case the recency of pointing input (pointing on the map + “what is this?”).

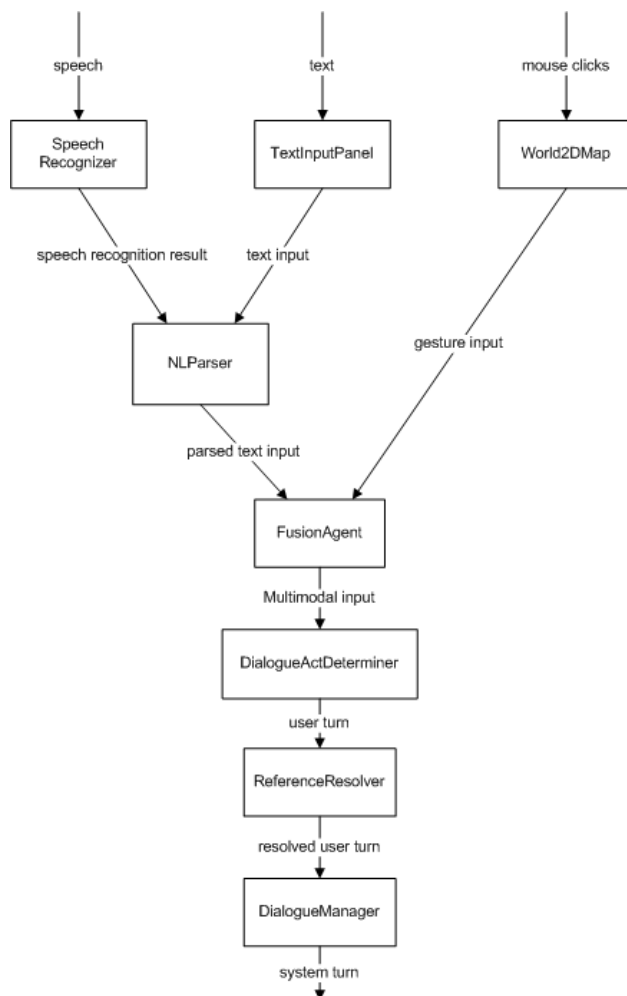


Figure 2.2: An overview of the application architecture

The **Reference Resolver** does the same for *anaphoric expressions*. An anaphoric expression is an instance of one expression referring to another. After this, the input analysis is complete.

The reference resolver implements a modified version of the Lappin and Leass’s algorithm [LL94] to determine the *referents* of referring expressions in a user utterance. This will be explained shortly. The modification makes the algorithm suitable for multimodal dialogues.

The following dialogue shows an example:

U: Where is the bathroom?
 S: I marked the bathroom on the map.
 U: How do I get there?

In this example “the bathroom” and “there” refer to the same object. This object is called

the referent. “the bathroom” and “there” are the references to this object.

As described in [HodAN03], weights are assigned to references, based on a set of *salience factors*. These factors include recency, the type of input (gesture or verbal) and whether the reference in a phrase is a head noun or the subject of a sentence. A salience value is the sum of values associated with a set of salience factors. The salience value of a referent is the sum of the salience values of all utterances.

Two kinds of references can be distinguished: those that consist of demonstrative pronouns such as “this” or “there”, and content words such as nouns and adjectives. If a demonstrative pronoun is detected, the reference resolver will only look at earlier referents. The most recent referent is always preferred. After calculating the salience values, the referent with the highest salience value is selected and passed on in a resolved user turn to the facilitator.

2.3.3 World

The state of the world is maintained by the `World` class. The world is initialised with an XML file that contains the locations of different objects in the world. It also uses a *dictionary text file* that specifies how the user can refer to each object in the user utterances. The fusion agent uses the `World` class to match certain user input with objects in the world. The world also uses an ontology for its objects, although at this state it was not accessible for hierarchical information (see section 2.8.2). A study of the possibility of recovery of the ontology will be discussed section 2.8.

2.3.4 History

The `History` class maintains the state of the dialogue, except for the reference resolution. It lists all the dialogue acts that occurred in the dialogue and orders them in subdialogues. Each subdialogue is a list of dialogue acts and may in turn have subdialogues. The main dialogue is considered a subdialogue as well. See section 2.7 for an example of a subdialogue.

The history is aware of the running subdialogues at all times. This is accomplished as follows: the history is initially an empty stack. After a dialogue starts with a dialogue act, this dialogue act is placed on the stack. This is considered as the main dialogue. If a new subdialogue starts, it is pushed on top of the stack, and when it ends it is taken off the stack. This way the top of the stack always represents the current subdialogue.

2.3.5 Turn Taking

The `dialogue act determiner` will request multimodal input from the facilitator. The facilitator also connects the reference resolver to the dialogue manager. The dialogue manager is subscribed to *resolved user turn* messages from the facilitator.

The `Dialogue Manager` uses turns in a dialogue with the user. The turn is held by either

the *system* or the *user*. Technically speaking, not in the traditional sense, the user holds the turn almost all the time in this architecture. As the user is holding the turn, the dialogue manager waits for input from the user. The input is processed in a relatively short time. The successful processing of dialogue input leads to system actions being put on the **action stack**. The action stack holds the actions the system still needs to execute. See section 2.6 for a more detailed description of what actions the system can execute. After processing the input, the turn is passed on to the system. When the system holds the turn, it will execute the actions that were put on the action stack. After output is created, the **Dialogue Manager** sends a system turn message to the facilitator.

The following sections will discuss these features in greater detail: text parsing, user dialogue act interpretation and system actions.

2.4 Text Parsing

User text input is parsed with the grammar that uses the lexicon to build feature structures representing the user utterance. This section will discuss the grammar and lexicon, the construction of feature structures and the converting process of feature structures into an XML structure: the *S_Parse*.

2.4.1 Grammar and Lexicon

The lexicon and grammar used by the Virtual Guide were made by Rieks op den Akker, and are used for recognizing user input. The lexicon consists of 1400 words. When, for example, the user enters “*hoi*” (hi), this word is looked up in the lexicon. Below is an example of how a lexical item is formatted in the lexicon:

```
\w hoi
\c IJECT
\g GREET_START
\f
```

This example displays a lexicon entry of the Dutch word “*hoi*” (hi), an interjection of type GREET_START. This tag means this interjection is an introductory greeting. \w is the lexical entry, \c is the part of speech type of the word and \g is the semantic type.

\f is for miscellaneous properties, and can denote if the word is plural or singular or its gender (in case of a noun) or verb type information (in case of a verb) etcetera. The lexicon can be expanded easily by adding entries and their properties.

The grammar used by the Virtual Guide is a *unification grammar*. The first grammar rule is shown below:

```
Rule{s}
Z -> S
    <Z head> = <S head>
    <Z stype> = <S stype>
    <Z sem> = <S sem>
```

The *s*-rule prescribes that each utterance (Z) is transformed into a sentence (S) with a head, a *sentence type* (*stype*) and semantic content (*sem*). The *iject*-rule, for interjections such as greetings, is displayed here:

```

Rule{iject}
S -> IJECT
  <S stype> = inter
  <S sem> = <IJECT sem>

```

This rule ties the elements of the *s*-rule to the *iject*-rule. The *stype* becomes *inter* and the semantic content of the S will refer to the lexical entry of the interjection. The parse structure is shown in figure 2.3.



Figure 2.3: (simplified) parse tree of the interjection ‘hoi’

2.4.2 Feature Structures

Using the grammar and lexicon, the following feature structure is formed after the user enters “hoi”:

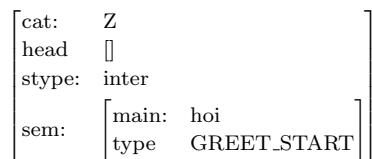


Figure 2.4: The feature structure for the sentence “hoi”

The category of the structure is *Z*, the *head* is empty because the *iject*-rule has no *head*, the sentence type is *inter* (interjection) as prescribed by the *iject*-rule. The semantic contents of the structure is “hoi” and the type of this interjection is *GREET_START*.

2.4.3 SParses

The feature structures that are created by parsing text input are converted into **SParses**. SParses are XML-structures used to determine the *user dialogue act* as well as for matching in *action templates* to select system actions (see section 2.5). The reason for using SParses is the ease of handling the slim XML-structures involved in both analysing and making templates for matching, in contrast to the rather bulky feature structures.

Each feature structure is scanned for its sentence type. Each sentence type is handled differently in the parse analyser, and has its own functions to retrieve specific contents from the feature structure which are needed to form an SParse. The construction of the SParses is handled by the parse converter. The resulting SParse for “hoi” is displayed in figure 2.5.

```
<s_parse id="2" left="0" right="1" surface="IJECT">
  <iject_iject>
    <iject_parse id="1" left="0" right="1" type="GREET_START"/>
  </iject_iject>
</s_parse>
```

Figure 2.5: An SParse for ‘hoi’

The `<s_parse>` is the main element of the SParse, with *surface form* IJECT (see table 2.1. This SParse has a main `<iject_iject>` element which contains a `<iject_parse>` element that represents the interjection “hoi” with *type* GREET_START. The *id* is the unique ID in the scope of a top-level sentence parse, *left* is the index of the word where the parse starts and *right* is the index of the word where the parse ends (that word is not included in the parse).

Notice that the semantic content is not actually in the SParse, because this is not needed in action template matching or user dialogue act interpretation. The information is not lost however, because the raw, unchanged string of the user utterance will remain an available part of the dialogue act.

Surface form	Description
ADJP	Adjective phrase.
ADVP	Adverbial phrase.
CONT	Continuer (conjunctive plus sentence).
DECL	Declarative sentence.
IJECT	Interjection.
IMP	Imperative.
NP	Noun phrase.
OTHER	Other surface form.
WHQ	Wh question.
YNQ	Yes/no question.

Table 2.1: Surface forms of SParses

2.5 User Dialogue Act Interpretation

In order for the Guide to ‘understand’ what the user wants, the user’s utterance must be connected to a user *dialogue act*. A dialogue act (or DA) represents the speaker’s intention. The **dialogue act determiner** selects one dialogue act from a list of possible user dialogue acts which is generated by this agent. This results in one instance of *UserDialogueAct*, which consists of a *forward tag*, a *backward tag* (see next section), the raw text input and a multimodal parse of the text input and possible gesture input.

2.5.1 Dialogue Act Types

The Virtual Guide uses the **DAMSL** (Dialogue Act Markup in Several Layers) scheme [AC97] for annotating dialogs. It incorporates Forward Looking and Backward Looking functions.

The Forward Looking Function determines how the current utterance constrains the future actions of the participants and affects the discourse. *The Backward Looking Function* determines how the current utterance relates to the previous discourse [AC97].

A dialogue act is determined by a **forward tag** and a **backward tag**. Forward tags are listed in table 2.2, and backward tags in table 2.3.

Example:

```
U: hallo (OPEN:NULL)
S: Hallo, zeg het maar. (OPEN:ACK)
```

The dialogue act of the user’s utterance (*U*) has forward tag OPEN (because the utterance is an opening) and backward tag NULL (because this is the first user dialogue act and there is no previous dialogue act). The system reply is an (OPEN:ACK). The forward tag is OPEN which means it’s a greeting. The backward tag is ACK: the Guide has understood the last utterance of the user and continues the dialogue.

2.5.2 User Dialogue Act Selection

The dialogue act determiner will analyse the alternative SParses in the multimodal input it receives from the fusion agent and selects one or more of them to create the most likely list or lists of dialogue acts. If a list contains more than one dialogue act, it means that the user performed more than one dialogue act in one turn. For example, the user utterance: “hi, where is the bathroom?” is represented by two dialogue acts: a greeting (*OPEN*) and an open question (*WHQ*). This feature was not yet available in the original implementation, however, but was implemented later. If the multimodal input is empty, which happens when the user’s utterance cannot be parsed, an uninterpretable dialogue act will be created.

Forward tag	Description
ASSERT	Speaker asserts something, usually the answer to a question.
CHECK	Speaker checks information that the hearer knows about.
CLOSE	Speaker closes the dialogue.
COMMIT	Speaker promises something.
NULL	This constant indicates that no dialogue act has occurred yet in the dialogue.
OFFER	Speaker offers something.
OPEN	Speaker opens the dialogue.
REQ	Speaker requests something.
THANK	Speaker says thank you.
WHQ	Speaker asks an open question.
WHQ_ELLIPSIS	Speaker intends to repeat the last open question but replaces one parameter.
YNQ	Speaker asks a yes/no question.

Table 2.2: Dialogue Act Forward Tags

Backward tag	Description
ACCEPT	Speaker accepts a request or an offer.
ACK	Speaker understood the last utterance and continues the dialogue.
HOLD	Speaker starts a subdialogue instead of replying to the last utterance.
MAYBE	Speaker reacts to a request or an offer by neither accepting nor rejecting it.
NULL	Speaker starts a dialogue.
REJECT	Speaker rejects a request or an offer.
REPEAT	Speaker repeats (part of) the last utterance.

Table 2.3: Dialogue Act Backward Tags

The process of selecting the most likely dialogue act for a user utterance will be described next. After parsing the user utterance and ending up with a collection of possible alternative SParses, the surface form (see table 2.1) of each parse is read. For each surface form, different methods are implemented to get the possible interpretation or list of interpretations of the SParse.

Next, the most likely interpretation is selected. This is done by finding the last system dialogue act in the current subdialogue, together with the last dialogue act in the underlying subdialogue (the dialogue level from which the current subdialogue was initiated), if that dialogue act exists and if it is a system dialogue act.

From the list of possible interpretations, the interpretation that matches best with a previous system dialogue act will be selected. This selection process will be discussed now.

If a subdialogue is running and the last dialogue act in the underlying dialogue was a system dialogue act, the dialogue act determiner always tries to end the subdialogue by finding an interpretation that likely follows the last system dialogue act in the underlying dialogue.

If the agent is unable to find such an interpretation, it will look at the last system dialogue act in the current subdialogue.

If this fails as well, the first available interpretation is selected and if the last dialogue act in the underlying subdialogue was a system dialogue act, it is assumed that the current subdialogue is closed.

If there was no earlier system dialogue act at all, the dialogue act determiner will select interpretations that most likely occur at the start of a dialogue.

After this, the most likely interpretation for each of the alternative parses of the multimodal input will have been determined. Next, the list of interpretations will be filtered to get the single most likely interpretation.

To determine the most likely follow-up dialogue act of a previous system dialogue act, the dialogue act determiner uses a *preferences map*. This dialogue act preferences map is an XML-file that is used to find the most likely follow-up user dialogue act after a certain system dialogue act. The dialogue act determiner takes the forward tag of the system dialogue act and uses the preferences map to determine the most likely follow-up backward tags and forward tags for that forward tag. The preferences map also defines a list of forward tags that most likely occur at the start of a dialogue, when there is no system forward tag to use.

2.6 System Actions

A set of action rules determine what action should be performed, based on the dialogue act's forward tag. The action stack (see 2.3.5) contains the actions the system still needs to perform, but it also has methods to create new system actions. See table 2.4 for a list of possible system actions.

2.6.1 Action Templates

An *action template* is an XML-file that contains user dialogue act templates in the form of SParses and maps them to system actions classes. These system actions realise the system response, by answering user's questions in text and speech and by marking objects and routes on the mini-map. If a user dialogue act matches the template, the system action can be created to be put on the action stack.

Possible system actions are displayed in table 2.4. An example of an action template is displayed in figure 2.6. This template corresponds with the user utterance “*where can I find the bathroom*”. The element `<action_template>` contains the action template. The `<s_parse>` element should match with the SParse formed out of the input, and the input's forward tag should match with that of the template (here *WHQ*). If the input SParse matches the SParse template and the Forward tags match as well, the connected action class is called. In this case, this is the action class *ActionTellLocation*.

2.6.2 Action Arguments

System actions may or may not require an argument, such as the object of which a user wants the location. These arguments are found in the SParse that was constructed based on user's utterance. The correct argument is selected in the action template by numbering the elements in the template and by specifying which element will contain the argument in the `<arglist>` element. In case of the example, this is the noun phrase element, or `<np_parse>`. In the example utterance “*where can I find the bathroom?*”, this element corresponds to “*the bathroom*”, which will be the action argument.

Some actions do not need an argument, such as *ActionGreet* and *ActionClose*. These actions simply result in the guide responding to a user greeting or a closing utterance such as “good bye”, and do not need any more information. Other actions can have optional or multiple arguments. For instance, *ActionTellLocation* can both tell the location of an object and the location of the user by either supplying an object as argument or leaving the argument empty, and *ActionTellPathFromTo* needs two objects as arguments to be able to the path between them.

Whenever a user dialogue act is uninterpretable, when for instance a word could not be found in the lexicon, *ActionUninterpretable* is called which results in an explanation that the utterance could not be understood.

System Actions	Description
ActionClose	Respond to a closing.
ActionDescribeObject	Identify an object the user has clicked on in the mini-map.
ActionFillArgument	Looks at action on top of the stack and makes it executable by filling in the missing argument
ActionGoTo	Take the user to the location.
ActionGreet	Respond to an opening.
ActionNoMatch	Executed when there is no matching action template for a user dialogue act.
ActionTellLocation	Mark the location of the object on the mini-map.
ActionTellPathFromTo	Tell the path from one object to another and mark it on the mini-map.
ActionTellPathTo	Tell the path to the object and mark it on the mini-map.
ActionThank	Respond to a user thanking the Guide. Not functioning.
ActionUninterpretable	Executed when a user dialogue act is uninterpretable, for example because the parser could not recognise one or more words of the user utterance.

Table 2.4: System actions

```

<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="1" >
    <subj>
      <np_parse id="2"/>
    </subj>
    <main_verb>bevinden</main_verb>
    <adv_pps>
      <adv_parse id="3" adv="waar"/>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="2"/>
  </arglist>
</action_template>

```

Figure 2.6: An action template for "where is X", with X representing a world object.

2.7 Example Dialogues

This section will demonstrate some example dialogues possible with the original implementation of the Virtual Guide.

Where question and route navigation question

U: hallo (OPEN:NULL)
S: Hallo, zeg het maar. (OPEN:ACK)
U: waar is de zaal (WHQ:ACK)
S: Ik heb zaal aangegeven op de kaart. (STATEMENT:ANSWER)
U: hoe kom ik daar (WHQ:ACK)
S: Ik heb de route naar zaal gemarkeerd op de kaart. Je komt bij de voordeur. Je gaat schuin links, nu sla je schuin linksaf en dan sla je de tweede rechtsaf. Daar vind je zaal. (STATEMENT:ANSWER)

This example shows the Guide explaining the location of an object. Next, the User refers to the hall again with a demonstrative adverb (“daar” (there)), and asks for the route to that object. Besides explaining the correct route to the user, this route is subsequently displayed on the mini-map.

From-to navigation question

U: hoe kom ik van keyboard naar toilet? (WHQ:ACK)
S: Ik heb de route van keyboard naar toilet gemarkeerd op de kaart. Je gaat schuin rechts. Je gaat rechts. Je slaat hierna schuin linksaf. Dan sla je linksaf, je slaat schuin linksaf en je gaat schuin links. Hier ga je schuin rechts. Dan kom je bij toilet. (STATEMENT:ANSWER)

This example demonstrates the possibility to ask for a route from one object to another.

User location question

U: waar ben ik (WHQ:ACK)
S: U bevindt zich bij het kruis op de kaart. (STATEMENT:ANSWER)

Informing the user where on the map he or she is located.

A subdialogue in the main dialogue

U: waar is de koe
S: Wat zoekt u? (WHQ:HOLD)
U: de zaal (STATEMENT:ANSWER)
S: Ik heb de zaal gemarkeerd op de kaart (STATEMENT:ANSWER)

This example shows a subdialogue, which is indented. The Guide holds the dialogue and waits for input of the user, because it was unable to find an object called “koe”. After the user’s (elliptic) answer, the subdialogue is closed.

Uninterpretable

* U uses a word that is not in the lexicon
U: UNINTERPRETABLE
S: Ik heb niet begrepen wat u bedoelt. (STATEMENT:NACK)

The user uses one or more words that are not found in the lexicon.

No match in action template

U: ik wil graag fietsen (REQUEST:ACK)
S: Ik verstond: "ik wil graag fietsen", maar ik weet niet wat ik hiermee moet doen. (STATEMENT:NACK)

The user’s utterance can be properly parsed, but cannot be mapped to a system action.

2.8 Issues

This section discusses issues in the original implementation of the Virtual Guide.

2.8.1 Language Recognition

Initial try-outs made clear that the Guide's vocabulary was severely limited. As later tests would show (as discussed in the next chapter), this severely limited the user's freedom in expressing him- or herself. When, for example, the user would refer to the toilet object with "bathroom", the Guide could not interpret the user utterance. Other recognition issues will be discussed in chapter 3.

2.8.2 Ontology

Another issue the object *ontology* the Guide used to store its world objects. The ontology was made in a previous project [BK04], but it did not seem to function properly. The ontology's goal was to create a hierarchical structure of objects and their properties. For instance, a chair object was a member of the group furniture, that was in turn member of the group objects. This chair might have properties, such as a color. The ontology should be used to refer to objects in the VMX which have more than one instance, such as doors, chairs and stairs. If the user would ask the location of a chair, the ontology was supposed to help the Guide differentiate between the instances of chairs, and ask which chair of the group of chairs the user meant. The failure of the ontology resulted in the following dialogue deadlock:

U: waar is de stoel (WHQ:ACK)

S: Kunt u iets preciezer zeggen welke u bedoelt.(WHQ:HOLD)

In this dialogue, the user asks for the location of the chair, and the Guide is unable to determine what chair the user means. The vague answer of the Guide also raises questions about the correct handling of questions that are out of domain. Attempts to solve this problem are discussed in section 3.7.1.

2.8.3 Out of domain

The Guide's replies when the Guide is unable to help the user, answers to questions that are *out of domain*, are vague. It is not always clear what the Guide wants of the user. The difference between an uninterpretable dialogue act (when words of the user are not in the dictionary) and a failure to match an action template is also unclear to the user (see section 2.7). More importantly, it is unclear what the user should do next. Should the user rewrite the sentence or should he or she use entirely different words?

2.8.4 Miscellaneous Issues

Try-outs also made clear that not all functions of the Guide were working properly. For instance, the following gesture input example did not function at all because the question “what is this” was not mapped to a system action:

```
* U performs a mouse click on the keyboard object on the mini-map
  U:  wat is dat (YNQ:NULL)
  S:  Dat is keyboard. (STATEMENT:ANSWER)
```

Asking this question did not result in a successful match to a system action, and the Guide would simply respond as in the ‘No match in action template’-example of section 2.7.

CHAPTER 3

Robustness

3.1 Introduction

The limited number user utterances that were recognized by the Virtual Guide (see section 2.8 for this and other issues) caused a great lack of freedom in user input. For instance, most objects in the VMC world could only be referred to by a single word, and sentences were restricted to only a few basic forms.

User tests that were planned to validate the improvements and new features to the Virtual Guide in the future would be severely limited. Before the user tests could be organised or before any new additions to the system could be made, it was decided to improve the current implementation and increase its **robustness**. Robustness in this project will be explained through the following main goal: **increase the recognition rate of user input**

This goal is divided into the following subgoals:

1) Increase the size of the lexicon by adding synonyms of verbs and objects:

Find synonyms for all objects in the world and for verbs that are often used in navigation tasks.

For instance:

toilet: bathroom, restroom

show, as in “can you show me the way”: point out, tell

2) Increase support for sentence structures:

Increase the ways a user can ask a question. Instead of only supporting “where is the hall?”, add support for sentences variations as “where can I find the hall?” or “I’m looking for the hall”. This can be done by extending the grammar and action templates.

3) Add new actions

For instance, allow the user to give comments on the Guide’s actions, divided into positive and negative comments.

4) Add support for short phrases

Add support for short phrases such as “thank you” and “see you later” which currently cannot be processed.

5) Add support for more advanced sentence structures:

Add support for **Dependent clauses**, as is displayed between brackets in “could you tell me [where the hall is]?”

Add support for **Multiple dialogue acts per user turn**, as in the utterance “hello, where is the hall?” and “I’m lost, where is the exit?”.

Besides improving recognition, two other secondary goals were pursued:

6) Improve replies on out-of-domain questions:

See section 2.8 for a problem description. Improve clarity of out of domain replies.

7) Improve the ontology:

Investigate if and how the ontology (see 2.3.3) can be fixed and improved.

3.2 Corpus Analysis

In order to increase the recognition rate, the following information was needed:

- Word usage (how does the user refer to objects in the environment?)
- Short phrases (what short phrases are used?)
- Sentence structures (how are user utterances formed?)

3.2.1 The Bussink Corpus

Available for data analysis was a corpus created for an experiment by Dirkjan Bussink in 2006 [Bus05]. The purpose of this experiment was a performance test in speech recognition.

The corpus was created in a Wizard-of-Oz style experiment in which a user had to perform several tasks in the VMC environment. The raw audio collected in these experiments was transcribed into text files. This corpus was (partly) usable as research material for this project because the setting and user goals (finding the way in the VMC by interacting with a guide) were comparable to that in our environment.

Transcriptions of utterances from 13 different sessions, with 13 different users were available. The transcriptions consists of around 4500 words. The users all received one or more goals to achieve, such as locating a certain room in the building. A sample of the corpus is displayed in figure 3.1.

The transcriptions start with the user reading the goal to accomplish (in this example to find out how many trees there are 'outside' of the VMC), followed by user comments while he or she is trying accomplish this goal.

Other examples of user goals were *“find the toilet”* and *“how many coffee counters are there in the building?”*. Because of these goals, the corpus contains user answers such as *“the toilet is over here”*, *“I found the toilet”*, *“there are three toilets”* and *“there’s another desk upstairs”*. Of course, not all of these use utterances were relevant to this project. Statements like *“I found the toilet”* could be interpreted as a user confirming he or she found the object the Guide told the location of. But other statements such as on the number of coffee desks (*“there are three coffee desks”*) are only relevant in the test environment, and were discarded. Other discarded user utterances were of the user asking specific route information such as *“do I have to go left here?”*. These questions cannot be answered in the current version of the Virtual Guide, and adding this functionality does not lie within the scope of this project.

```
volgende opdracht,  
hoeveel bomen staan er buiten  
oke is goed hoe kom ik naar buiten  
waar is de hoofdingang  
oh waar is waar is de buitendeur  
{ 'k ik } weet al hoe ik naar beneden moet lopen  
oke  
dus hoe kan ik naar beneden  
moet ik hier linksaf  
oke moet ik hier linksaf  
nou, oke  
oh nee  
moet ik hier rechtdoor  
mmm moet ik hier linksaf  
ja  
mmm moet ik hier rechtdoor  
nee moet ik hier rechtdoor  
ja  
moet ik hier linksaf  
ja  
nou volgens mij zijn er twee bomen  
ja  
nou mooi
```

Figure 3.1: A sample of the Bussink corpus

3.2.2 Results

After the analysis was complete, the following data was gathered:

New words:

- buitendeur (object Frontdoor)
- hoofdingang (object Frontdoor)
- buiten (object Frontdoor)
- aha (affirmation interjective)
- ach (negative reply)

New multi-word expressions (see section 3.7.2:

- dank u (thank you)
- nou mooi (that's great)
- even kijken (pause)
- en nou? (ask for advice)
- en toen? (ask for advice)

New sentence structures (besides combined dialogue acts):

- ik wil naar ... (I want to go to ...)
- ik moet naar ... (I have to go to ...)
- ik ben op zoek naar ... (i'm looking for ...)
- dat is mooi (That's great)
- ik ben verdwaald (I'm lost)
- kunt u me de weg wijzen naar... (can you direct me to ...?)

This data was subsequently used in the robustness improving process described in the next sections.

3.3 Improving Dialogue Act Recognition

After the analysis of the implementation and research data was complete, the improvements for robustness could be added to the Guide. The following sections discuss these improvements in lexicon and action templates.

3.3.1 Lexicon

The first (and simplest) way to increase user input recognition was to add more synonyms of verbs and objects in the world to the lexicon. For example, in the original implementation, only the word “toilet” could refer to the toilet object in the VMC, and only that word was present in the lexicon. What seems obvious and what was also concluded in user tests, was that users will not confine themselves to only this word. The same holds for verbs like ‘explaining’ and ‘describing’ as synonyms of the verb ‘telling’ applied in a question like “*could you tell me where the exit is?*”.

The number of synonyms for objects and verbs in the lexicon were increasing the to improve the recognition. The synonyms were found in the corpus, dictionaries [Taa07] and, later, from user tests (see section 3.8). See 3.1 for a number of example synonyms that were collected.

Original term	Example Synonym
toilet	wc, plee, closet
aanwijzen	tonen, markeren, aangeven
hallo	hoi, hi, goedendag

Table 3.1: Some examples of collected synonyms

3.3.2 Action Templates

The second point of improvement was the list of action templates. This list is an XML-file that contains the templates used to match different parses of user input and connects them to system actions (as described in section 2.6). Originally, only a few variations of feature structures were implemented. This meant the Virtual Guide was unable to recognize all but some basic sentence structure variations. For example, questions formed as “*where is X?*” (with X representing the referred object in the world) were recognized, but not “*where can I find X?*”.

As discussed in section 2.4.3, the action templates represent SParses which in turn represent text input parsed by the grammar. To create new action templates, SParses were needed to model the templates after. Also, later on, new SParse types were to be constructed and tested. To avoid the long starting process of the Virtual Guide program each time an SParse was needed or to be tested, a program was written that was quickly able to generate and

write the SParses to a log file. Besides this new program, a locally available program called *Dug* was used to show parse trees and feature structures generated from input.

Often, more than one parse of a user input is possible. Because some words in the lexicon have multiple interpretations or types, this will result in a separate parse for each different interpretation. For instance, in Dutch, the verb “zijn” (being) can either refer to an object’s properties, its position or the existence of an object. Only one interpretation actually applies in each situation. After the correct parse is selected from the list of found parses by hand, its feature structure is added to the action template file, together with the forward tag. The feature structure is subsequently connected to a system action. See also section 2.6.

Some actions require a part of the parse as argument (see section 2.6.2). For example, when the Virtual Guide receives a parse of the sentence “*Where is the toilet?*”, the corresponding action, *ActionTellLocation* is called. This action makes the Guide mark the object on the mini map. In this case, the action requires an argument: an object to point to. As such, part of the input, namely: “*the toilet*”, is submitted as argument (see section 2.6.2). In order to make this happen, a pointer to the element of the feature structure that contains the reference to the object is added to the action template. See figure 2.6 for an example of an action template.

By following the described method, user sentences from the corpus were used to add more feature structures to the action templates to increase the dialogue act recognition. A list of added action templates (which was also expanded in the following sections) can be found in appendix B.

3.4 Dependent Clauses

The original implementation did not support dependent clauses. An example of a dependent clause is displayed between brackets in the following example:

Kun je me vertellen [waar het toilet is]?

or in English:

Could you tell me [where the toilet is]?

This section describes the implementation of dependent clauses in the Virtual Guide by making changes to the grammar and the SParse creation process for these cases.

3.4.1 Grammar

This sentence can be separated into two parts. The first part (“*could you tell me*”) was parsed with the original grammar without problems. The second part (“*where the toilet is*”) can be interpreted as a normal WH-Question (“*where is the toilet*”), except for the difference in word order. By transforming the dependent clause into such a normal sentence, the the action templates (see section 2.6.2) that were in use for normal sentences could be reused for dependent clauses, effectively saving a lot of time and increasing efficiency.

To add support for dependent clauses, the grammar had to be adjusted first. The following grammar rule (see also section 2.4.1) was in use:

```
Z -> S
S -> V NP VP BSE (stype = jnq)
```

Where Z is the root, S is a sentence, V is a verb, NP a noun phrase and VP a verb phrase. BSE represents the dependent clause. This rule was usable as it was, but no proper rule existed to parse a BSE element that starts with an adverb. This is necessary for our dependent clauses (starting with “*waar*” (where)). Thus, the rule for parsing this kind of dependent clauses was added:

```
Rule{bse}
  BSE -> ADV SB
    <BSE head> = <SB head>
    <BSE sem adv> = waar
    <BSE sem main> = <SB sem>
    <BSE stype> = <SB stype>
```

The rule splits the BSE into an adverb (ADV) and an SB (the main part of the dependent clause), which in turn splits into an noun phrase and a verb phrase. The parse tree is displayed in figure 3.2.

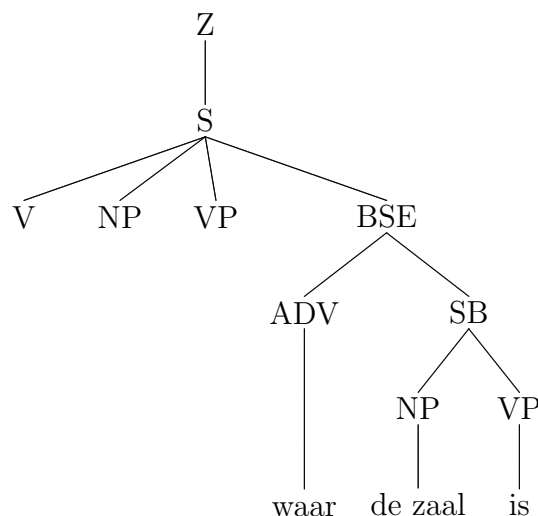


Figure 3.2: (simplified) parse of the dependent clause “waar de zaal is”

As stated before, the dependent clause has, when taken literally, the same function as the normal question “*where is the room?*”. By using the BSE to parse the sentence, the feature structure in figure 3.3 was created. See section 2.4.2 for information on feature structures.

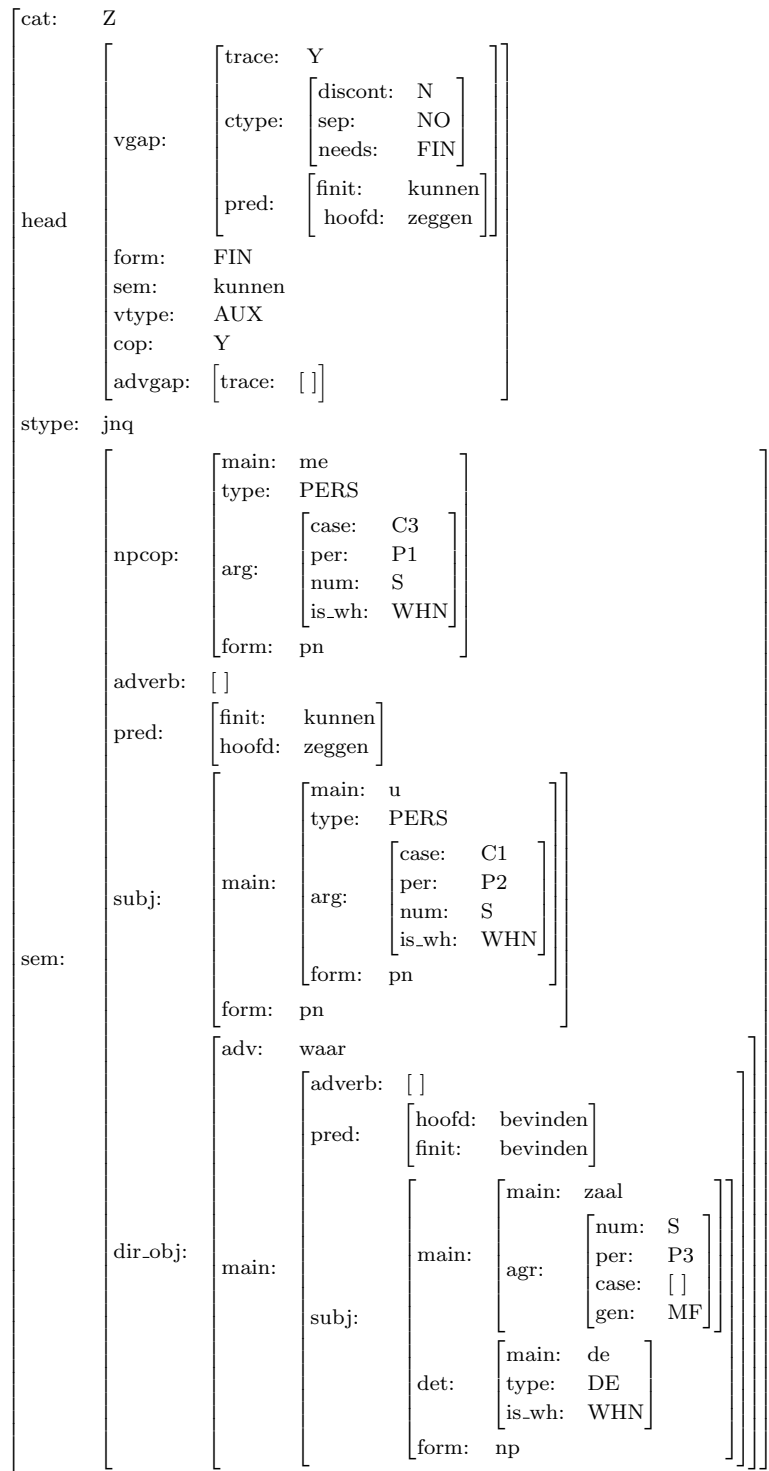


Figure 3.3: The feature structure for the sentence 'kunt u me zeggen waar de zaal is?'

```

<s_parse id="5" left="4" right="8" surface="WHQ">
  <subj>
    <np_parse id="3" left="5" right="7">
      <det>
        <det_parse id="1" left="5" right="6" type="DEF" det="DE"/>
      </det>
      <head>
        <n_parse id="2" left="6" right="7" noun="zaal"/>
      </head>
    </np_parse>
  </subj>
  <aux_verbs>
    <verb>bevinden</verb>
  </aux_verbs>
  <main_verb>bevinden</main_verb>
  <adv_pps>
    <adv_parse id="4" left="4" right="5" adv="waar" is_wh="true"/>
  </adv_pps>
</s_parse>

```

Figure 3.4: The SParse for the sentence “kunt u me zeggen waar de zaal is?”

3.4.2 SParses

When the example sentence (“can you tell me where the hall is?”) is taken literally, it is a yes/no-question, and it will be tagged as such. The converter that builds the SParses (see section 2.4.3) is modified so that when a feature structure of sentence type yes/no-question is detected, it can expect a dependent clause. The whole dependent clause is contained in the direct object clause of the feature structure, *dir_obj*. The processing starts when the *dir_obj* representing the dependent clause is isolated from the rest of the feature structure. Next, the isolated feature structure is transformed into an SParse of a ‘normal’ WH-Question. To allow this, many content retrieval functions for SParse converting had to be adjusted so they would work on dependent clauses. For instance, a function for retrieving a subject from a sentence had to be adjusted to work on dependent clauses.

After all code adjustments were made, the following SParse of figure 3.4 could be created:

As planned, this SParse could be recognized by existing action templates (see section 2.6), in this case the template for “waar is de zaal?” (where is the hall?).

3.5 Multiple Dialogue Acts per User Turn

A more significant improvement to the Virtual Guide in terms of amount of changes and additions made to the source code, was adding support for multiple dialogue acts in a single user turn. A list of user dialogue acts can be found in section 2.5.1.

An example of a multiple dialogue acts in a single user turn is the utterance:

“ik ben verdwaald, waar is de zaal?”
(I’m lost, where is the hall)

and:

“hoi, ik zoek de wc”
(hi, I’m looking for the bathroom)

Originally, only one user dialogue act per user turn could be handled. Based on research of the corpus (see section 3.2), it became clear that combined dialogue acts made up a significant percentage of user turns (10-20% of all the WH-Questions in the corpus).

For Example:

“ik ben verdwaald, waar is de grote zaal”
Which is a statement (I am lost) + a where question (where is the great hall)

By adding support for these utterances, the recognition rate of queries should subsequently increase with said percentages. This section describes the modifications that were made to realise this feature.

3.5.1 Grammar

The original grammar (see section 2.4.1 for a description of the grammar) for parsing a complete sentence began as follows:

```
Rule{s}
  Z -> S
```

Where Z was the user utterance being parsed and S (sentence) represented the parse of a sentence. The S was then parsed further. The following splitting rule was added to accommodate multiple dialogue acts:

```
Rule{split}
  Z_0 -> Z_1 Z_2
```

Meaning: an utterance can consist of two separate utterances, which, in turn, can be split into two other utterances, and so on. Now, an utterance was allowed to consist of multiple

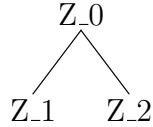


Figure 3.5: The split rule

sentences, but the contents of the elements still had to be properly parsed. The parse tree for “*ik ben verdwaald, waar is de zaal?*” (I’m lost, where is the bathroom?) is displayed in figure 3.6. It shows the two separate parse trees for each dialogue act connected by the new grammar rule.

In order to append the correct contents in the feature structure, the split rule was expanded as follows:

```

Rule{split}
  Z_0 -> Z_1 Z_2
    <Z_0 stype> = comb
    <Z_0 sem fst sem> = <Z_1 sem>
    <Z_0 sem fst stype> = <Z_1 stype>
    <Z_0 sem fst head> = <Z_1 head>
    <Z_0 sem last sem> = <Z_2 sem>
    <Z_0 sem last stype> = <Z_2 stype>
    <Z_0 sem last head> = <Z_2 head>

```

Using this rule, each combined sentence feature structure is separated in a first (**fst**) and a last (**last**) part, and assigned the sentence type **comb**. Whenever more than 2 dialogue acts are parsed, the third and latter dialogue acts are recursively nested in the first two feature structures representing the dialogue acts. The feature structure for “*ik ben verdwaald, waar is de zaal?*” is shown in figure 3.7.

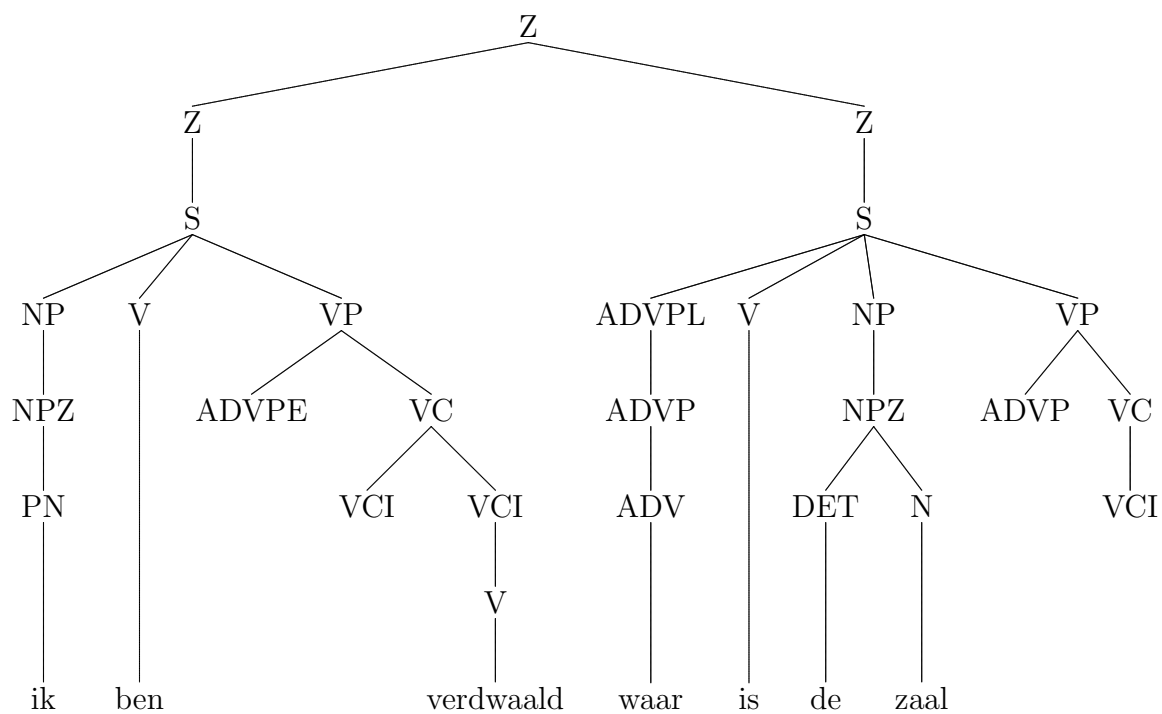


Figure 3.6: Parse tree of the combined dialogue act clause ‘ik ben verdwaald, waar is de zaal?’

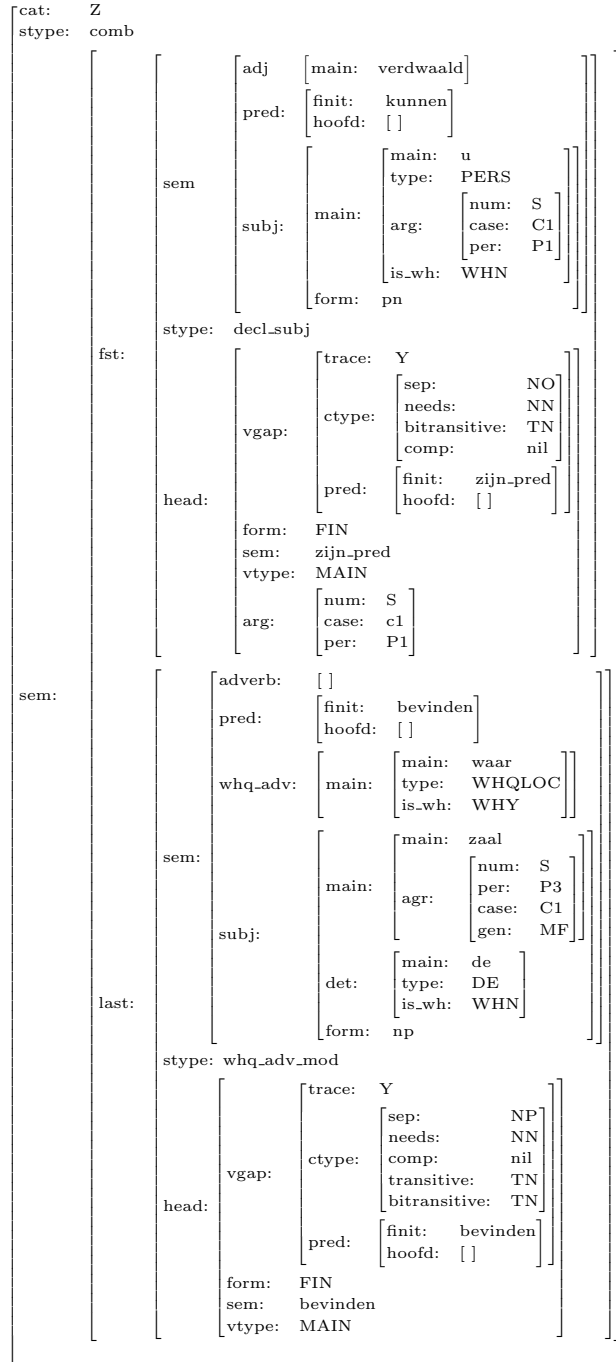


Figure 3.7: The feature structure for the sentence 'ik ben verdwaald, waar is de zaal?'

Besides the first Z rule, the grammar also had an existing rule considering sentences that start with an interjection:

```
Rule{i_comma_s}
  Z_0 -> IJECT PUNCT Z_1
```

Meaning: an utterance can start with an interjection, followed by a comma and another utterance. This rule was applicable to for example: *Hallo, waar is de zaal?* (Hello, where is the room?). It must be noted that the interjection part of the sentence would be discarded. Because the new split rule was more general and made this rule obsolete, it was removed from the grammar.

3.5.2 SParses

After the changes were applied that were described in the preceding sections, the grammar was able to properly accept multiple dialogue acts in a single user turn. This was tested by entering combined sentences from the corpus, such as:

“ik moet naar het toilet waar kan ik die vinden”
(I have to go to the bathroom, where can I find this?)

Next, the system had to be adjusted so that a proper SParse could be generated for utterances with combined dialogue acts. See section 2.4.3 for a description of the SParse. This was accomplished by attaching the sentence type **comb** to the combined sentence using the grammar. Now the parse converter could be used to recognize the sentence containing multiple dialogue acts by this sentence type. A method was written to retrieve the separate dialogue acts from the combined sentence feature structure. These are then each converted to SParses and placed into a list. This list is subsequently used to create an assembled SParse for the combined sentence. The structure of this new SParse is displayed in figure 3.8.

In some cases, the grammar found both a single SParse and two separate **combs** SParses for an utterance (and as such the system finds either a single or two dialogue acts). This happens in the following example:

“Hoe kom ik bij de zaal?” (how do I get to the hall?).

This utterance can be parsed in two main ways (with each having minor variations in parsing further down the tree), as shown in figure 3.9a and b. Figure 3.9a shows the parse tree for a single SParse, while 3.9b shows the (incorrect) parse tree which results in two separate **combs** SParses. Experience shows that the single SParse, as is the case in this example, usually is the only correct parse. The implementation of the dialogue act determiner has been set up to always prefer the single SParse interpretation above the multiple **combs** alternative to maximize the amount of correct parses.

```

<s_parse>
  <comb>
    <s_parse>
      Parse of the first dialogue act
    </s_parse>
    <s_parse>
      Parse of the second dialogue act
    </s_parse>
    <s_parse>
      . . . etc . . .
    </s_parse>
  </comb>
</s_parse>

```

Figure 3.8: SParse structure for combined dialogue acts.

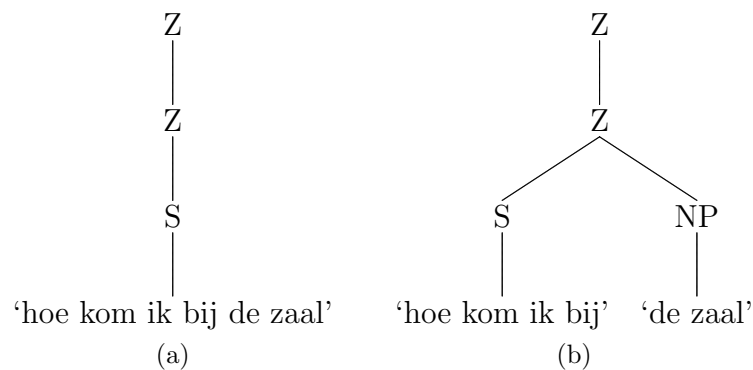


Figure 3.9: Alternative (simplified) parse trees of 'hoe kom ik bij de zaal'

```
User Utterance:  "hi, where is the exit?"

Text of dialogue act A: [hi]
Text of dialogue act B: [where is the exit]

Results after parsing and converting:  3 combined SParses with each 2
interpretations of the 2 dialogue acts

Each SParse is analysed and possible dialogue acts are placed in a list:
Results SParse 1 :  <<A1>,<B1>>
Results SParse 2 :  <<A2>,<B2>>
Results SParse 3 :  <<A2>,<B2>>

With Ax and Bx representing the interpretations for SParse x of user
dialogue act A and B respectively.

Final merged dialogue act list:  <<A1 + A2 + A3>,<B1 + B2 + B3>>
```

Figure 3.10: Merging the lists of interpretations of user utterances into a final list that is used to determine the user dialogue acts.

3.5.3 Turns

The final task is to create one or more system turns for the combined dialogue acts (see section 2.3.5). For each different parse of a user utterance, multiple possible interpretations of dialogue acts are selected and placed in a list. For example, when the user input is “Hi, where is the exit?”, 3 alternate SParses are generated based on different interpretations of the grammar. These parses are sequentially analyzed in the dialogue act determiner to determine their type and to retrieve parts of their contents needed for processing (see section 2.5 for a description of this process). Because the example sentence represents 2 dialogue acts (*OPEN* and a *WH-Question*), this will result in 2 lists of possible interpretations of dialogue acts per SParse. Because 3 SParses were generated, 3 main lists with each 2 sub-lists of dialogue acts is the result. These 3 lists are then merged, resulting in a final list of interpretations that is used in determining the next system turn. In contrast to single dialogue acts, the merging of these lists is a little more complex. Figure 3.10 displays this merging process.

After selecting the best fitting pair of interpretations from the final dialogue act list, user dialogue acts *OPEN:NULL* and *WHQ:NULL* are selected. This user turn results in the corresponding 2 system dialogue acts: an answer to the user greeting and an answer to the question of where the room is. This user turn is displayed in figure 3.11.

```
Alternative 1:
| Dialogue act A:
|   | Forward tag: OPEN
|   | Backward tag: NULL
|   | Utterance: hi
| Dialogue act B:
|   | Forward tag: WHQ
|   | Backward tag: NULL
|   | Utterance: where is the exit
```

Figure 3.11: An interpretation of "Hi, where is the exit"

For each of the system dialogue acts in the user turn, a system action is put on the action stack (see section 2.3.5). However, when processing multiple dialogue acts, the first system action that is selected is also placed on the stack first, and so on. So, when popping multiple actions from the stack, the ordering is reversed. This, at first, resulted in the system replying to the dialogue acts out of order.

To avoid this problem, the dialogue manager places the the dialogue acts on the stack in reversed order. The situation including the contents of the stack after the user dialogue act and the history order before and after this adjustment is show in figure 3.12. This figure shows that in the original situation the OPEN:ACK system dialogue act is shown before STATEMENT:ANSWER. The situation after the stack adjustment shows the system dialogue acts executed in the correct order. Adjustments were made to make sure the history (see section 2.3.4) remained unchanged. If the history would be left reversed as the execution of actions are, it would show an incorrect system state.

Situation before adjusting history:

U: hoi waar is de koe (OPEN:NULL)

U: hoi waar is de koe (WHQ:NULL)

S: Ik heb de zaal (zaal) gemarkeerd op de kaart. (STATEMENT:ANSWER)

S: Hallo, zeg het maar. (OPEN:ACK)

Stack: WHQ
 OPEN
 _____/

History: OPEN, WHQ

After reversing history order:

U: hoi waar is de koe (WHQ:NULL)

U: hoi waar is de koe (OPEN:NULL)

S: Hallo, zeg het maar. (OPEN:ACK)

S: Ik heb de zaal (zaal) gemarkeerd op de kaart. (STATEMENT:ANSWER)

Stack: OPEN
 WHQ
 _____/

History: OPEN, WHQ

Figure 3.12: Issue resolvment with history and multiple dialogue acts in a user turn. U is a user turn, S is a system turn.

3.6 New User Dialogue Acts

In this stage of the project, only minor user dialogue act types were added to the existing list (see section 2.5.1). First of all, a dialogue act to thank the Guide. Secondly, 2 user dialogue acts to give either positive or negative comments. System actions were created to reply to each of the new user dialogue acts.

3.6.1 Thanking

The framework for the action in which the user could thank the Guide was already there. After some small changes it could be used, and the user was able to thank the Guide. The only thing that had to be done, was to add a link to the *ActionThank* (see table 2.4) to the action template (see section 2.6.1'). It can be found in appendix B. After a user expresses his or her gratitude, the guide will answer with “*you’re welcome*”.

3.6.2 Positive and Negative Feedback

Adding two new dialogue act types from scratch took a little more effort. The goal of this addition was that user would be able to give feedback by saying phrases such as: “*that’s good!*” or “*that’s awful*”, and “*nice!*”. The system reply for a positive comments would be “*at your service*”, and for negative comments “*oops!*”.

First of all, a number of changes were made to the lexicon (see section 2.4.1 for more information on lexical entries and types). Because interjections such as “*good!*” and “*awful!*” are to be considered a positive or negative comment, these words received a special tag for the semantic type in the lexicon: POSITIVE or NEGATIVE. This new type of interjection had to be introduced into the dialogue act determiner to allow them to be parsed correctly and subsequently recognized as either positive or negative user dialogue acts.

Two new forward tags (see section 2.5.1) had to be added as well: POSITIVE and NEGATIVE. New entries were made in the dialogue act preferences map (see section 2.5) so the Guide would be able to handle the new types. Action templates (see section 2.6.1) were constructed for dialogue acts with the new forward tags.

A different solution was needed for sentences such as “that is good”, where “good” is an adjective. As with interjections, a type was added for adjectives in the lexicon: POSITIVE, NEGATIVE. The default type is OTHER (when there type is no type available). Adjustments were made in the grammar to include the type of the adjective in the feature structures. Further adjustments were made in the functions needed to construct an SParse (see section 2.4.3). These functions retrieve the new adjective type from the feature structures. Lastly, action templates were constructed which, depending on the adjective type, either selected *ActionPositive* or *ActionNegative* as system action.

Two new system action classes (see section 2.6) were constructed to generate a system reply.

These action classes are based on existing action classes, and result in the presentation of a canned sentence, mentioned at the beginning of this section.

Next was adding action templates (see section 2.6.1 to the action template file. These can be found in appendix B.

After completing the changes, the user was able to express his or her opinion about the Guide.

3.7 Miscellaneous Changes

This section describes work on various elements of the Virtual Guide that were not discussed in previous sections.

3.7.1 Replies to Out of domain Questions

If the Virtual Guide receives a question that cannot be answered, for example when the user is not understood, a specific system action is called (see section 2.6). These actions are `ActionUninterpretable` and `ActionNoMatch`. The responses connected to these actions consist of *canned* (predefined), static sentences. An issue with the original sentences was that they were somewhat unclear. For example, the sentence “*I did not understand you*”, did not shed much light on why the user was not understood, or on what the user did ‘wrong’. Should the user try to formulate his or her question differently, or should the user just repeat himself?

The issue of selecting the proper out of domain reply is discussed by Patel et al in [PLT06]. This paper distinguishes different out of domain classes. Proposed classes that apply to the Virtual Guide system are *Don’t understand*, *Out of domain* and *Unknown* and are shown in table 3.2. The Don’t understand class covers questions that are uninterpretable. The Out of domain class covers the questions that ask something that is not about the topics the character is prepared to talk about. This happens when the Guide is able to ‘understand’ (being able to properly parse the utterance with lexicon and grammar) what the user says, but does not know what to do with it. The Unknown class is for questions that concern the domain, but for which the character does not have an appropriate answer. In our case, asking for navigation related task concerning an object that is unknown in the world.

The sentences that cover these kind of out of domain replies now describe what is the matter. For example, when the user asks for an object that does not exist in the Muziekcentrum, the Virtual Guide will respond with “*this object isn’t in the Muziekcentrum as far as I know. If you are sure it is, could you rephrase it?*”. It must be noted that the unknown object that the user refers to, must be present in the lexicon. Otherwise, the user utterance is considered uninterpretable. Other examples are shown in table 3.2.

Another situation manifests when the user asks for an object that has more than one instance in the world. For example, there are more than one table in the Muziekcentrum. The same goes for doors, chairs, and stairs. Before, the system response would be “*could you be more specific as to where you want to go to?*”. This description is vague, and a better solution was investigated. The new system response is as follows: “*could you be more specific as to what you are looking for? I don’t know which one of the [object] you mean. Choose one: [list of objects]*”. Although the fact that the Guide is unable to select the right instance is made clear to the user, the user is as of yet not able to refer to these objects the same way the Guide names them. This is caused by the world ontology (see section 2.3.3) that cannot be accessed properly in this implementation and is considered future work.

Class	Old Reply	New Reply
Don't understand	Ik heb u niet verstaan	Ik begrijp niet wat u bedoelt, probeer het eens anders te formuleren alstublieft
Out of domain	Ik verstond: [user utterance] maar ik weet niet wat ik hier mee moet doen	unchanged
Unknown	Wat zoekt u?	Ik geloof niet dat we dit hebben in het muziekcentrum. Misschien heb ik u verkeerd begrepen. Als u zeker bent van uw zaak, kunt u het dan eens anders proberen te formuleren?

Table 3.2: Out of domain replies

3.7.2 Multi-word Expressions

In everyday life, short sequences of words such as “*hi there*” and “*thank you*” are used often. The IRME Project (Identification and Representation of Multi-word Expressions)[dJO07] calls these phrases *Multi-Word Expressions*, or MWEs. IRME’s MWEs include multi-word terms (‘grammar book’), idioms (‘he kicked the bucket’), collocations (‘red roses’), support verb constructions (‘pay a visit’), phrasal verbs (‘he showed up’), sayings (‘live by the sword, die by the sword’), and so on. The IRME group states that these MWEs are an important bottleneck for NLP-applications such as question answering and information retrieval and the IRME project is researching solutions to the issue of MWE recognition and lexical representation. In this project, the MWEs will not include idioms or sayings, but mere grouped words that do not have a deeper, figurative meaning to them. This means that no special lexical representation of meaning is needed. The raw utterance (text) will be kept intact for later processing (see chapter 4), which also removes any need for the representation of special properties in the lexicon. This project also limits to MWEs that have a fixed sequence of words and that can be represented by a synonym that consist of a single word. This removes any need for a complicated deeper analysis of potential figurative meaning of a phrase.

The original implementation of the Virtual Guide did not have grammatical support for any of these short phrases. Adjusting the grammar of the Guide was unpractical because of the great number of variations in grammatical structure of these expressions. Adding the

phrases to the lexicon was not an option either, because it only supports single words. So, to avoid having to add rules for each of these different phrases to the grammar, a different and simpler solution was used.

It was decided to replace the phrases with a single word that conveys the meaning of the corresponding word sequence. This word represents the MWE and is present in the lexicon. This solution solved any recognition problems.

The processing of multi-word expressions is described next. First, a list of MWEs was collected. These phrases came from the corpus analysis (see section 3.2) and later on from user tests 3.8). Nlparser (see section 2.3) is responsible for parsing the user input. First of all, a lexicon was created containing all the MWE's distributed in MWE groups. Each MWE group has its own lexical representation. Examples of phrase groups are displayed in 3.3. When user text input is entered, the *multi-word expressions lexicon* is used to look for MWE's. The MWE's in the raw input are subsequently replaced by the corresponding lexicon word for the group the MWE belongs to. A complete list of the lexical entries of the MWE lexicon can be found in appendix A.

Phrase Group	Example Phrase	Lexical representation
open	<i>hallo daar</i>	<i>hallo</i> (hello)
close	<i>tot ziens</i>	<i>doei</i> (bye)
thank	<i>dank je wel</i>	<i>bedankt</i> (thanks)
positive	<i>goed zo</i>	<i>goed</i> (good)
negative	<i>nou ja zeg</i>	<i>slecht</i> (bad)

Table 3.3: Phrase classes, examples of phrases and their lexical representation

This method makes it relatively easy to parse MWEs and to add support for new MWEs and new MWE classes by simply adding them to the MWE lexicon.

3.8 Usability Tests

In order to validate the changes and additions made for robustness described in this chapter, a number of limited user tests were held. These test were only organised to get an indication of the functionality of new system additions and increase data for the lexicon, MWE's and sentence forms.

3.8.1 Goals

The usability test has the following goals and subgoals:

1. **Are the additions for robustness functioning?**
 - Are dependent clauses functioning?
 - Are combined dialogue acts per user turn functioning?
 - Are the users informed properly by the Out-of-domain replies?
2. **What are currently the biggest limitations in this implementation?**
 - Question asking
 - Does the system understand the user?
 - * What words are not in the lexicon?
 - * What sentence structures are not in the grammar?
 - * What actions should be added?
 - Question answering
 - Is the answer satisfying?
 - * Does the user understand what the Guide means?
 - * Does the Guide give correct answers?
 - How well is the system coverage (out of domain or not)?
3. **Retrieve data:**
 - List of new words
 - List of new sentence structures
 - List of new multi-word expressions

3.8.2 Setup

Target group:

The target group consisted of students (aged 20-26). The number of participants was 3. The group had experience in working with computers and instant messaging.

Material:

The implementation described in this chapter (without the alignment model of chapter 4) of the Virtual Guide, installed on a personal computer with internet access and the ability to play audio.

Realization:

The test was realised by logging a session of each test participant with the Virtual Guide. The Virtual Guide was installed locally on a personal computer. The participant received instructions before the start of the test. He or she was told the purpose of the Virtual Guide, what it can and cannot do, and introduced to the VMC environment. The participant was asked to assume the role of an VMC visitor, who has a goal to accomplish. The participant was allowed to explore the 3D world while completing these goals.

A selection of possible goals is listed here:

- Try to find the bathroom/room/exit on the map
- Try to find some place where coffee is served
- Try to get the *route* to the bathroom/room/exit objects
- Finally, each participant was asked to try the system out by asking for whatever they could come up with and enter it into the console.

Each participant started a conversation with the Virtual Guide by entering his or her utterances with the keyboard. Speech recognition did not play a part in this, but speech synthesis was used in the system's answers. The text of the conversations together with the internal system output were logged in a file for later analysis. The tests took about 15 minutes per participant.

3.8.3 Results

The usability tests had the following results:

1) Are the additions for robustness functioning?
- All new sentence forms (dependent clauses and combined user utterances) were used by the test subjects and functioned properly for the sentence forms that were implemented. - A user user complained that the system did not repeat the word that was not understood, making it was difficult for the user to understand how he or she should improve the utterance.
- Out of domain questions such as: “is this the end?” were covered by a proper reply (see section 3.7.1).
2) What are currently the biggest limitations in this implementation?
<i>General issues</i>
- See table 3.5 for question forms that were not recognized.
- All routing and location answers were displayed correct, except for the route to ‘de zaal’, this error is outside this project’s domain.
- The lack of error correction in user input
- Crashing of the system when entering numbers like “123” which was discovered when the user was allowed to try the system out.
<i>What actions are missing?</i>
- System apologize. This might also be combined with ActionPositive (see section 3.6)

Table 3.4: Usability test results

Failed user input recognition	Reason
“wat is de weg naar de wc” (what is the way to the bathroom?)	No support for ‘what is the way’
“wat is de route naar de wc” (what is the route to the bathroom?)	No support for ‘what is the route’
“stomme computer” (stupid computer)	Not in MWE list
“wat is de route?” (what is the route?)(after initial question about location of an object)	Not supported
“waar is de zaal en het toilet?” (where is the hall and the bathroom)	Structure not supported
“je bent mijn held” (you’re my hero)	Not in MWE list
“dat is niet best” (that’s not good)	Not in MWE list
“hoe kom ik bij de toiletten?” (how do I get to the toilets?)	‘toilets’ not in lexicon
“waar is de bioscoop?” (where is the cinema?)	‘bioscoop’ not in lexicon

Table 3.5: Failed user utterances

New words
bioscoop, toiletten, de weg, de route
New Multi-word expressions
stomme computer
dat is [niet best]
je bent mijn held
New sentence structures:
wat is de route
wat is de weg naar de zaal
waar ben ik, hoe kom ik bij de toiletten
waar is de zaal en het toilet?
hey, waar is de zaal

Table 3.6: New words, MWEs and sentence structures

3.8.4 Conclusions

After studying the test logs, the performance of the system can be characterised as follows:

Combined sentences were used, although less than the 20% found in the Bussink corpus (see section 3.2). The bulk of the wh-questions could be processed and answered by the Virtual Guide. Some new words and sentence structures were collected. These can be found in table 3.5. Support for some of these newly found forms were added to the system.

The only serious issues of instability manifested when the user tried to enter numbers into the console. This bug that discovered when a user was allowed to try the system out. A user also expressed that sometimes the answers of the Guide when a word is out of lexicon are somewhat unclear. Perhaps the Guide should say exactly what word was not understood, to allow the user to try again. This remains to be implemented.

The final list of supported sentences is given in appendix C.

CHAPTER 4

Alignment

4.1 Introduction

This chapter presents an expansion that was made to the Virtual Guide: the addition of an **alignment model**. This model uses *language alignment* which is based on *politeness* and *formality of language* for the analysis of user utterances and *natural language generation* of system replies.

The alignment model's goals are twofold: firstly to add *language alignment* and secondly to increase *variation* in the Virtual Guide's utterances for greater character believability.

The **character believability** of a character (in our case the Virtual Guide) refers to the elements that make a character life-like. Elements of a life-like character can include personality, emotion, expression of intentionality and movement [Bat94]. Character intentionality is the manner of choices and actions that make a character appear natural to observers. The intentionality is a result of exchanges between the beliefs, desires, intentions as well as internal and external motivations of the character. It makes what the character does and say seem more rational and thus realistic.

Language alignment is something that happens automatically in dialogues between humans. Implementing this, together with more variation in interaction will increase the believability of embodied conversational agents, as is discussed by Walker et al. [WCW96].

As said, **Politeness** and **formality of language** are the basis for the alignment process in this model. Formality of language plays an important role in determining social distance between two speakers [HD99]. In order to decrease this social distance and thus increase a

sense of interaction, the Guide will try to align its formality of language to that of the user. The Guide has variations in formality to choose from. Politeness has an important impact on the way questions are asked and answered [BL87]. Asking the Guide politely when using the politeness model will result in a polite answer, while a rude question will result in a less polite answer. The Guide will choose between different politeness tactics, depending on how polite the user addresses the Guide. Applying this alignment should result in a more realistic and varied style of answering.

A general description of theories and literature on alignment of language, politeness and formality of language will be given at the beginning of this chapter. After this, a discussion of related work will take place. Finally, a description of the alignment model based on the discussed literature and inspired by related work and a description of the realisation process will be given in the final sections of this chapter, followed by examples of generated dialogue and a general discussion.

4.2 Alignment of Language

Garrod and Pickering [GP04] asked themselves why conversation is so much easier to do than presenting or listening to speeches. In dialogues, alignment of language is for instance realised by using ellipsis in order to match the answer to a question. Example: “what time is it?”. An aligning listener would answer with “five O’clock” instead of “the time is five O’clock” because the first answer reuses the structure of the question. They present the explanation that this is because of the interactive alignment of linguistic representation between partners that takes place. The interactive alignment of language is caused by the interactive nature of a conversation. The workload of the conversation is distributed between the interlocutors because the information computed by the other is reused. Garrod and Pickering state humans are designed for dialogue, rather than monologue because of the automatic alignment channels that are present in humans.

Levelt and Kelter [LK82] also conclude that speakers tend to reuse material from previous talks so it does not have to be regenerated. Their findings indicate that the surface form of a question can effect the form of the answer, even when this alignment has no semantic or conversational consequences. This ‘correspondence effect’ (see the example of “five O’clock” versus “the time is five O’clock”) may diminish when additional verbal information is presented to the listener.

Branigan et al. [BMJPMC06] report an experiment in which they researched the alignment between computers and test subjects. The experiment was set up so that the test subject were told they would either talk to a person or a computer. In fact, in both cases the test subjects were conversing with a computer program with pre-scripted utterances. Test subject displayed strong language alignment in typed communication by repeating the syntactic form of the preceding utterance of the computer program. It did not matter if the subjects believed they were in dialogue with a computer or a real person. Other research concluded that there was no difference alignment if the form of dialogue was in text or in speech. They finally conclude language alignment to be an automatic process and is not effected by the mental state of the interlocutors.

Politeness	Strategy		Example
<i>Low</i>	On-record		
↓	<i>Bald:</i>		“Give me a drink!”
↓	<i>Redressive:</i>	1: Positive politeness:	“Hey pal, would you give me a drink?”
↓		2: Negative politeness:	“I wondered if you could give me a drink.”
↓	Off-record		“This talking always makes me thirsty.”
<i>High</i>	Don’t do FTA		Do nothing’

Table 4.1: Examples of Brown and Levinson’s strategies

4.3 Politeness

The *Theory of Politeness* proposed by Brown and Levinson ([BL87]) claims that politeness is applied by a speaker to save the hearer’s **face**. This section will discuss this theory. Specific attention will be given to how politeness is expressed in the Dutch language, based on the work of Roel Vismans [Vis94a] [Vis94b].

4.3.1 Face

Face is a public self image that every person wants to pursue. The concept of face is divided in positive face and negative face. **Negative face** states that a person has a need for autonomy from others. **Positive face** is the need for a person to be approved of by others. Whenever a speech act goes against what the face (the self image) of the speaker or hearer wants, this is called a *Face Threatening Act* (**FTA**). These FTA’s threaten the speaker’s or hearer’s sense of autonomy or self esteem.

Brown and Levinson discuss three main strategies that influence the linguistic choices that the speaker makes. These choices represent the social and cultural conventions that are relevant for the people involved in the conversation. These strategies are: the *On-record*, the *Off-record* and the *Don’t do FTA* strategy. These will be explained next. Examples of each of these strategies are displayed in table 4.1. The strategies are sorted on increasing politeness as displayed in the table.

4.3.2 On-record strategy

The on-record strategy is used when the speaker’s meanings are unambiguous. This strategy splits into two sub-strategies: *bald* and *redressive*.

Bald on-record

Bald on-record strategy is the most clear in meaning of the two. Doing the FTA ‘baldly’ implies that the FTA is phrased in the most direct and blunt terms without accounting for face threat. A common form of bald on-record strategy is applying the imperative form: (“leave me alone!”). The bald strategy is used in emergency situations (“help me!”), when the threat to face is very small when something is stated (“the car is in the parking lot”) and in case the speaker is superior to the hearer (“did you finish your job yet?”). In short: the speaker is sincere, does not say more than required, is relevant and avoids any ambiguity and obscurity.

Redressive on-record

In the redressive on-record strategy a speaker performs an unambiguous speech act while using redressive language to tone down its force. Redressive language is the usage of language to moderate the force of the FTA.

The on-record strategy with redressive action is split up in **positive politeness** and **negative politeness**. Positive politeness is oriented at *approval* while negative politeness is oriented at *autonomy*.

A common example of positive politeness is using first names or in-group names when addressing the hearer to make the hearer believe that he is part of the group (“hey pal, what’s the time?”) or to flatter the hearer (“that’s a real nice job you did!”). These and other positive politeness tactics are displayed in table 4.2.

Tactic	Example
1. Notice hearer's admirable qualities or possessions, show interest, exaggerate.	"Hey love your new Palm-pilot, can I borrow it sometime?"
2. Employ phonological slurring to convey in-group membership.	"Heya, gimme a hand with this willya?"
3. Use colloquialisms or slang to convey in-group membership.	"Most are damn hard, but this one should be a piece-of-cake.
4. Use ellipsis (omission) to communicate tacit understandings.	(Do you) "Mind if I join you?"
5. Use first name or in-group name to insinuate familiarity.	"Hey Bud, have you gotta minute?"
6. Claim common view: assert knowledge of hearer's wants or that hearer has knowledge of speaker's wants.	"You know how the janitors don't like it when..."
7. Seek agreement; raise or presuppose common ground/ common values; engage in small talk/ joke.	"How bout that game last night? Did the Ravens whip the pants off the Giants or what!"
8. Give reasons: assert reflexivity by making activity seem reasonable to the hearer.	"I'm really late for an important appointment, so ..."
9. Use inclusive forms (we or lets) to include both speaker and hearer in the activity.	"We're not feeling well, are we?"
10. Assert reciprocal exchange or tit for tat.	"Do me this favor, and I'll make it up to you."
11. Give something desired - gifts, sympathy, understanding.	"You look like you've had a rough week."

Table 4.2: Tactics of Positive Politeness (from [MO03]), adapted from [BL87])

Negative politeness is used when the speaker threatens the hearer's autonomy, when for instance limiting their freedom of action. *Hedging* is one example of a negative politeness tactic. A hedge is described as 'a particle, word or phrase that modifies the degree of membership of a predicate or noun phrase in a set; it says of that membership that it is partial, or true only in certain respects, or that it is more true and complete than perhaps might be expected' [BL87]. An example is the word "just" in "I just wondered if you could give me that pen". Another negative tactic is *conventional indirectness* ("can you show me the way?"). This and other negative tactics are summed up in table 4.3.

Tactic	Example
1. Be conventionally indirect; inquire into the hearer's ability or willingness to comply.	"Can you tell me what time it is?"
2. Use hedges: words or phrases that diminish the force of a speech act.	"Can I perhaps/possibly trouble you?"
3. Use subjunctive to express pessimism about hearer's ability/willingness to comply.	"Could I ask you a question"?
4. Use words or phrases that minimize the imposition.	"I need just a little of your time."
5. Give deference by using honorifics: Sir, Mr., Ms., Dr.	"Can I help you, Sir."
6. Use formal word choices to indicate seriousness and to establish social distance.	"Could you tolerate a slight imposition on my part?"
7. Apologize: admit the impingement, express reluctance.	"I am sorry to bother you, but..."
8. Impersonalize the speaker and hearer by avoiding the pronouns "I" and "you."	"Is it possible to request a favor?"
9. Use the past tense to create distance in time.	"I had been wondering if I could ask a favor."
10. Nominalize (change verbs and adverbs into adjectives or nouns) to diminish speakers active participation.	"My asking you to leave is required by regulations."
11. State the FTA as a general rule.	"Regulations require that I ask you to leave."

Table 4.3: Tactics of Negative Politeness (from [MO03]), adapted from [BL87])

4.3.3 Off-record strategy

The off-record strategy is the indirect way of executing the FTA. The FTA is phrased in such a way that more than one interpretation is possible, and one of these interpretations does not pose a threat to the hearer's face. Because the FTA is phrased ambiguously, the hearer has no way of telling with certainty which of the interpretations the speaker intends, because the speaker can always claim an alternate interpretation. For instance, when someone says: "This weather always makes me thirsty" whilst visiting the hearer's home, the speaker implies that he would like a drink. In this case, the speaker can always claim the literal, non face-threatening informing interpretation instead of the indirect requesting interpretation.

4.3.4 Don't do FTA strategy

This strategy causes the speaker to omit to perform the FTA entirely. This way, the threat to the hearer is avoided completely.

4.3.5 Politeness in Dutch

This section describes particularly how politeness manifests itself in the Dutch language as was researched by Roel Vismans [Vis94a] [Vis94b].

Basic Sentences

Vismans researched the force of basic illocutions [Vis94b], chapter 8.4. In order to get an indication of what effect basic illocutions have on the overall face of the directive speech act, native speakers' reactions were measured. 24 subjects were asked to rate the politeness of utterances which were mutations of the same directive: "de deur dicht doen" (close the door). The utterances were judged (in random order) on the following six-point scale:

1. = too polite
2. = very polite
3. = polite
4. = neither polite nor impolite
5. = impolite
6. = very impolite

The context of the utterances was defined as being an informal setting with strangers. The sentences and their politeness scores attributed to them by the test subjects are displayed

Form	Utterance	Politeness score
INT	Zou je de deur dicht willen doen?	3.03
	Zou je de deur dicht kunnen doen?	3.05
	Kun je de deur dichtdoen?	3.17
	Wil je de deur dichtdoen?	3.35
DECL	Je zou de deur dicht moeten doen.	4.09
	Je moet de deur dichtdoen.	4.75
IMP	De deur dichtdoen.	5.00
	Doe de deur dicht.	5.03
	Deur dicht.	5.86

Table 4.4: Politeness ratings for basic illocutions (from [Vis94b], page 209)

in table 4.4. The results shows clearly that basic illocutions can be arranged on a scale of politeness with IMP as the most impolite and INT as the most polite. DECL is closer in politeness to IMP than to INT. This is especially the case for the conditional “zou” which has a mitigating effect.

Modal Particles

The politeness strategies described in previous sections also apply to the Dutch language, but the Dutch language uses modal particles more extensively to express politeness than English. Modal particles’ only function is that of reflecting the mood or attitude of the speaker or narrator. They do not have any mood attributed to themselves but only change the mood of the *verb*. An example is the modal particle ‘just’ in “I just want you to leave”. The usage of modal particles for politeness in Dutch is described by Vismans [Vis94a] and [Vis94b].

Modal particles are used especially in *directive sentences* in negative politeness strategies. Directive sentences try to get the hearer to perform an action, usually in the benefit of the speaker. One of these strategies is *hedging* (see previous section). Vismans’ research concentrated on the modal particles shown in table 4.5. The table shows the distribution of modal particles over different sentence types: declarative, interrogative and imperative sentences. Table 4.6 shows the ordering of the modal particles in different sentence types. He also separates the modal particles in *reinforcers* and *mitigators* (see table 4.7). Reinforcers apply more pressure to the hearer of the speech act, while mitigators do the opposite, having the same effect as *hedges* (see section 4.3).

Vismans conducted an experiment ([Vis94b], chapter 8) to determine the effect of reinforcers and mitigators on the force of FTA's. In this experiment, 111 test subjects judged the politeness of the same basic directive as described in section 4.3.5, where the only variable was the modal particle used (including none). The test sentences were again imperative, declarative and interrogative mutations of the same basic directive. This basic directive and some examples of test sentences are shown below (with [MP] a modal particle):

- de deur dicht doen (the basic directive)
- doe de deur [MP] dicht
- Je moet de deur [MP] dicht doen
- Kun je de deur [MP] dicht doen?

All possible pairings of sentences (52 in total: 10 pairs of declaratives and 21 pairs of imperatives and interrogatives) within each illocution were used, after which the test subjects were asked to indicate which sentence of a pair (in random order) was the most forceful.

This experiment had the following results:

1. The force of an FTA + a reinforcer is stronger than that of a FTA + a mitigator
2. The force of an FTA without mitigator is stronger than the force of an FTA with a mitigator
3. The force of an FTA + a reinforcer is stronger than the force of an FTA without a reinforcer.
4. The stronger the force of the basic FTA is, the weaker effect a reinforcer has
5. The weaker the force of the basic FTA is, the weaker effect a mitigator has

In short: $FTA + mitigator < FTA < FTA + reinforcer$

Particle	DECL	INT	IMP
dan	-	-	+
eens	+	+	+
even	+	+	+
maar	+	-	+
misschien	-	+	-
nou	-	+	+
ook	+	+	-
soms	-	+	-
toch	-	-	+

Table 4.5: Distribution of MPs in sentence types; DECL = declarative, INT = interrogative, IMP = imperative (from [Vis94a])

Type	Cluster order
DECL	ook, maar, eens, even
INT	nou, misschien/soms*, ook, eens, even
IMP	dan/nou*, toch, maar, eens, even

Table 4.6: Dutch particles order (from [Vis94a]). Particles marked with * are interchangeable.

Status	MP
Reinforcers	dan
	eens
	nou
	ook
	toch
Mitigators	even
	maar
	misschien
	soms

Table 4.7: Status of MPs (from [Vis94a])

T-V distinction

Besides particles, Vismans also points out the use of formal or informal form of address. Besides using official titles, Dutch also uses *T-V distinction* in personal pronouns (“je” or “u”). The name T-V distinction stems from the T-form (“tu”) and V-form (“vos”) used in

Latin, for informal and formal addressing respectively. See table 4.8 for more examples in Dutch.

Informal	Formal
u	je/jij
uw	je/jouw
dank je	dank u
alsjeblieft	alstublieft

Table 4.8: Formal and informal phrases

However, Johnson et al. [JMAR05] found that applying T-V distinction in German did not have any noticeable effect on the notion of politeness. This is why T-V distinction is not placed under politeness in this project.

Diminutives

Vismans also discusses diminutives (“huisje” (little house)) in politeness in Dutch.

For instance in the Dutch phrases:

“Ik wil een pils” (I want a beer)

“Ik wil een pilsje” (I want a little beer)

the diminutive form is considered more polite, and is used in the tactic of minimizing the imposition (see table 4.3). An interesting example is the use of the Dutch modal particle “even” (a little moment) which when in a directive in its diminutive form is even more polite:

“kun je me even helpen?” (could you help me for a moment?)

“kun je me eventjes helpen?” (could you help me for a little moment?)

4.4 Formality of Language

Everyone seems to have an idea about the difference between formal and informal use of words. Formality in this sense is set apart from for instance T-V distinction (see section 4.3.5), but covers the formality of individual words and phrases. An example of formal language might be the description of a law. A good example of informal use of language might be between friends in a relaxed conversation. It is difficult to make a clear and general definition of formality, and because of this the amount of literature available is limited.

4.4.1 Surface Formality and Deep Formality

Heylighen and Dewaele ([HD99]) distinguish two forms of formality: **surface formality** and **deep formality**. Surface formality can be characterized by attention to form for the sake of convention of form itself (using “sir”), while deep formality is attention to form for the sake of unambiguous understanding of the precise meaning of the expression (such as a law description).

Heylighen and Dewaele state that in general, a formal style is more detached, accurate, rigid and heavy while an informal style is more flexible, direct, implicit and involved, but conveys less information.

An empirical measure for deep formality is the formality score, or *F-score*, proposed by Heylighen and Dewaele [HD99] and [FJM98]. The F-score is a formula based on the frequency of different word classes in a text. Nouns, adjective, articles and prepositions are found more often in formal styles, while pronouns, adverbs, verbs and interjections are more frequent in informal styles. This corresponds with the notion that informal language is more *context dependent*. The word classes that were mentioned to imply informal style represent words such as “here”, “therefore”, “later” etcetera, and interjections such as “hi!” and “oh”, which are relevant in the context of the situation, also called *deictic* words. The other word class group associated with formal language, or *non-deictic* words, corresponds to logic predicates. These do not vary under normal circumstances, for example “tree”, “woman” or “red”.

Deep formality is described by Heylighen and Dewaele as theoretically more fundamental, because any attempts to determine surface formality are more ad-hoc, intrinsically limited and dependent on the specific language and culture. This is not the case for deep formality.

4.4.2 Surface Formality Model

Heylighen and Dewaele [HD99] further propose a formality model which describes the effect on language formality of factors such as speaker gender (males tend to be more formal), audience size (the larger, the more formal), background (familiarity of the speaker with the audience decreases formality), setting (when sender and receiver are in the same setting, formality decreases), time span (distance in time between sending and receiving, the more formal), feedback (lack of feedback increases formality), need for understanding (the greater

the need for understanding, the more formal), introversion (more introvert is more informal) and academic level (the higher the education, the more formal).

4.4.3 Fuzziness

Heylighen and Dewaele [HD99] use the term *fuzziness* to describe the situation where the reference of an expression is not precisely (unambiguously) determined, even when the complete context is known. For instance, “it is hot” could be the most precise description of a temperature. But what exact temperature does this description imply? 25 degrees, 40 degrees or somewhere in between? Fuzzy expressions such as this are context dependant, and are as such considered more informal.

This project will implement surface formality instead of the more complex deep formality, because the latter would take considerable time to properly implement, which was not available.

4.5 Related work

This section discusses related literature on variation through politeness and natural language alignment.

4.5.1 Variation through Politeness

Related work on the implementation of *the politeness theory* of Brown and Levinson [BL87] (see section 4.3 for a description) in some way for variation of language will be discussed here.

An important contribution in the research on linguistic style variation with politeness was done by Walker et al. [WCW96]. This paper introduces linguistic style improvisation for lifelike computer characters. Speech acts have abstract representations from which computer characters can improvise. They use face, described in Brown and Levinson’s politeness theory, to calculate the *threat* of the FTA to the hearer by the following formula:

Threat = the social distance between the speaker and the hearer + the power the hearer has over the speaker + the imposition of the speech act.

They ranked the imposition by how the speech act types relate to the basic desires of agents by looking at how they threaten their autonomy. An accept speech act has the lowest threat, followed by an inform, followed by a rejection and the request-act speech act posing the highest threat to autonomy. The calculated threat varies from 0 to 150, and is used to choose one of four FTA strategies. Direct strategies correspond with threat values up to 50, approval-oriented strategies to values up to 80, autonomy-oriented strategies for values up to 120 and off record strategies for values between 120-150.

Gupta et al. [GRW05] describe politeness and variation in a teacher-tutor system for learning English which also uses the politeness theory of Brown and Levinson [BL87], based on the system by Walker et al. [WCW96]. They discuss the importance to have variation in interaction to increase the believability of embodied conversational agents. The system they propose uses a language generation architecture which implements a content planner which sends a script with the sequence of speech acts and the basic components of the utterance to the language generator. This generator uses a sentence planner to create multiple variations with different politeness strategies. After this, the surface realizer generates strings as output and finally assigns appropriate prosody.

Gupta et al. [GWR07] use the politeness theory of Brown and Levinson [BL87] with POLLY, an English language learning system. POLLY’s language generation system is comparable to that in [GRW05]. They introduce **softeners**, which can be characterized as words or phrases that reduce the force associated with an FTA, comparable to mitigating modal particles or hedging (see 4.3.2). Gupta et al. conclude after test runs of their system with different user groups that POLLY’s output is generally consistent with the politeness theory. For instance, utterances to strangers need to be more polite than to friends. Their indirect strategies are seen as the rudest, while these are supposed to be the most polite according to the politeness

theory. This is caused by limitations in their system and limited number of indirect strategies their system supports. They also conclude that cultural differences exist in perception of politeness. For instance, Indians generally rated sentences higher on politeness than British test subjects, especially between friends, because Indians are generally more informal in their communication.

Another paper on linguistic variation based on politeness and notion of face is written by Porayska-Pomsta and Mellish [PPM04]. They present a simple language generation model for choosing appropriate surface realisations of tutoring responses based on the politeness notion of face, also basing their own system on that put forward by Walker et al.[WCW96]. They adapt the definition of face to fit the needs of the educational field to generate varied corrective feedback to students. A similar tutor-coaching system is proposed by Johnson et al. [JRB⁺04].

Morand and Ocker [MO03] suggest that the politeness theory can play an important role in computer-mediated communication. They discuss how politeness can be “reliably observed and quantitatively measured at a linguistic level of analysis”. They proceed with a description of the different politeness strategies, specifically for positive and negative politeness tactics (see section 4.3). Morand and Ocker end the paper with a study into the relation between politeness and various variables in computer-mediated communication.

Ardissono et al. [ABL99] use an approach to provide a formal framework on conventional indirect speech acts (see section 4.3), and to recognize underlying beliefs of these speech acts. This resulted in a plan-based representation of speech acts using politeness features. In future work they hope to evaluate the politeness based on these features.

André et al. [ARMB04] focus on the generation of stylistic variations in order to improve the user’s perception of the interaction. They also use the politeness theory and model used by Walker et al.[WCW96] with some of the same politeness strategies and tactics. However, they also incorporate a cognitive theory of emotions in their generation of language by including factors such as joy and sadness. These emotions influence the different factors that make up the threat of an FTA. They also state that the politeness scale proposed by Walker et al. [WCW96] is insufficient in that it does not account for how they came to the values between 0 and 150, and propose to construct a more detailed hierarchy which includes the politeness sub-strategies. They also debate whether the social distance between the speaker and the hearer and the power the hearer has over the speaker should be regarded as variables at all: the power ultimately resides with the user being able to shut the system down. Also, they state that the attribution of social distance to a computer system might be far-fetched. But they conclude the user actually does attribute social behaviours to the system, and the power the user has over the system by being able to switch it off fades in a dialogue where user and system assume a co-operative role in for instance a tutoring system.

4.5.2 Alignment in Natural Language

As with formality of language, little literature exists on alignment of language using natural language generation.

Brockmann et al.[BIOW05] argue that for a co-operative, satisfying interaction it is always better to align language than not to do so. They state affective alignment creates a sense of success for the dialogue. They researched the effect of altering personality variables (neuroticism and extraversion) to adjust alignment behaviour, and thus simulated how different personalities align in a greater or lesser extent. A strong aligner would align their model to that of the hearer, while a weaker aligner would stick to their own language model. They successfully realised language alignment by aligning the word order between agents in a conversation, using word sequences.

Isard et al. [IAB⁺06] propose CRAG, which generates aligned dialogues between two agents, based on their individual personalities. This is done by an over-generating and ranking method with language models using both personality and alignment. CRAG uses word sequence probability in the language alignment process, and can generate dialogues in which one agent takes a point (for instance, “I really liked that movie”), after which the other (depending on the personality setting) aligns his language to that of the other agent. They state that they plan to expand the personality-corpus resources on which personalities are based on to realise more visible variation.

4.6 Alignment Model

This section will look at the architecture and setup of the implementation of the alignment model in the Virtual Guide.

4.6.1 The Virtual Guide's Behaviour

The Virtual Guide's 'job' is to assist visitors in navigation (see also 2.1.2). Taking the saying 'the customer is always right' as her mission, the Guide should always try to be as polite as possible. 'As polite as possible' refers here to the simulation of a person who will try to be polite, but is influenced (as a human might be) by the attitude that exists towards the user.

When the user is impolite, the Guide will respond to this by being less polite. When the user is polite, the Guide will do the same. She will not, however, become more impolite or more polite than the user. Being too polite can be interpreted as overexaggerating and rude ([Vis94a]). Her behaviour does not alter the way she selects dialogue acts, but does change the linguistic form of her expressions. Besides politeness, the formality of language of her expressions is also aligned to that of the user. This alignment process will be described in more detail in the following sections.

As said, the goal of aligning is that it will introduce a more realistic variation of utterances from the Guide and thus increasing believability (see section 4.1 and [GRW05]) and increase satisfaction and co-operation. Example dialogues can be found in section 4.10.

4.6.2 The Alignment State

This section will describe the **alignment state** of the Guide. The variables that make up the alignment state are:

- Formality of language
- Politeness

These sub-states will be discussed now.

Formality of Language

The formality of language plays a role in social distance. In order to decrease this distance, the Guide will align the formality of its language to that of the user. The goal is to generate a sense of co-operation and satisfaction. The formality value of the Guide represents the average of formal or informal language usage at the surface (see section 4.4). A surface formality model is far less complex to realize than a deep formality language model, which

would not be feasible to implement due to time constraints. Formality of language also is not the main focus of this thesis.

In this project, formality refers to the individual formality values attributed to words or phrases and their synonyms in a formality lexicon. The higher the formality value, the more formal the language, and the lower the formality the more informal the language usage becomes.

These values are used to analyse user utterances. For instance, a user may greet the guide with “hi!”. This greeting is considered as a greeting that is used in informal occasions. In order to properly align, the Guide should respond with an equally informal greeting. Alignment is accomplished by selecting specific formal or informal versions of words and phrases and inserting them into the system utterance. Many verbs and nouns have synonyms that vary in formality as well. Consider the more informal “plee” (“the john”) to the more formal “toilet” (“toilet”), and informal “aanwijzen” (“to point at”) and formal “markeren” (“to mark”) respectively. The analysis of utterances of users for formality of language on which the Guide’s alignment state is based, will be discussed in section 4.7.1.

The formality variable has effect on the choice of isolated words or groups of words in generated system replies. This makes the process simple but at the same time gives room for a moderate amount of language variation.

The following example shows the insertion of variants of words based on the formality value. The positions in the template sentence where the words or phrases are inserted are between brackets.

Formality value set to informal:

“Ik heb [de wc] [gemarkeerd] op [het kaartje]”

(I [pointed out] [the bathroom] on [the little map])

Formality value set to formal:

“Ik heb [het toilet] [gemarkeerd] op [het plan]”

(I [marked] [the toilet] on [the diagram])

The formality value varies between -5, for highly informal and +5, for highly formal. 0 is considered neutral. The default setting is 0. This process of generating language with varying formality will be discussed in more detail in section 4.8.4.

Politeness

The politeness variable of the alignment state can be visualised as the impression the Virtual Guide gets of the user by analysing the politeness in the user's language. This analysis is based on politeness indicators discussed in section 4.3. It is a basis for the choices that are made in language generation of the Guide. It simulates how the Guide would respond to the user with the current politeness state, and thus gives an indication of how polite the Guide will react to future user input. The intention of aligning politeness is that the Guide will react in a natural way by aligning the amount of politeness put into its utterances to that of the user's utterances. When the user is rude, the Guide will also be rude (simulating it does not find it necessary to be polite to the user when the user is not polite himself), and vice versa for when the user is polite. See section 4.6.1 also.

As said, the Guide looks for politeness indicators in the user's input, as described in section 4.3 and especially 4.3.5. More specifics on these indicators will be discussed in section 4.7.2. The politeness value determines the strategy the Guide will take in expressing the speech act. These strategies and their tactics were discussed in section 4.3, the selection process is explained in section 4.8.

The politeness state varies between the boundaries of 5 and -5. 5 is considered the most polite, -5 is the least polite. 0 is neutral and is the initial value at the start of a dialogue, unless it has been set to another value (see 4.9).

4.6.3 System Description

An overview of the alignment model architecture is displayed in figure 4.1.

The main class of the alignment architecture is the agent class **Alignment State Manager**. This newly created agent runs parallel with the system agents of the original implementation, which were discussed in chapter 2.

The alignment state manager maintains and updates the alignment state of the Guide. The alignment state manager is based on the **Dialogue Manager**. It too can send and receive messages to and from the **Facilitator** (see section 2.2.2). Communication between the alignment state manager and the dialogue manager as seen in figure 4.1 happens via a facilitator.

Whenever a user performs one or more dialogue acts in a turn, and the turn is returned to the system, the alignment state manager will receive a user dialogue act list message from the facilitator. Next, the alignment state manager analyses the text of the utterance embedded in the dialogue acts of this list for formality and politeness information (see next sections). After the analysis and updating the alignment state accordingly, the alignment state manager sends a *state ready* message to the facilitator. This message informs that the analysis and updating of the alignment state is complete.

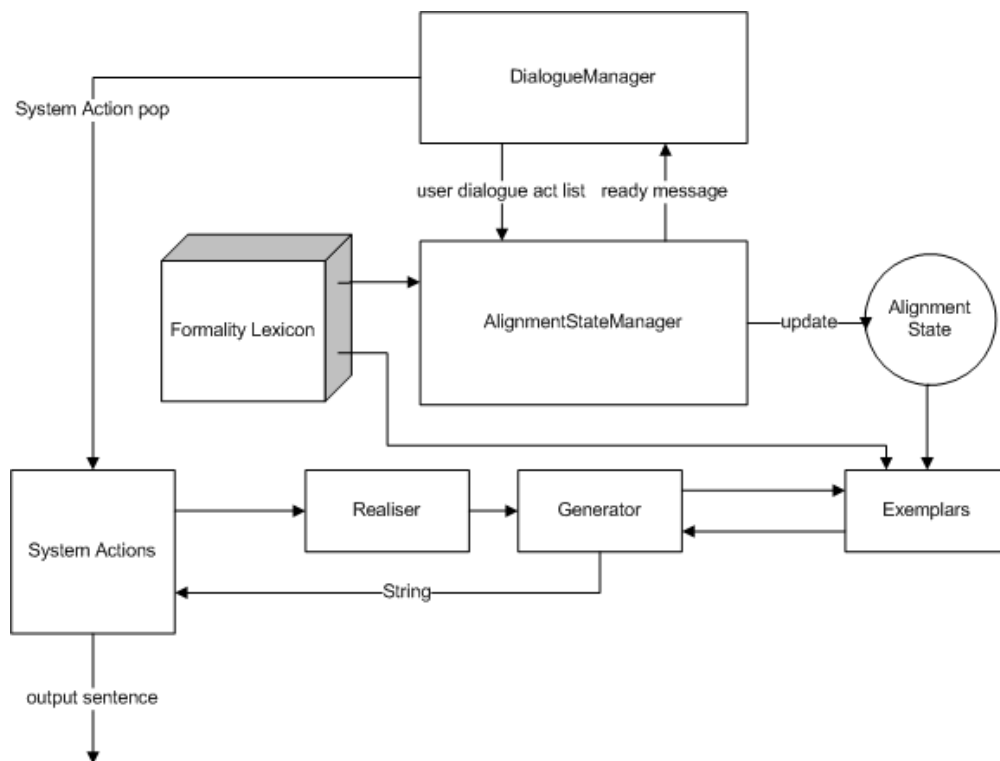


Figure 4.1: The architecture

The dialogue manager is subscribed to this message type, and will wait for a state ready message from the facilitator before executing any system actions that are on the action stack. This way, the alignment state will always be updated before a system reply is generated. This prevents the Guide from generating sentences based on an old state.

The alignment manager then pops the actions from the action stack. As the action classes are executed, they each call the **Realiser** which in turn calls the **Generator** to generate a system utterance for that action using **Exemplars** (see section 2.2.3). Based on the action arguments and the current state, the exemplars will generate an aligned sentence for the system reply (see section 4.8).

The different stages of the overall process are grouped in input analysis and language generation and are shown in the following enumeration:

- **Input Analysis** -

1. User enters utterance
2. User utterance is analysed for politeness and formality information
3. Alignment state is updated based on analysis results

4. Alignment State Manager sends **state ready** message
- **Language Generation** -
5. 'state ready' message received by Dialogue Manager
6. System actions are executed and sentence generation called
7. Politeness Strategy is selected based on politeness variable
8. Politeness tactic is chosen resulting in a *template sentence*
9. Words are inserted into template sentence based on formality variable
10. Sentence receives final edits
11. Sentence is presented to the user

These stages will be described in the following sections.

4.7 User Input Analysis

The Guide is set up such in a way that the user must always initiate the dialogue. This gives the guide the advantage to scan this first input to get information about what style it should use. A proposal to extract emotional information from text is that of Guinn and Hubal [GH03]. They propose to apply emotional or attitudinal tags to grammar rules. When the user uses words like “please” in his or her utterance, this adds to the politeness of the sentence. In the end, the average score of a sentence is calculated by adding all the individual scores of the utterance and normalizing them. This idea has been adjusted for this project for both politeness and formality analysis.

The input will be analysed for 2 types of information:

1. Formality

- Formality keywords
- T-V distinction

2. Politeness indicators

- Sentence form
- Modal particles and softeners

The formality and politeness analysis processes will be discussed in the next sections.

4.7.1 Formality Analysis

A *shaded lexicon* is used to calculate the surface formality scores of each utterance as proposed by Fleischman and Hovy [FH02]. A shaded lexicon contains words and phrases with have a *shade* attributed to them. This shade represents the formality value of the word or phrase. This shaded lexicon is used to calculate the average formality of a user utterance by looking up the shade of keywords in it as proposed by Guinn and Hubal [GH03]. This process will be described in this section.

Shaded lexicon

The shaded lexicon is based on the original lexicon (see section 2.4.1), but has been created separately. This way, changes to the lexicon will not interfere with the original implementation. This does mean, however, that new lexical entries must be added to both the original and shaded lexicon.

An example of a lexical entry in the shaded lexicon is shown in figure 4.2.

```
\w VERB MARK
\n gemarkeerd
\s 1
\n aangegeven
\s -1
\n aangestipt
\s -3
```

Figure 4.2: A shaded lexicon entry

A lexical entry is denoted by `\w`. In this example, verbs expressing the concept ‘mark’ (action) with varying shades of formality are identified with `\n` and their formality shade by `\s`. The shades in the lexicon vary between -5 (very informal) to 5 (very formal). It must be noted that due to lack of proper data of formality in Dutch words, shades of the lexical entries are not only based on the work of Heyligen et al. [HD99] but also on information found in a dictionary (Van Dale On-line [Taa07]) and the writer’s intuition.

For instance, when ranking the interjections “hoi” (hi) and “goedendag” (good day), “hoi” will be considered informal because this greeting is used between friends, and as such more context dependent. Furthermore, Van Dale [Taa07] also indicates this last greeting as informal. Other greetings such as “yo” are even more informal because these are usually only used in an ingroup. The opposite goes for “goedendag”, which is more formal because it is less ambiguous [HD99]. Another example is the choice between “aanstippen” (to check off with a dot) and “markeren” (to mark). The first is more involved and metaphorical because the marking of the object on the map is described as actually drawing a dot on the map

with a pen, resulting in an informal score. it can also be seen described as more fuzzy (see section 4.4.3) and subsequently informal. The second verb is more detached (abstract) and accurate and receives a more formal shade.

Calculating Formality State

When a dialogue act list is received by the alignment state manager (see section 4.6), the individual dialogue acts are retrieved from that list. From these dialogue acts, the parsed text input is processed word by word. Each word is looked up in the lexicon and its formality shade (if available) is returned. After this, the individual scores of the words are summed up and the average formality score for the sentence is calculated by dividing the sum of the shades by the number of shaded words.

$$UtteranceFormality = (\sum_1^k wordShade(i))/k$$

The new formality state is calculated with the following formula:

$$Formality(i + 1) = alpha * State(i) + (1 - alpha) * UtteranceFormality(i + 1)$$

With *UtteranceFormality* the average formality of the last utterance. *alpha* varies between 0 and 1 and determines how changeable the politeness state is. The closer *alpha* comes to 1, the less effect new sentences have on the original preset politeness state. *alpha* can be set in the control panel (see section 4.9).

T-V Distinction

T-V distinction is placed under formality instead of politeness, because research suggested T-V distinction in German did not really have an effect on the notion of politeness (Johnson et al. [JMAR05]). Because the German language and culture can in many ways be considered similar to the Dutch, this conclusion has adopted for this project.

The detection of T-V distinction is realised by scanning for words and phrases, such as “je” and “u” the same way as in the previous section. When formal versions of these phrases are detected, the T-V distinction value is set to 1, otherwise to 0. The default state is 0.

The T-V-distinction state is separated from the formality variable in the system, because it is difficult to predict the effect of it on overall formality of an utterance. If T-V distinction would be implemented based solely on the alignment variables, the Guide might start using formal or informal T-V distinction at irregular or inappropriate times. For instance, when for a certain informal alignment state the user is addressed informally, this might change to formal when the formality variable increases. Then it might change back to informal again, without the user ever changing his or her T-V distinction at all. To guarantee consistency, our approach only changes the default informal T-V distinction state to formal *after* when the user does so him/herself first. After this, the T-V distinction state may alter again

depending on the user's input. The mentioned irregularity in language would also manifest itself if T-V distinction would be used as a politeness tactic (see section 4.3.5).

The realisation of T-V distinction in generated language is described in section 4.8.5.

4.7.2 Politeness Analysis

Because the amount of utterances the system supports is limited, the analysis can be limited to the supported sentences (see appendix C).

Specific data on politeness in Dutch was available from the works of Roel Vismans (see [Vis94b]) and [Vis94a] (see section 4.3.5). This data is used as a basis for extracting *politeness indicators* from user input. The politeness indicators are separated in *sentence form*, *modal particles* and other *softeners*. Because no data exist to base an analysis on for interjections such as greetings, these are treated differently. This will be discussed shortly.

Sentence form

Vismans' research (see section 4.3) is used to analyse sentence structures for politeness in the Virtual Guide. Vismans' findings indicate that interrogative form is the most polite, followed by declarative as neither polite nor impolite and imperatives as least polite. This concurs with the politeness theory (section 4.3).

The scale used in this test is adjusted to the scale used in this project, -5 to 5. Vismans notes that in his experiment, scale 1 was almost never chosen, and suggests it be eliminated from the scale.

On Vismans' scale, 4 would refer to neither polite nor impolite. This would transfer to 0 on our scale. 2 on Vismans' scale (very polite) would transfer to 5 on our scale, and 6 to -5. See table 4.9 for the approximate results.

Form	Example	Vismans' score	Adjusted scale
INT	Zou je [de deur dicht] willen [doen]?	3.03	3
	Zou je [de deur dicht] kunnen [doen]?	3.05	3
	Kun je [de deur dichtdoen]?	3.17	3
	Wil je [de deur dichtdoen]?	3.35	2
DECL	Je zou [de deur dicht] moeten [doen].	4.09	0
	Je moet [de deur dichtdoen].	4.75	-2
IMP	[De deur dichtdoen].	5.00	-3
	Doe [de deur dicht].	5.03	-3
	[Deur dicht].	5.86	-5

Table 4.9: Sentence structures and approximate alignment of scales

These values are connected to user input and used to calculate the new system state. To give an indication on how our scale fits the experiment of Vismans, table 4.10 has been filled out with variations of the illocution “toon me de zaal” (show me the hall). Variations of the examples of Vismans received the same score, for other forms a politeness score has been selected by the writer. This was done by comparing the sentences with Vismans’ examples and scoring the sentences accordingly. For instance, based on the forceful verb “moeten” (must) in “je moet...” (you must...) or the mitigating verb “zouden” (could) in “zou je me kunnen vertellen...” (could you show me tell me...). Declarations that do not address the Guide directly such as “ik zoek de zaal” (I’m looking for the hall) have been ranked as well. Because these requests are indirect, they do not have a low score as the imperatives. The forceful phrase “ik wil...” (I want...) or “ik moet...” (I must...) have been ranked lower than the neutral “ik zoek de zaal”. The inability to apply the lowest imperative sentence structure in table 4.9 resulted in the lowest score in table 4.10 being -3. The score of user utterances can be lower however when modal particles are used. Other dialogue act types are included in the analysis as well. Opening and closing greetings receive a politeness score of +2, while a thank dialogue act receives +3, being even more polite.

Modal Particles

The sentence structure is not the only influence on sentence politeness. Mitigating and reinforcing modal particles have an effect too. In a second experiment, Vismans looked into the force of modal particles in Dutch ([Vis94b], chapter 8.2). A list of modal particles and their distribution, order and distinction reinforcers and mitigators was already discussed in section 4.3.5.

Another politeness indicator is the presence of **softeners** as described in [GWR07] (see section 4.5.1). Dutch softeners that are supported are “misschien” (perhaps) and “graag” (gladly). Examples of these softeners applied in sentences are:

“weet u misschien waar de zaal is?” (do you perhaps know where the hall is?)

“ik wil graag naar de zaal” (I would like to go to the hall)

The effect of modal particles and softeners will be discussed next.

Calculating Politeness State

As stated in 4.3.5, the force of an FTA is altered by mitigating or reinforcing particles. This is implemented in this system for all the modal particles mentioned in table 4.7 as follows: the lower the politeness attributed to the FTA, the less effect a reinforcer has, and the higher the politeness attributed to the sentence structure, the less effect a mitigator has (see section 4.3.5).

Therefore, the following formula is used to calculate the effect of a modal particle or softener

Form	Example	Politeness score
INT	Waar is de zaal?	0
	Waar vind ik de zaal?	0
	Waar kan ik de zaal vinden?	1
	Wat is de weg?	0
	Hoe vind ik de zaal?	0
	Hoe kan ik de zaal vinden?	1
	Hoe kom ik daar/hier/bij de zaal?	0
	Hoe kan ik daar/hier/bij de zaal komen?	1
	Hoe moet ik daar/hier/bij de zaal komen?	-1
	Zou je de zaal willen tonen?	3
	Zou je de zaal kunnen tonen?	3
	Kun je de zaal tonen?	3
	Wil je de zaal tonen?	2
	Kun je me vertellen waar de zaal is?	3
	Kun je het toilet aanwijzen/tonen/markeren/aangeven?	3
	Weet je waar de zaal is?	3
DECL	Je zou de zaal moeten tonen.	0
	Je moet de zaal tonen.	-2
	Ik zoek de zaal.	0
	Ik ben verdwaald.	0
	Ik moet naar de wc.	-1
	Ik wil naar de wc.	-1
	Je moet me vertellen/zeggen waar de zaal is.	-2
	Je moet me vertellen/zeggen hoe ik bij de zaal kan komen.	-2
IMP	Toon (me) de zaal.	-3
	Breng me naar de wc.	-3

Table 4.10: Sentence structures and their politeness score

on the overall politeness of an utterance:

$$T = (5 - |P|) / 5 * E$$

E is the effect of the mitigator, reinforcer or softener and T is the total effect of the mitigator, reinforcer or softener that will be used to calculate the new state. Softeners and mitigators have politeness +1, while enforcers have politeness -1. P is the politeness attributed to the sentence form (see 4.7.2). Using this formula, the effect of mitigators and reinforcing particles will be reduced on the extreme sides of the politeness scale.

The politeness of an utterance is calculated as follows:

$$UtterancePoliteness = SentenceFormPoliteness + T$$

Finally, the politeness variable of the alignment state is calculated as follows:

$$Politeness(i+1) = alpha * Politeness(i) + (1 - alpha) * UtterancePoliteness(i+1)$$

Where $politeness(i)$ is the politeness variable in state (i), $alpha$ the alignment variable as discussed in section 4.7.1 and $UtterancePoliteness$ the calculated politeness of the user utterance. See section 4.10 for examples of this calculation.

4.8 Language Generation

This section describes the language generation process, which consist of the following stages:

1. System actions are executed and sentence generation called
2. Politeness strategy is selected based on politeness variable
3. Politeness tactic is chosen resulting in a *template sentence*
4. Words are inserted into template sentence based on formality variable
5. Sentence receives final edits
6. Sentence is presented to the user

4.8.1 Sentence Planning

When a system action is popped (see section 2.3.5), the realiser class is called which sends the action type (see section 2.6) to the generator, together with the current alignment state. The generator calls the exemplars and based on the action that was popped, a response is selected. For this system response, a politeness tactic is selected depending on the politeness value of the alignment state. This process will be discussed in section 4.8.3. This results in a template sentence. The *template sentence* is a basic sentence structure with sockets where words of varying formality are to be inserted. These words are selected based on the formality value of the alignment state. See section 4.8.4 for a description of this process.

4.8.2 Exemplars

The choice for using Exemplars was based on previous satisfactory usage an the availability of the package for the generation of route descriptions [BK04]. See section 2.2.3 for an introduction into exemplars.

The original route generating exemplars differ from those in the alignment model in that the route generating sentences are constructed solely with exemplars and based on random variation, while the generated sentences in the alignment model use external functions to apply linguistic variations and are based on a changing alignment state. It must be noted that our model does not alter the way route descriptions are generated.

The new exemplars were built on the routing generation exemplars. Because words of varying formality were retrieved from a lexicon which was realised via java methods (see section 4.8.4) which already had a random selection process, the random functionality of the exemplars was not used. Also, java methods were found more compact than writing new exemplars for each utterance, which is the general approach. This is why a single exemplar was set

up to hold all the variations of template sentences for one utterance type. These tactics are selected using java methods.

The exemplars support the following utterance types:

- ActionGreet
- ActionClose
- ActionThank
- ActionNegative
- ActionPositive
- ActionTellLocation
- ActionTellLocationSelf
- ActionTellLocationMultiple
- ActionTellLocationNotHere
- ActionDescribeObject
- ActionDescribeObjectWhichOne
- ActionUninterpretable
- ActionGoTo
- ActionTellPathTo
- ActionTellPathFromTo
- ActionTellPathEndUnclear
- ActionNoMatch

When a system action is popped, the utterance type is sent from the action classes to the exemplars as an argument, together with the formality and politeness variable. These utterance types match the system actions (see section 2.6). Some actions have multiple utterance types, such as *ActionTellLocation*. Depending on the action argument, a different utterance is required (see section 2.6.2. For example, *ActionTellLocation* tells the user the location of an object, *ActionTellLocationSelf* tells the user his or her location, *ActionTellLocationMultiple* informs the user there are multiple instances of the same object and *ActionTellLocationNotHere* informs the user the object was not found. Each utterance type

is connected to one or more exemplars that each generate a sentence. Most utterances only have a single sentence, but for instance the *ActionUninterpretable* uses three exemplars: first of all an sentence which informs the user that he or she was not understood, followed by a sentence informing the user that he or she used words that are not supported by the Guide and finally a request to try again. These 3 sentences are all placed together in the final system utterance.

As said, for each of the sentences, an exemplar was constructed with all variations of template sentences. This list is then chosen from by evaluating the politeness variable of the alignment state together with a random factor, which are both realised with java objects. This process will be described in the next sections.

4.8.3 Politeness Tactic Selection

The politeness tactic selection process is based on the scale proposed in [WCW96], where the main politeness strategies, a based on Brown and Levinson ([BL87]) are sorted by increasing politeness as follows:

1. **Direct** (bald on-record)
2. **Approval** (redressive on-record: positive politeness tactics)
3. **Autonomy** (redressive on-record: negative politeness tactics)
4. **Indirect** (off-record)

See also table 4.1. The approval strategy uses positive politeness tactics, while the autonomy strategy uses negative politeness tactics.

These strategies have been adapted to fit the politeness scale of this project (see section 4.6.2) as follows (where the politeness value corresponds with the politeness value of the alignment state):

1. A politeness value of -5 to 4 maps to on-record strategies, where:
 - (a) A politeness value of -5 to -2 maps to the direct strategy.
 - (b) A politeness value of -2 to 4 maps to on-record with redressive action strategies, where:
 - i. A politeness value of -2 to 1 maps to approval strategy
 - ii. A politeness value of 1 to 4 maps to autonomy strategy
2. A politeness value of 4 to 5 maps to the indirect strategy

Brown and Levinson ([BL87]) only distinguish the mentioned main politeness strategies, and do not further rank the individual politeness tactics. Because obvious differences exist in the politeness attributed to the individual tactics of a strategy, these strategies have been refined into multiple tactic clusters. A list of individual strategies and tactics with examples and corresponding politeness ranges for requests is given in table 4.11. Some tactics can also be applied to inform actions, which can be found in table 4.12. The politeness values of politeness tactics of these tables only show completed sentences, with neutral formality (0).

The tactics are based on the positive and negative tactics of Brown and Levinson ([BL87]), but also inspired by Gupta et al. [GRW05] and Vismans [Vis94b]. No proper indirect strategies were used for inform acts, because the strategies suggested there by the politeness theory (hinting, being vague, joking and tautologies) are not implemented in this system, but are considered as future work. As said, the approval groups harbor the positive politeness tactics, while the autonomy groups consists of negative politeness tactics. Other examples of these tactics can be found in section 4.3.

Based on Vismans' data (see table 4.9), the direct strategy was separated into an imperative (1) and a declarative (2). Other tactics mentioned in table 4.9 have been placed the tactic groups that correspond with these scores as well. Because no data besides Vismans' research was available, the remaining tactics in the strategies were sorted on intuition.

Because many politeness tactics could not be properly applied to inform acts, it was not necessary to create tactic clusters as with the requests. Also, because informs do not need to be as polite as requests, the indirect tactic could be omitted altogether with the politeness scale adjusted accordingly. In order to increase variation, some system dialogue acts that originally had informs associated with them, have been transformed into requests. For instance, the utterance “Ik heb de zaal gemarkeerd op het kaartje” (I marked the hall on the map) have become “kijk op het kaartje waar ik de zaal heb aangegeven” (look at the map where I marked the hall). This increased the number of available politeness tactics, improving variation.

Strategy	Inform forms	Tactic
DIRECT	dat is de zaal	
APPROVAL	dat is de wc, vriend	ingroup
AUTONOMY	ik heb je niet helemaal verstaan	minimize imposition
	het lijkt erop dat ik je niet heb verstaan	seems
INDIRECT	-	-

Table 4.12: Politeness strategies and tactics for informs.

Politeness Tactic	Sentence	Politeness
DIRECT		
1) Imperative	kijk op het kaartje waar ik de zaal heb aangegeven	-4 to -2.5
2) Declarative	je moet het anders zeggen	-2.5 to -2
APPROVAL 1		-2 to -0.5
3) Ellipsis	nog een keer proberen?	
4) Inclusive	zullen we het nog een keer proberen?	
5) Ability	is het mogelijk dat je op het kaartje kijkt waar ik de zaal heb gemarkeerd?	
6) Ingroup name	probeer het nog eens, vriend	
APPROVAL 2		-0.5 to 1
7) Optimism	je vindt het vast niet erg om het nog een keer te proberen	
8) Give reason	als je het anders formuleert, lukt het vast wel	
9) Mind	als je het niet erg vindt, kun je op het kaartje kijken waar ik de zaal heb gemarkeerd	
10) Declarative 1	je zou het nog een keer moeten proberen	
AUTONOMY 1		1 to 2.25
11) Nominalize	de vraag is of je het anders wilt formuleren	
12) Impersonalize	is het mogelijk dat het nog een keer geprobeerd kan worden?	
13) Distance in time	ik vroeg me af of je op het kaartje wilt kijken waar ik de zaal heb gemarkeerd	
14) Conventionally indirect	wil je het nog een keer proberen?	
AUTONOMY 2		2.25 to 3.2
15) Subjunctive pessimism 1	zou je op het kaartje willen kijken waar ik de zaal heb gemarkeerd?	
16) Subjunctive pessimism 2	zou je het anders kunnen formuleren?	
17) Conventionally indirect 2	kun op het kaartje kijken waar ik de zaal heb gemarkeerd?	
AUTONOMY 3		3.25 to 4
18) Hedging	kun je het misschien anders formuleren?	
19) Minimize imposition	kun je eventjes op het kaartje kijken waar ik de zaal heb gemarkeerd?	
20) Apologize	sorry, maar kun je het nog eens proberen?	
INDIRECT		4 to 5
21) Indirect	iemand zou het nog een keer moeten proberen	

Table 4.11: Politeness strategies and tactics for requests.

4.8.4 Variation in Formality

As said, variation in formality is accomplished by inserting words with varying formality which most closely match the formality value of the alignment state into the template sentences that were selected in the previous sections.

A proposal that inspired this part of the project is put forward by Michael Fleischman and Eduard Hovy [FH02]. They propose to generate multiple lexical variations, after which one of these alternatives is chosen based on the current system state (in this case the speaker's attitude) and the total score (shade) of the variation. The distance between the score of the sentence and the current system state is calculated, and the generated sentence with shortest distance is selected as utterance. For instance witness may describe the scene of an accident in two ways: "the red car hit the blue car" or "the red car crashed into the blue car". The verbs "hit" and "crash" display a different attitude toward the red car, where the latter is more negative. In the end, the verb that most closely matched the preset system state is selected.

This proposal has been adjusted for retrieving words with different formality shades from a shaded lexicon (see section 4.7.1). This is the same shaded lexicon used for formality analysis.

In the template sentence in the exemplars, sockets are present where words are inserted. (Simplified) example:

```
ik heb de [getWord("NOUN_ROUTE")] naar {object} [getWord("VERB_IS_MARKED")] op
[getWord("NOUN_MAP")]
```

This template sentence represents a variation of the sentence "ik heb de route naar {object} gemarkeerd op het kaartje" (I've marked the route to {object} here on the map), with sockets for formality variations shown between brackets and {object} representing the object in the VMC the user was referring to. The function `getWord` retrieves the best fitting formality variation. In this example the variations for 'NOUN_ROUTE', 'VERB_IS_MARKED' and 'NOUN_MAP' are found in the shaded lexicon and inserted into the sentence template to form a complete sentence.

The insertion of words of varying formality is based both on the current formality variable of the alignment state as well as a random factor. The function for selecting words of varying formality selects a word from a pool of synonyms that exists for each value range. These ranges are: 5 to 3 for very formal, 3 to 1 for formal, 1 to -1 for neutral, -1 to -3 for informal and -3 to -5 for very informal. The formality ranges are used to make up for the lack of valid and exact data on the formality of these words (see section 4.7.1).

An example of formality pools for the different formality ranges is given in table 4.13. The current formality state of the Guide determines the pool. From this pool a word is randomly selected and inserted into the socket of a template sentence. In this case, a greeting.

Formality range	NOUN_ROUTE	NOUN_MAP	VERB_IS_MARK
5-3	-	het plan	-
3-1	-	-	gemarkeerd
1-1	route, weg	de plattegrond, de kaart	aangegeven
-1-3	-	het plattegrondje	aangestipt
-3-5	-	het kaartje	-

Table 4.13: Word pools for different formality value ranges

For instance, high formality would result in:

ik heb de route naar {object} gemarkeerd op het plan

while low formality might result in:

ik heb de weg naar {object} aangestipt op het kaartje

4.8.5 Sentence Revisions

After the generation of sentences is complete, post edits are applied to them using exemplars. T-V distinction is applied by replacing personal pronouns (such as “je” and “u”), selecting the properly matching verb and other words that contain personal pronouns (such as “alsjeblieft”) according to the current alignment state. See table 4.8 for a complete list. These words are tagged in the structure selection process, after which they can be replaced when these tags are encountered.

Another edit is adding the system state at the end of each system utterance for it to be printed in the console. This is for debugging purposes, similar to the way the type of dialogue act is printed after each dialogue act. Finally, a dot is added to each ending of a sentence that did not already have a punctuation mark. An example run of the edits (except adding the system state) is given below.

Tagged structure:

`<persp>kunt</persp> <persp>u</persp> het nog eens proberen?`

Formal edit:

Kunt u het nog eens proberen?

Informal edit:

Kun je het nog eens proberen?

4.9 Virtual Guide User Interface

A tab was added to the Virtual Guide GUI to include the parameters for the alignment model (see figure 4.3).

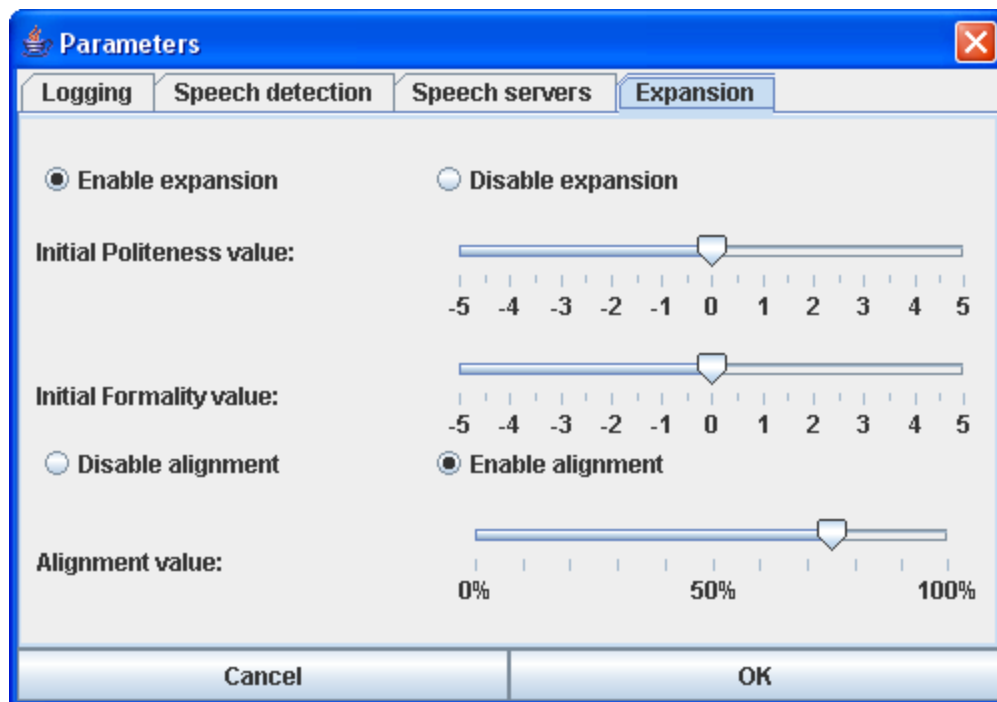


Figure 4.3: The alignment parameters

The alignment parameters tab allows the user to disable or enable the alignment model described in this chapter. The formality and politeness state sliders can be adjusted to set the starting values which are used when the Virtual Guide is initialized. Furthermore, the user may choose to either enable language alignment with the user defined formality and politeness values as starting point or to disable alignment and keep these defined values unchanged (static) during the dialogue. Finally, a value of alpha (see section 4.7.1) can be set between 0 and 100% which influences how fast the Guide will align its language. These parameters allow the user to experiment with different settings, as well as to return to the original pre-alignment model situation by disabling the alignment model.

4.10 Example

The system starts out in a neutral state with the system alignment state being:

- Politeness state = 0
- Formality state = 0
- T-V distinction = informal
- alpha is set to 0.75

The politeness and formality variables are neutral at this stage, but will be directly influenced by the first user utterance. The user says:

”Hoi, weet je misschien waar de zaal is?”

after which the system starts analysing the utterance.

Input Analysis

The Guide is able to extract the following information from this user utterance:

- The dialogue act *greeting* (“*hoi*”) has politeness +2
- The sentence structure “*weet je*” has politeness +3 (see table 4.10)
- The user uses a softener: “*misschien*” with politeness +1
- “*je*” is an informal personal pronoun, indicating informal T-V usage
- “*hoi*” has formality shade -1

Calculating the new alignment state

The new *formality variable* is calculated using the formula in section 4.7.1:

$$UtteranceFormality = -1$$

$$Formality = 0,25 * 0 + 0.75 * -1 = -0.75$$

The new *politeness variable* is calculated using the formula of section 4.7.2:

$$UtterancePoliteness = 3 + ((5 - |3|)/5) * (1) = 3,40$$

$$Politeness = 0,25 * 0 + 0.75 * 3,40 = 2,55$$

T-V distinction setting will remain on informal.

Generating a System Reply

First of all, a reply to the Open action is generated. This reply is entirely based on the formality state. A random greeting is selected from the pool with formality ranging from -1 to 1: “hoi”.

Secondly, a proper answer must be given to the WH-question. A reply should be generated based on the politeness state of 2,55. This maps to the negative politeness tactics of the autonomy 2 group (see table 4.11). A random negative politeness tactic of this group is selected, in this case subjunctive pessimism 1. The selected sentence t could be:

Zou <persp>u</persp> op ["NOUN_MAP"] willen kijken waar ik {obj} heb
[VERB_IS_MARKED]?

The capital words between brackets indicate sockets for words with varying formality. Based on the formality state, words are selected and inserted into the sentence structure, the same way as happened for the greeting. {obj} denotes the object of which the location was asked. After inserting the words of varying formality, the sentence looks like:

Zou <persp>u</persp> op het plattegrondje willen kijken waar ik {obj} heb
aangegeven?

Printing the System Reply

Finally, after applying sentence edits for T-V distinction, the reply of the Guide will become:

“Hoi.”

“Zou je op het plattegrondje willen kijken waar ik obj heb aangegeven?”

4.11 Discussion

Hayes-Roth [HRBS95] proposes 5 lifelike qualities that are required of improvisation characters:

1. *Interesting variability* of the interpretation of the user input on different occasions
2. *Random variability* in the way the agent responds on different occasions (such as a greeting)
3. *Idiosyncrasies* in the behaviour of different characters
4. *Plausible motivations* for agent behaviour
5. *Recognizable emotions* associated with agent behaviour and interactions

Although all user utterances are interpreted in the same way each time, our alignment model does approach the first quality by reacting differently on user input for different variable values of politeness, formality and alignment. The second quality is reached by the random factor that is used in the choice of politeness tactic and the same for words which have the same formality. By applying alignment based on the politeness theory, it can be argued that idiosyncrasies in a way are realised as well. With different setting for alpha, the Guide will align differently to the user, simulating for instance a strong or a weak aligning personality. Also, for different preset formality and politeness variables, the Guide can start off a dialogue with the user with any combination of politeness and formality of language, and with or without alignment as well. Plausible motivations, for choice of words at least, are simulated by aligning the choice of politeness tactics of the Guide to the politeness of the user. This, in turn, allows the Guide to display emotions on a certain level by being polite or impolite.

Walker et al. [WCW96] also discuss the lack of distinction in imposition in requests. For instance, asking to borrow a car differs greatly from asking to borrow some sugar.

Vismans [Vis94a] criticizes Brown and Levinson's theory for being too much oriented on individualistic western cultures and neglecting the more community centered cultures of the Far East.

André et al. discuss some of the problems of using Brown and Levinson's theory as a basis for politeness strategies [ARMB04]. Firstly, the ordering of politeness strategies from direct over approval-oriented and autonomy-oriented to off-record may lead to inconsistencies. The context of the dialogue may influence the politeness that is experienced for a strategy, such as the ability of the hearer to perform a task. They state that that autonomy-oriented strategies are not always seen as more polite than approval-oriented strategies. Secondly, there is no clear indication of how combined strategies within one utterance should be handled. Another

issue is the usability of the application of vagueness and indirectness for they might be seen as impolite instead of most polite. After this, they propose to improve the strategy selection process by including not only the user's emotional state, but also the personal profile and a classification of events, actions and object that caused it. For instance, the system does not keep record of the number of attempts a user has made to ask something. The user might become frustrated, causing some politeness tactics to become offensive instead of polite (by for instance saying "please try again").

In order to look into the issues discussed by André et al. and other possible problems of the application of the politeness theory, the alignment model will be evaluated in the next chapter.

CHAPTER 5

Evaluation of the Alignment Model

5.1 Introduction

This chapter discusses a limited evaluation of the alignment model for the Virtual Guide as described in the preceding chapter. The alignment model involves multiple system variables: politeness, language formality and alignment. This makes it unpractical to properly evaluate these values independently. A *questionnaire* has been filled out by test subjects to evaluate of the validity of the selection of *politeness* tactics, the *formality* values and perception of *alignment*.

5.2 Questionnaire

A questionnaire allowed for a relatively easy and independent evaluation of sentences and dialogues with different formality, politeness and alignment settings by allowing test subjects to judge sentences and dialogues that were generated with different preset system variables.

5.2.1 Goals

The questionnaire had several goals, each placed under a separate section:

1. Let the test subjects evaluate sentences that were generated with different formality settings to get an indication of the validity of the used formality values (see section 4.8.4)
2. Let the test subjects evaluate sentences that represent all the different politeness tactics to get an indication of the validity of the politeness values and value ranges that were attributed to the tactics and tactic groups in the alignment model (see section 4.8.3)
3. Let the test subjects evaluate dialogues that were generated with different alignment settings to get an indication of whether alignment is successfully implemented and whether different alignment values are noticable (see sections 4.7.1 and 4.7.2)
4. Let the test subjects pick their preferred version of a dialogue and let them express their opinion on the dialogues to get an indication of what alignment value is preferred, together with general feedback on the dialogues

5.2.2 Setup

The questionnaire was published on-line using WinSurvey 2005 [Ace05]. Test subjects of both sexes were invited via e-mail and a forum (<http://www.nerden.nl/ugmobstar/>) to enter into the experiment, after which 25 respondents of both sexes participated. Their age varied between 18 and 55 years, with 26 years as average. The participants filled in the questionnaire in their internet browser, so no special software was needed to participate. The questionnaire was set up in such a way that each page displayed one question. This was done to discourage comparing sentences and should make the judgements more independent of each other. After each question, there was room for the participant to add his or her comment.

The questionnaire can be found in appendix D.

At the start of each of the three sections of the questionnaire, the participant received new instructions. The following sections discusses the three parts of the questionnaire: formality, politeness and alignment evaluation.

5.2.3 Formality Values

To evaluate the effect of different values for language formality, the participants were asked to judge 20 sentences generated with different formality settings. Four sentences were generated for each of the following formality values: -5, -2.5, 0, 2.5 and 5 (see section 4.8.4). A list of the 20 sentences can be found in table 5.1. The ordering of these sentences was randomized for more reliable results.

Sentence	Formality Variable
1) Je hebt waarschijnlijk woorden gezegd die ik niet snap.	-5
2) Als je het nog eens probeert, lukt het vast wel.	-5
3) Hey. (begroeting)	-5
4) Als je het niet erg vind, kun je op het plattegrondje kijken waar ik de zaal heb aangestipt.	-5
5) Er zijn woorden gezegd die ik niet snap.	-2.5
6) Geen probleem.	-2.5
7) Doe.	-2.5
8) Het aanstippen van de weg van het toilet naar de zaal is klaar.	-2.5
9) Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven.	0
10) Je zult het anders moeten zeggen .	0
11) Goedenacht.	0
12) Ik heb je niet helemaal begrepen.	0
13) Je zult het op een andere wijze moeten verwoorden.	2.5
14) Tot je dienst.	2.5
15) Je zou op de plattegrond moeten kijken waar ik de zaal heb gemarkeerd.	2.5
16) Je locatie is gemarkeerd door het kruis.	2.5
17) Adieu.	5
18) Je zou op het plan moeten kijken waar ik de zaal heb gemarkeerd.	5
19) Je zult het op een andere wijze moeten verbaliseren.	5
20) Het indiceren van de route naar toilet is gereed.	5

Table 5.1: 20 sentences with their formality setting that were used in the questionnaire

3: Je zult het *anders* moeten zeggen.

☐ Zeer formeel
☐ Formeel
☐ Niet formeel en niet informeel
☐ Informeel
☐ Zeer informeel

Commentaar:

Figure 5.1: A formality question from the questionnaire

The test subjects were asked to judge the formality by marking their perceived formality for a sentence on a Rikert scale. The scale maps to the formality values as follows (see section 4.7.1):

- *very formal*: -5 to -3
- *formal*: -3 to -1
- *neither formal nor informal (neutral)*: -1 to 1
- *informal*: 1 to 3
- *very informal*: 3 to 5

An example of a formality question from the questionnaire with the *Likert-style scale* on which the user may indicate his or her perceived formality can be seen in figure 5.1. After this section, the questionnaire continued on the subject of politeness tactics.

5.2.4 Politeness Tactics

In order to retrieve an indication of the validity of the politeness values attributed to all of the politeness tactics implemented in the system (see 4.8), the overall politeness of generated sentences has been rated by test subjects. This took place in a fashion similar to the rating of formality. Instead, a Likert scale from -5 (very impolite) to +5 (very polite) was used. A similar test was conducted by Gupta et al. [GWR07].



4: je zou het nog een keer moeten proberen

☐ Zeer beleefd
☐ Beleefd
☐ Niet beleefd en niet onbeleefd
☐ Onbeleefd
☐ Zeer onbeleefd

Commentaar:

Figure 5.2: A politeness question from the questionnaire

The politeness scale for this test is defined by the following politeness values (see section 4.7.2).

- *very impolite*: -5 to -3
- *impolite*: -3 to -1
- *neither polite nor impolite (neutral)*: -1 to 1
- *polite*: 1 to 3
- *very polite*: 3 to 5

Subjects judged 21 example request acts which each represented one of the politeness tactics that are used in the implementation. Of course, to properly evaluate these tactics, multiple forms per politeness tactic should have been used, but this would make the questionnaire too long and would discourage the test subjects. These tactics and the sentences are listed in table 5.2, together with their tactic groups and their corresponding politeness ranges used in the tactic selection process (see section 4.8.3). See figure 5.2 for an example of a questionnaire question for politeness.

5.2.5 Alignment Values

The most complex of the 3 questionnaire sections was the evaluation of the of alignment. This setup of this part of the evaluation is based on the evaluation of the alignment model by Isard et al. [IAB⁺06], which done by generating dialogues with different personality settings that would be judged by test subjects.

Politeness Tactic	Sentence	Politeness
DIRECT		
1) Imperative	kijk op het kaartje waar ik de zaal heb aangegeven	-4 to -2.5
2) Declarative	je moet het anders zeggen	-2.5 to -2
APPROVAL 1		-2 to -0.5
3) Ellipsis	nog een keer proberen?	
4) Inclusive	zullen we het nog een keer proberen?	
5) Ability	is het mogelijk dat je op het kaartje kijkt waar ik de zaal heb gemarkeerd?	
6) Ingroup name	probeer het nog eens, vriend	
APPROVAL 2		-0.5 to 1
7) Optimism	je vindt het vast niet erg om het nog een keer te proberen	
8) Give reason	als je het anders formuleert, lukt het vast wel	
9) Mind	als je het niet erg vindt, kun je op het kaartje kijken waar ik de zaal heb gemarkeerd	
10) Declarative 1	je zou het nog een keer moeten proberen	
AUTONOMY 1		1 to 2.25
11) Nominalize	de vraag is of je het anders wilt formuleren	
12) Impersonalize	is het mogelijk dat het nog een keer geprobeerd kan worden?	
13) Distance in time	ik vroeg me af of je op het kaartje wilt kijken waar ik de zaal heb gemarkeerd	
14) Conventionally indirect	wil je het nog een keer proberen?	
AUTONOMY 2		2.25 to 3.2
15) Subjunctive pessimism 1	zou je op het kaartje willen kijken waar ik de zaal heb gemarkeerd?	
16) Subjunctive pessimism 2	zou je het anders kunnen formuleren?	
17) Conventionally indirect 2	kun op het kaartje kijken waar ik de zaal heb gemarkeerd?	
AUTONOMY 3		3.25 to 4
18) Hedging	kun je het misschien anders formuleren?	
19) Minimize imposition	kun je eventjes op het kaartje kijken waar ik de zaal heb gemarkeerd?	
20) Apologize	sorry, maar kun je het nog eens proberen?	
INDIRECT		4 to 5
21) Indirect	iemand zou het nog een keer moeten proberen	

Table 5.2: Politeness tactics in the questionnaire

A variation of this setup is used in this project to get an indication of whether the implementation of alignment is perceived as explicit enough by test subjects. Three different versions of 3 dialogues of a user with the Virtual Guide are written out with constant user input and different alignment settings, forming a total collection of 9 different dialogues. To keep the questionnaire accessible, the number of dialogues to be rated by test subjects had to be limited. Therefore, only 3 different user profiles were used: in dialogue 1, the user is informal and impolite, in dialogue 2 the user is formal and polite and in dialogue 3 the user is formal and polite. This selection was made to create an interesting variation of dialogues but is nevertheless arbitrary. Each of the 3 versions show one of these three dialogues with alignment values *alpha* (see section 4.7.1 and 4.7.2) set to either 0% (no alignment), 100% (immediate alignment) and the intermediate 50%. This resulted in 9 dialogues which were judged independently in a random order.

The test subjects were be asked to score their perception of alignment on the following Rikert-like scale:

1. The Guide does not align
2. The Guide aligns a little
3. The Guide aligns moderately
4. The Guide aligns strongly

The dialogues can be found in table 5.3, 5.4 and 5.5. The 9 questions for the alignment of individual dialogues were followed up by 3 final questions in which the test subjects were presented with all the versions of each of the mentioned dialogues put together in a random order, resulting in three lists of dialogues where alignment was the only variable. These lists of versions of the dialogues are gathered in the tables mentioned above. The test subjects were asked to choose the version of the dialogue they preferred best. The test subjects were asked to explain their choice. The question was intentionally kept vague to provoke the user to express their thoughts on any aspect of the dialogues.

Dialogue 1	Alpha
BEZOEKER: Hi GIDS: Goeienavond. BEZOEKER: Toon me de zaal GIDS: Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven. BEZOEKER: Doe GIDS: Dag.	0%
BEZOEKER: Hi GIDS: Hallo daar. BEZOEKER: Toon me de zaal GIDS: Is het mogelijk dat je op het plattegrondje kijkt waar ik de zaal heb aangestipt? BEZOEKER: Doe GIDS: Tot een andere keer.	50%
BEZOEKER: Hi GIDS: Hi. BEZOEKER: Toon me de zaal GIDS: Kijk op het kaartje waar ik de zaal heb aangestipt. BEZOEKER: Doe GIDS: Doe.	100%

Table 5.3: Dialogue 1 with different alignment settings (alpha)

Dialogue 2	Alpha
BEZOEKER: Goedendag GIDS: Goeienavond. BEZOEKER: Weet u misschien waar het toilet is GIDS: Als je het niet erg vind, kun je op de kaart kijken waar ik het toilet heb aangegeven. BEZOEKER: Dank u wel GIDS: Graag gedaan.	0%
BEZOEKER: Goedendag GIDS: Goeienavond. BEZOEKER: Weet u misschien waar het toilet is GIDS: Zou u op de kaart willen kijken waar ik het toilet heb aangegeven? BEZOEKER: Dank u wel GIDS: Het was me een genoegen.	50%
BEZOEKER: Goedendag GIDS: Goedendag. BEZOEKER: Weet u misschien waar het toilet is GIDS: Kunt u misschien op het kaartje kijken waar ik de zaal heb gemarkeerd? BEZOEKER: Dank u wel GIDS: Het was me een genoegen.	100%

Table 5.4: Dialogue 2 with different alignment settings (alpha)

Dialogue 3	Alpha
BEZOEKER: Hey GIDS: Goeiemiddag. BEZOEKER: Ik wil graag naar de zaal GIDS: Je vind het vast niet erg om op de kaart te kijken waar ik de zaal heb aangegeven. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Tot ziens.	0%
BEZOEKER: Hey GIDS: Hallo daar. BEZOEKER: Ik wil graag naar de zaal GIDS: Is het mogelijk dat er op het plattegrondje gekeken wordt waar ik de zaal heb aangestipt. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Doeg.	50%
BEZOEKER: Hey GIDS: Hey daar. BEZOEKER: Ik wil graag naar de zaal GIDS: Ik vroeg me af of je op het kaartje kunt kijken waar ik de zaal heb aangestipt. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Later.	100%

Table 5.5: Dialogue 3 with different alignment settings (alpha)

5.3 Results and discussion

5.3.1 Formality Values

Sentence	Preset Formality Value	Results
1) Je hebt waarschijnlijk woorden <i>gezegd</i> (0) die ik niet <i>snap</i> (-1).	-5	-0.8
2) Als je het nog eens <i>probeert</i> (0), lukt het vast wel.	-5	-0.9
3) <i>Hey</i> (-4). (begroeting)	-5	-3.3
4) Als je het niet erg vind, kun je op het <i>plattegrondje</i> (-1) kijken waar ik de zaal heb <i>aangestipt</i> (-3).	-5	-0.9
5) Er zijn woorden <i>gezegd</i> (0) die ik niet <i>snap</i> (-2).	-2.5	0.2
6) <i>Geen probleem</i> (-2).	-2.5	-0.2
7) <i>Doei</i> (-2).	-2.5	-2.7
8) Het <i>aanstippen</i> (-2) van de <i>weg</i> (-1) van het toilet naar de zaal is <i>klaar</i> (-4).	-2.5	-0.1
9) Je zou op de kaart moeten kijken waar ik de zaal heb <i>aangegeven</i> (0).	0	-1.1
10) Je zult het anders moeten <i>zeggen</i> (0).	0	-1.3
11) <i>Goedenacht</i> (1).	0	2.0
12) Ik heb je niet <i>helemaal</i> (0) <i>begrepen</i> (0).	0	0.5
13) Je zult het <i>op een andere wijze</i> (2) moeten <i>verwoorden</i> (2).	2.5	1.5
14) Tot je <i>dienst</i> (2).	2.5	1.1
15) Je zou op de <i>plattegrond</i> (1) moeten kijken waar ik de zaal heb <i>gemarkeerd</i> (1).	2.5	-0.1
16) Je <i>locatie</i> (2) is <i>gemarkeerd</i> (1) door het <i>kruis</i> .	2.5	0.7
17) <i>Adieu</i> (2).	5	-1.5
18) Je zou op het <i>plan</i> (5) moeten kijken waar ik de zaal heb <i>gemarkeerd</i> (1).	5	0.1
19) Je zult het <i>op een andere wijze</i> (2) moeten <i>verbaliseren</i> (4).	5	2.9
20) Het <i>indiceren</i> (4) van de <i>route</i> (0) naar toilet is <i>gereed</i> .	5	3.2

Table 5.6: questionnaire results for formality

The first thing that stands out in the formality questionnaire results in table 5.6 is that for a great deal of utterances, the set formality value do not match the results. This is most likely caused by the lack of variations of words with extreme formality values, such as -5. However, the individual formality shades of words that were inserted in the template sentence (see section 4.8.4) are displayed between brackets in table 5.6 show better results.

The first four sentences with formality setting of -5 did actually all receive a negative formality value, and the results all lie within a distance less than 1 from the average sentence formality of the words that were inserted.

For the next setting of -2.5, sentence 6 was seen as neutral instead of informal, while the results of sentence 7 closely match the lexicon value. Sentence 8 was judged neutral as well, while it was intended as informal.

For the setting of 0, sentence 9 and 10 were unexpectedly considered less formal than intended. Judging by the test subjects' comments, this is probably caused by test subjects finding form of question forceful and impolite, influencing their score.

For the setting of 2.5, only sentence 15 deviated from the intended value. This is most likely caused by its low politeness as well.

For the setting of 5, the closing greeting of sentence 17("adieu") was judged very informal instead of very formal. Sentence 18 also did not meet the expected results, and has also been judged lower than anticipated. This is probably caused by the usage of the extreme formal but obscure noun "plan" (map). 19 did receive the intended score, and the verb "verbaliseren" (to verbalize) was even commented upon as being too formal. The same can be said for "indiceren" (to indicate) in sentence 20.

Because the ratings were of complete sentences instead of singled out words, the surrounding words influenced the formality ratings. Also, from what could be concluded from subjects' comments is that a few subjects took other influences such as the politeness of the utterance into account when rating formality of language. Rating individual words might have better results, but only on a single word basis. In contrast, the rating of complete sentences with combinations of words of different formality, which the Guide would actually use, is a more realistic approach. For the limited amount of time available, a choice had to be made and this was for the latter.

5.3.2 Politeness Tactics

Politeness Tactic	Sentence	P	Result
DIRECT			
1) Imperative	kijk op het kaartje waar ik de zaal heb aangegeven	-4 to -2.5	-1.7
2) Declarative	je moet het anders zeggen	-2.5 to -2	-2.2
APPROVAL 1		-2 to -0.5	
3) Ellipsis	nog een keer proberen?		-2
4) Inclusive	zullen we het nog een keer proberen?		1.9
5) Ability	is het mogelijk dat je op het kaartje kijkt waar ik de zaal heb gemarkeerd?		2.6
6) Ingroup name	probeer het nog eens, vriend		-1.8
APPROVAL 2		-0.5 to 1	
7) Optimism	je vindt het vast niet erg om het nog een keer te proberen		-0.2
8) Give reason	als je het anders formuleert, lukt het vast wel		0.3
9) Mind	als je het niet erg vindt, kun je op het kaartje kijken waar ik de zaal heb gemarkeerd		1.9
10) Declerative 1	je zou het nog een keer moeten proberen		0.2

Table 5.7: questionnaire Results for politeness tactics, part 1

Politeness Tactic	Sentence	P	Result
AUTONOMY 1		1 to 2.25	
11) Nomalize	de vraag is of je het anders wilt formuleren		-0.6
12) Impersonalize	is het mogelijk dat het nog een keer geprobeerd kan worden?		2.6
13) Distance in time	ik vroeg me af of je op het kaartje wilt kijken waar ik de zaal heb gemarkeerd		1.9
14) Conventionally indirect	wil je het nog een keer proberen?		1.8
AUTONOMY 2		2.25 to 3.2	
15) Subjunctive pessimism 1	zou je op het kaartje willen kijken waar ik de zaal heb gemarkeerd?		2.5
16) Subjunctive pessimism 2	zou je het anders kunnen formuleren?		2.7
17) Conventionally indirect 2	kun op het kaartje kijken waar ik de zaal heb gemarkeerd?		1.6
AUTONOMY 3		3.25 to 4	
18) Hedging	kun je het misschien anders formuleren?		2.6
19) Minimize imposition	kun je eventjes op het kaartje kijken waar ik de zaal heb gemarkeerd?		1.2
20) Apologize	sorry, maar kun je het nog eens proberen?		3.2
INDIRECT		4 to 5	
21) Indirect	iemand zou het nog een keer moeten proberen		-0.2

Table 5.8: questionnaire Results for politeness tactics, part 2

See table 5.7 and table 5.8. In the **direct group**, the *imperative* tactic (1) was seen as more polite than the *declarative* (2), which goes against Vismans’ findings (see section 4.3.5). This is probably caused by the forceful verb “moeten” (must), which was mentioned in test subjects’ comments. Also, the sentence gives a good deal of information (“waar ik de zaal heb aangegeven” (where I marked the hall)) which probably accounts for the higher politeness result.

For **approval 1 group**, the *Ellipsis* (3) and *Ingroup name* (6) tactics both had the intended results, with the latter sometimes commented upon as being sarcastic (“friend”). The *Inclusive* tactic (4) was seen by test subjects as both helpful by some and, in contrast, patronizing by others. The tactic was perceived more polite than intended, which also goes for the *ability* tactic (5).

In **approval group 2**, only tactic 9 (*Mind*) did not fit into the group boundaries by being rated higher than 1. This tactic was also described by one person as “as you would talk to a child”. More comments on this tactic were given in the next section of the questionnaire (see section 5.3.3).

In **autonomy 1 group**, the perceived politeness of the *Nomalize* tactic (11) is rated lower than the system value, and that of the *Impersonalize* tactic (12) higher. Both tactics were sometimes described as “unnatural” but also as “polite”.

For **autonomy 2**, only tactic 17 (conventionally indirect) was rated too low for the group. The *Subjunctive pessimism* tactics of 15 and 16 both fit in the group and their politeness scores lie close to each other, which corresponds with Vismans’ data [Vis94b].

Tactics 18 (*Hedging*) and 19 (*Minimalizing imposition*) of **autonomy group 3** are judged lower. The effect of hedging and minimalizing imposition using “misschien” and “eventjes” respectively, probably has been overestimated. The *Apologize* tactic 20 was closer to the intended target. The failure to reach the extreme high politeness value probably means the upper bound of the autonomy group should be lowered.

The final group with the **Indirect tactic** is judged far too low, indicating problems with the subtle indirect tactics without proper context as which is also discussed by Gupta et al. [GRW05] (see also section 4.11).

Overall, the number of tactics crossing over to any of the other main strategies (direct, approval, autonomy) with a greater difference in politeness than 1 compared to the boundaries of the group they belong to is only 3: tactics 1, 5 and 11. This crossing over was predicted by André et al. [ARMB04] (see section 4.11).

The following adjustments were made (for cohesion, no tactics were moved to other main strategy groups):

- tactic 1) has been shortened
- tactic 4) and 5) have been moved to Approval 2
- tactic 9) has been removed for *ActionTellLocation*
- tactic 19) has been moved to autonomy 2

5.3.3 Alignment Values

dialogue 1	Alpha	Results
BEZOEKER: Hi GIDS: Goeienavond. BEZOEKER: Toon me de zaal GIDS: Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven. BEZOEKER: Doe GIDS: Dag.	0%	2.24
BEZOEKER: Hi GIDS: Hallo daar. BEZOEKER: Toon me de zaal GIDS: Is het mogelijk dat je op het plattegrondje kijkt waar ik de zaal heb aangestipt? BEZOEKER: Doe GIDS: Tot een andere keer.	50%	2.16
BEZOEKER: Hi GIDS: Hi. BEZOEKER: Toon me de zaal GIDS: Kijk op het kaartje waar ik de zaal heb aangestipt. BEZOEKER: Doe GIDS: Doe.	100%	3.64

Table 5.9: dialogue 1 with different alignment settings (alpha) and test results

dialogue 2	Alpha	Results
BEZOEKER: Goedendag GIDS: Goeienavond. BEZOEKER: Weet u misschien waar het toilet is GIDS: Als je het niet erg vind, kun je op de kaart kijken waar ik het toilet heb aangegeven. BEZOEKER: Dank u wel GIDS: Graag gedaan.	0%	3.16
BEZOEKER: Goedendag GIDS: Goeienavond. BEZOEKER: Weet u misschien waar het toilet is GIDS: Zou u op de kaart willen kijken waar ik het toilet heb aangegeven? BEZOEKER: Dank u wel GIDS: Het was me een genoegen.	50%	3.28
BEZOEKER: Goedendag GIDS: Goedendag. BEZOEKER: Weet u misschien waar het toilet is GIDS: Kunt u misschien op het kaartje kijken waar ik de zaal heb gemarkeerd? BEZOEKER: Dank u wel GIDS: Het was me een genoegen.	100%	3.32

Table 5.10: dialogue 2 with different alignment settings (alpha) and test results

dialogue 3	Alpha	Results
BEZOEKER: Hey GIDS: Goeiemiddag. BEZOEKER: Ik wil graag naar de zaal GIDS: Je vind het vast niet erg om op de kaart te kijken waar ik de zaal heb aangegeven. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Tot ziens.	0%	2.28
BEZOEKER: Hey GIDS: Hallo daar. BEZOEKER: Ik wil graag naar de zaal GIDS: Is het mogelijk dat er op het plattegrondje gekeken wordt waar ik de zaal heb aangestipt. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Doeg.	50%	2.6
BEZOEKER: Hey GIDS: Hey daar. BEZOEKER: Ik wil graag naar de zaal GIDS: Ik vroeg me af of je op het kaartje kunt kijken waar ik de zaal heb aangestipt. BEZOEKER: Bedankt GIDS: Geen dank. BEZOEKER: Later GIDS: Later.	100%	3.2

Table 5.11: dialogue 3 with different alignment settings (alpha) and test results

Dialogue 1

For the first dialogue (see table 5.9), the first two versions were received the *little* alignment score for alpha as 0% and 50% respectively, while the second version with 100% got *moderate* to *strong* alignment. The the 50% version is seen as aligning less than the 0% version. This is probably caused by the politeness tactic used in the second system utterance. The 0% version uses the *Declarative 1* tactic, while the second version uses the *Ability* tactic (see table 5.9). The Ability tactic is rated by the test subject as considerably higher in politeness than the system standard (2.6 where the system standard was set between -2.0 and -0.5). Taking this rating into account, the test subjects probably rated the second version of dialogue 1 as less aligning because the Guide responds to an imperative user question (“show me the hall”) of very low politeness with a reply of moderate politeness (the Ability tactic), where in the first version the Guide has a a reply of neutral politeness.

The last version with alpha = 100% is commentend upon as the Guide being as impolite as the user, but also somebody commented the Guide adjusted too strongly.

In judging which version of dialogue 1 was preferred, 3 persons voted for version 1, 12 for version 2 and 9 for version 3.

Version 1 (50%) was mainly chosen because the Guide stays polite, even when the user is not. Also, the variation in greeting reply was found appealing. Version 2 (0%) was preferred by persons who enjoyed the variation in word choice, where the Guide stays polite even when the user is not. It does show “the Guide thinks the user is being rude” in the second utterance of the Guide, as some test subjects described it. Test subjects who chose version 3 (100%) said the Guide best mirrors the user in for instance the greetings. Some subjects found this mirroring also happens in real-life situations. Ironically, the mirroring of greetings was what other subjects found annoying about this version. One person who voted for version 3 found the second replies of the Guide in versions 1 and 2 sound sarcastic.

Dialogue 2

The second dialogue has results (see table 5.10) that are close together (all three received a *moderately* alignment score) but do increment as intended. The closing greeting of “het was me een genoegen” is seen as “too much of a good thing” by several subjects in the versions with 50% and 100% respectively.

In judging which version of dialogue 2 was preferred, 8 persons voted for version 1, 11 for version 2 and 4 for version 3.

Voters for version 1 (0%) wrote they found the other 2 versions too polite. Version 2 (50%) was chosen for not echoing “misschien” (maybe), variation in greeting and just as being polite from both sides. Version 3 (100%) was again chosen for best mirroring the user’s utterances.

General remarks on this dialogue was that one subject did not like being answered with another question, and many subject found the “als je het niet erg vindt” (if you don’t mind) out of place, stating: ”why would I mind?”, indicating the absence of any threat to autonomy.

Dialogue 3

The third dialogue (see table 5.11 displays the same upgoing trend as the second, corresponding with the system design, but with more distance between the scores. The 0% version is perceived as little alignment, the second as *little to moderate* alignment and the third as *moderate* alignment.

In judging which version of dialogue 3 was preferred, 6 persons voted for version 3, 8 for version 2 and 9 for version 3.

Version 1 (0%) was mainly voted for because the Guide’s replies in the other versions were found unnatural or annoying, and for being polite, yet informal. Version 2 (50%) was chosen for being direct and “aligning the best”. Voters for version 3 (50%) commented that the Guide mirrors best here. The answer of “goeienacht” (good night) to the greeting “goedendag” (good day) is experienced by some test subjects as if the system was correcting the user, which was found annoying.

The questionnaire results per test subject are summed up in appendix E.

5.4 Conclusions

It may be concluded that more formality words should be added for more different formality values, especially for the extreme low ranges. The best matching word is selected (see section 4.8.4). When, for example, a word with a formality of -4 is needed from the shades lexicon, and the lowest available word has a formality of -1, this word is selected. This may cause the difference between the set formality and the word which ends up being selected to be too large, resulting in a sentence whose formality words differ from the preset value. Finally, some adjustments should be made for specific words, such as for instance “adieu”, which should receive a negative formality value, and the verb “markeren”, which should receive a neutral formality value. Some obscure, highly formal verbs might be avoided as well, although these are only used in extreme formal situations, which are rare. Then again, that is what formal language is all about. The suggested adjustments in the formality have been applied to the shaded lexicon.

The general trend in politeness scores given by test subjects matches the system values. Some tactics need to be moved to other groups, and some need to be adjusted. For instance, tactic (1) should be shortened to “kijk op het kaartje” (look at the map) to decrease the amount of information in the utterance. This should reduce its politeness. Other tactics should be

removed, such as the *mind* tactic (9), which received a lot of criticism from test subjects for being unnatural. The suggested changes have been applied to the implementation (see results and discussion).

An issue with the last part of the questionnaire is that untested politeness tactics were used, possibly influencing the ratings. The different alignment settings were noticeable by the test subjects, and their ordering generally agreed with the value α . Dialogue 3 had the best results. Only in the second dialogue, versions with 0% and 50% did not have the expected results, but this could be explained by the politeness tactics that were probably used incorrectly. This indicates the amount of alignment can be properly adjusted by α . For dialogue 1 the test subjects generally preferred versions with 50% and 100% with a 100% begin most popular. For dialogue 2 versions with 0% and 50% were preferred with 50% being most popular. Versions of 50% and 100% received about the same amount of votes for dialogue 3. The lack of clear preference here is probably caused by the not so well received second utterances of the Guide. Overall, a setting of 50% to 100% for α seems to be the best setting.

CHAPTER 6

Conclusions

This chapter presents the conclusions of this project and discusses them. Finally, future work on the Virtual Guide and the alignment model is proposed.

6.1 Conclusions

The project started with a thorough study of the Virtual Guide which can be found in the second chapter. Because such a detailed and ready available manual was not available before, this study will be usable for future work on the Virtual Guide. Some issues were encountered, with the the limitations of the input recognition being the most significant. Other issues were the ontology and the vague answers to questions that are out of domain.

Because of that, the robustness of the Virtual Guide has been improved in the sense that many times more utterances of users can be recognized than was the case at first (see appendix C. Dependent clauses and multiple dialogue acts per user utterance are implemented, as well as support for Multi-word expressions. Finally, answers to questions that are out of domain were designed to be more informative. The supported sentence forms have been tested and were in working order. Limited usability tests also indicated the proper functioning of these new features, although there is still room for improvement such as the reinstatement of the ontology.

In order to increase believability by adding language alignment and increasing language variation, a functional alignment model based on user politeness and formality has been implemented. The alignment model makes more varied dialogues with the user possible. The user's utterances can now be scanned for formality and politeness information, after

which the alignment state is adjusted accordingly. Based on this alignment state, the Guide can generate its utterances and is able to choose politeness tactics and language of varying formality.

A questionnaire indicated that the politeness tactics used by the Virtual Guide generally conveyed the politeness that was intended in the situation, and based on user comments, showed alignment of language had the intended positive effect on the user experience. It also indicated some small changes in the politeness tactics were needed, which were subsequently applied. Determining the formality of language in user utterances and generating language with varying formality proved to be a difficult challenge due to lack of proper and validated research data. Although the trend of the preliminary results for formality were hopeful, more data on the formality of words should be collected to be used for variation in formality.

It can be said however, that the main goal of adding language alignment and more varied system utterances has been met. The alignment of language was noticeable by users and the politeness tactics and on a certain level the formality was perceived by the test subjects as intended. This questionnaire, however, was only an indication, and more research should be conducted on the politeness tactics, formality of language and alignment to substantiate the preliminary findings. Also, extended practical usability tests on the system are necessary.

Finally, recommendations and future work will be discussed.

6.2 Recommendations and Future Work

This project focused on language generation, not agent behaviour. The process of system dialogue act selection is unaffected by the alignment state, only the language used for the dialogue act is varied. It would be interesting for the Virtual Guide to be able to change its behaviour based on the alignment state, and allow it to choose different dialogue acts altogether. This would for instance create the possibility for the Guide to refuse to answer when approached rudely by a user, or to take the initiative in a dialogue. The implementation of indirect politeness tactics is also still left open.

An important issue with in the sentence generation process is the lack of proper data on word formality, and in order to increase the validity, the perceived formality of words by persons could be statistically determined on a greater scale than presented in the previous chapter. As became clear in the evaluation, many more varieties of formality should be added to let the Guide match the language more closely with that indicated by the alignment state. This would also further increase the variety of expressions of the Guide.

Another issue, as discussed by André et al. [ARMB04], is the selection of tactics. They further plan to refine the tactic selection process, and their future research results could prove useful to this implementation. They also propose to include contextual information as discussed in 4.11.

The ontology's functionality remains to be restored (see section 2.8.2). And, because of

time constraints, somewhat less efficient methods were used in the implementation of the alignment model, which could be improved upon. There's always room to increase the recognition of user utterances even more by enlarging the lexicon, expanding the grammar and adding action templates.

Finally, interesting dimensions in emotional expression are generated speech and avatar animation. To further increase the believability, acoustical cues could be added to generated sentences to vary the Guide's tone so that even more politeness variations may be included, as discussed in chapter 7 of Vismans' book [Vis94b]. Variation in intonation might introduce subtle differences in pronunciations so that more complex verbal tactics such as sarcasm ("that's *great*.") can be implemented. The same can be said for (facial) animations of the avatar (smiling when saying something, or looking away from the user might result in a different interpretation of the things that are said).

APPENDIX A

List of MWE's and their lexical representation

The lexical entry that replaces the Multiword Expression is placed after the MWE class and arrow ([CLASS] -> lexical representation). See section 3.7.2.

```
;THANK -> bedankt
\d dank je wel
\d dank je
\d dankje wel
\d dank jewel
\d dank u wel
\d dank u
\d danku wel
\d dank uwel
\d hartstikke bedankt
\d heel erg bedankt
\d veel dank
\d ontzettend bedankt
```

```
;GREETING -> hallo
\g hey daar
\g yo yo
\g hallo daar
\g hey hoi
\g hey hallo
```

APPENDIX A. LIST OF MWE'S AND THEIR LEXICAL REPRESENTATION

```
\g hoi hoi

;AFSCHEID -> doei
\ a de mazzel
\ a de mzzl
\ a tot ziens
\ a ik check je later
\ a ik ga je horen
\ a tot later

;POSITIVE -> goed
\ p geef me de vijf
\ p vette shit
\ p de mazzel
\ p hartstikke mooi
\ p gaat lekker
\ p goed man
\ p mooi man
\ p goed zo
\ p mooi zo
\ p fijn zo

;NEGATIVE -> slecht
\ n nou ja zeg
\ n niet best
\ n bar slecht
\ n oh nee
\ n oh oh
\ n stomme computer
\ n stom ding

;end
```

APPENDIX B

Action Templates

The following action templates were added:

```
<!--je moet de wc tonen -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="6" left="0" right="5" surface="DECL">
    <subj>
      <np_parse id="2" left="0" right="1">
        <head>
          <pn_parse id="1" left="0" right="1" type="PERSON" number="SG" person="2" />
        </head>
      </np_parse>
    </subj>
    <dir_obj>
      <np_parse id="5"/>
    </dir_obj>
    <aux_verbs>
      <verb>moeten</verb>
    </aux_verbs>
    <main_verb>aanwijzen</main_verb>
  </s_parse>
  <arglist>
    <action_argument parse_id="5"/>
  </arglist>
</action_template>
```

```
<!-- ik ben verdwaald -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="4" left="0" right="3" surface="DECL">
    <subj>
      <np_parse id="2" left="0" right="1">
        <head>
          <pn_parse id="1" left="0" right="1" type="PERSON" number="SG" person="1"/>
        </head>
      </np_parse>
    </subj>
    <cop_pred>
      <adj_parse id="3" left="2" right="3" adj="verdwaald"/>
    </cop_pred>
    <main_verb>zijn_pred</main_verb>
  </s_parse>
  <arglist>
    <action_argument parse_id="2"/>
  </arglist>
</action_template>
```

```
<!-- ik zoek de wc -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="1" surface="DECL">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>zoeken</main_verb>
  <dir_obj id="4">
    <np_parse id="5"/>
  </dir_obj>
</s_parse>
<arglist>
  <action_argument parse_id="5"/>
</arglist>
</action_template>
```

APPENDIX B. ACTION TEMPLATES

```
<!-- ik wil naar de wc -->
<action_template fwd_tag="REQUEST" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="9" left="0" right="6" surface="DECL">
    <subj>
      <np_parse id="2" left="0" right="1">
        <head>
          <pn_parse id="1" left="0" right="1" type="PERSON" number="SG" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>willen</main_verb>
    <adv_pps>
      <adv_parse id="3" left="2" right="3" adv="graag" is_wh="false"/>
      <pp_parse id="8" left="3" right="6">
        <prep>
          <prep_parse id="4" left="3" right="4" prep="naar"/>
        </prep>
        <main>
          <np_parse id="7" left="4" right="6">
            <det>
              <det_parse id="5" left="4" right="5" type="DEF"/>
            </det>
            <head>
              <n_parse id="6" left="5" right="6"/>
            </head>
          </np_parse>
        </main>
      </pp_parse>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="7"/>
  </arglist>
</action_template>
```

```
<!-- waar kan ik de wc vinden? waar vind ik de wc?-->
<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="1">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>vinden</main_verb>
    <adv_pps>
      <adv_parse id="4" adv="waar"/>
    </adv_pps>
    <dir_obj id="5">
      <np_parse id="6"/>
    </dir_obj>
  </s_parse>
  <arglist>
    <action_argument parse_id="6"/>
  </arglist>
</action_template>
```

```

<!-- toon (me) de wc -->
<action_template fwd_tag="REQUEST" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="4" surface="IMP">
    <dir_obj>
      <np_parse id="3"/>
    </dir_obj>
    <main_verb>tonen</main_verb>
  </s_parse>
  <arglist>
    <action_argument parse_id="3"/>
  </arglist>
</action_template>

```

```

<!-- kun/wil je het toilet aanwijzen/tonen/markeren/aangeven?
zou je het toilet kunnen/willen aanwijzen/tonen/markeren/aangeven?-->
<action_template fwd_tag="REQUEST" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="6" >
    <subj>
      <np_parse id="2" >
        <head>
          <pn_parse id="1" type="PERSON" number="SG" person="2" />
        </head>
      </np_parse>
    </subj>
    <dir_obj>
      <np_parse id="5"/>
    </dir_obj>
    <aux_verbs/>
    <main_verb>aanwijzen</main_verb>
  </s_parse>
  <arglist>
    <action_argument parse_id="5"/>
  </arglist>
</action_template>

```

```

<!-- ik moet naar de wc -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="6" surface="DECL">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="1" type="PERSON" number="SG" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>moeten</main_verb>
    <adv_pps>
      <pp_parse id="5">
        <prep>
          <prep_parse id="3" prep="naar"/>
        </prep>
        <main>
          <np_parse id="4"/>
        </main>
      </pp_parse>
    </adv_pps>
  </s_parse>

```


APPENDIX B. ACTION TEMPLATES

```
</s_parse>
  <arglist>
    <action_argument parse_id="4"/>
  </arglist>
</action_template>
```

```

<!-- weet je waar de wc is? -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="5" left="0" right="6" surface="YNQ">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="1" type="PERSON" number="SG" person="2"/>
        </head>
      </np_parse>
    </subj>
    <cop_pred>
      <np_parse id="3"/>
    </cop_pred>
    <main_verb>bevinden</main_verb>
    <adv_pps>
      <adv_parse id="4" adv="waar" is_wh="true"/>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="3"/>
  </arglist>
</action_template>

```

```

<!-- Ik wil naar de wc -->
<action_template fwd_tag="REQUEST" action_class="project.dialogue.navigation.ActionTellLocation">
  <s_parse id="1" surface="DECL">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" number="SG" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>willen</main_verb>
    <adv_pps>
      <pp_parse id="4">
        <prep>
          <prep_parse id="5" left="2" right="3" prep="naar"/>
        </prep>
        <main>
          <np_parse id="6" left="3" right="5"/>
        </main>
      </pp_parse>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="6"/>
  </arglist>
</action_template>

```

APPENDIX B. ACTION TEMPLATES

```
<!-- hoe kom ik van de wc bij de tafel? -->
<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellPathFromTo">
  <s_parse id="1">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>komen</main_verb>
    <adv_pps>
      <adv_parse id="4" adv="hoe"/>
      <pp_parse id="5">
        <prep>
          <prep_parse id="6" prep="van"/>
        </prep>
      </pp_parse>
      <pp_parse id="7">
        <prep>
          <prep_parse id="8" prep="bij"/>
        </prep>
      </pp_parse>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="5"/>
    <action_argument parse_id="7"/>
  </arglist>
</action_template>
```

```
<!-- waar moet ik heen? -->
<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellPathTo">
  <s_parse id="5" left="0" right="4" surface="WHQ">
    <subj>
      <np_parse id="2" left="2" right="3">
        <head>
          <pn_parse id="1" left="2" right="3" type="PERSON" number="SG" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>moeten</main_verb>
    <adv_pps>
      <adv_parse id="3" left="0" right="1" adv="waar" is_wh="true"/>
      <adv_parse id="4" left="3" right="4" adv="heen" is_wh="false"/>
    </adv_pps>
  </s_parse>
  <arglist/>
</action_template>
```

```

<!-- hoe kan ik de wc vinden? hoe vind ik de wc?-->
<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellPathTo">
  <s_parse id="1">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>vinden</main_verb>
    <adv_pps>
      <adv_parse id="4" adv="hoe"/>
    </adv_pps>
    <dir_obj id="5">
      <np_parse id="6"/>
    </dir_obj>
  </s_parse>
  <arglist>
    <action_argument parse_id="6"/>
  </arglist>
</action_template>

```

```

<!-- hoe kom ik daar? hoe kan ik daar komen? -->
<action_template fwd_tag="WHQ" action_class="project.dialogue.navigation.ActionTellPathTo">
  <s_parse id="1">
    <subj>
      <np_parse id="2">
        <head>
          <pn_parse id="3" type="PERSON" person="1"/>
        </head>
      </np_parse>
    </subj>
    <main_verb>komen</main_verb>
    <adv_pps>
      <adv_parse id="4" adv="hoe"/>
      <adv_parse id="5" adv="daar"/>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="5"/>
  </arglist>
</action_template>

```

APPENDIX B. ACTION TEMPLATES

```
<!-- wat is de weg / route naar ...? -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionTellPathTo">
  <s_parse id="10" left="0" right="7" surface="DECL">
    <subj>
      <np_parse id="8" left="2" right="7">
        <det>
          <det_parse id="1" left="2" right="3" type="DEF"/>
        </det>
        <head>
          <n_parse id="2" left="3" right="4" noun="weg"/>
        </head>
        <pps>
          <pp_parse id="7" left="4" right="7">
            <prep>
              <prep_parse id="3" left="4" right="5" prep="naar"/>
            </prep>
            <main>
              <np_parse id="6" left="5" right="7"/>
            </main>
          </pp_parse>
        </pps>
      </np_parse>
    </subj>
    <main_verb>zijn_pred</main_verb>
    <adv_pps>
      <adv_parse id="9" left="0" right="1" adv="wat" is_wh="false"/>
    </adv_pps>
  </s_parse>
  <arglist>
    <action_argument parse_id="6"/>
  </arglist>
</action_template>
```

```
<!-- wat is dit, met aanwijzen -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionDescribeObject">
  <s_parse id="4" left="0" right="3" surface="DECL">
    <subj>
      <np_parse id="2" left="2" right="3">
        <head>
          <pn_parse id="1" left="2" right="3" type="DEMO"/>
        </head>
        </np_parse>
      </subj>
      <main_verb>zijn_pred</main_verb>
      <adv_pps>
        <adv_parse id="3" left="0" right="1" adv="wat" is_wh="false"/>
      </adv_pps>
    </s_parse>
    <arglist>
      <action_argument parse_id="2"/>
    </arglist>
  </action_template>
```

```
<!-- dank -->
<action_template fwd_tag="THANK" action_class="project.dialogue.navigation.ActionThank">
  <s_parse id="1"/>
  <arglist/>
</action_template>
```

```
<!-- dat is [positief] (meningsuiting, type van adjective positief -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionPositive">
  <s_parse id="4" left="0" right="3" surface="DECL">
    <subj>
      <np_parse id="2" left="0" right="1">
        <head>
          <pn_parse id="1" left="0" right="1" type="DEMO"/>
        </head>
      </np_parse>
    </subj>
    <cop_pred>
      <adj_parse id="3" type="POSITIVE"/>
    </cop_pred>
    <main_verb>zijn_pred</main_verb>
  </s_parse>
  <arglist/>
</action_template>
```

```
<!-- dat is [negatief] (meningsuiting, type van adjective negatief -->
<action_template fwd_tag="YNQ" action_class="project.dialogue.navigation.ActionNegative">
  <s_parse id="4" left="0" right="3" surface="DECL">
    <subj>
      <np_parse id="2" left="0" right="1">
        <head>
          <pn_parse id="1" left="0" right="1" type="DEMO"/>
        </head>
      </np_parse>
    </subj>
    <cop_pred>
      <adj_parse id="3" type="NEGATIVE"/>
    </cop_pred>
    <main_verb>zijn_pred</main_verb>
  </s_parse>
  <arglist/>
</action_template>
```

```
<!-- positieve interjectie -->
<action_template fwd_tag="POSITIVE" action_class="project.dialogue.navigation.ActionPositive">
  <s_parse id="1"/>
  <arglist/>
</action_template>
```

```
<!-- negatieve interjectie -->
<action_template fwd_tag="NEGATIVE" action_class="project.dialogue.navigation.ActionNegative">
  <s_parse id="1"/>
  <arglist/>
</action_template>
```

APPENDIX C

Supported Sentences

The following sentence structures are supported after the robustness improvements:

Standard sentences

INTERROGATIVE

Waar is de zaal

Waar vind ik de zaal

Hoe kom ik bij de wc

Hoe vind ik de wc

Hoe kom ik daar

Hoe kom ik hier

Zou je de zaal willen tonen

Zou je de zaal kunnen tonen

Kun je de zaal tonen

Wil je de zaal tonen

DECLERATIVE

Je zou de zaal moeten tonen

Je moet de zaal tonen

Ik zoek de zaal

Ik ben verdwaald

Ik moet naar de wc

Ik wil naar de wc

Dat is mooi

Dat is stom

IMPERATIVE

Toon (me) de zaal
Breng me naar de wc
Wat is de weg

Dependent clauses

INTERROGATIVE

Waar kan ik de zaal vinden
Kun je me vertellen waar de wc is
Kun je het toilet aanwijzen/tonen/markeren/aangeven?
Weet je waar de wc is
Hoe kan ik bij de wc komen
Hoe kan ik de wc vinden
Hoe moet ik bij de wc komen
Hoe moet ik daar komen
Hoe moet ik hier komen
Hoe kan ik daar komen
Hoe kan ik hier komen

DECLERATIVE

Je moet me vertellen/zeggen waar de zaal is
Je moet me vertellen/zeggen hoe ik bij de zaal kan komen

Examples of combined dialogue acts

Hoi waar is de zaal
Hoi, waar is de zaal
Ik ben verdwaald waar is de zaal
Ik ben verdwaald en waar is de zaal
Waar is de wc en hoe kom ik daar
Ik ben verdwaald, waar is de zaal
Ik zoek de zaal hoe kom ik daar
Ik zoek de zaal waar kan ik die vinden
Ik zoek de zaal, waar is die
Ik zoek de zaal waar is die

APPENDIX D

Questionnaire

Deel 1: Formele of informele woordkeuze

De keuze van woorden van de Virtuele Gids hangt af van de keuze van woorden van de bezoeker. Als de bezoeker bijvoorbeeld alledaagse taal zoals ‘doei’ gebruikt, zal de Gids ongeveer op dezelfde manier antwoorden. Hetzelfde geldt wanneer je wat moeilijkere woorden gebruikt. Iedereen heeft wel een idee of bepaalde woorden wat netter of meer alledaags zijn. Zo zal je taalgebruik tegen een vreemde vaak netter (formeler) zijn, en ben je meer alledaags (informeel) als met je vrienden praat.

Hierna zullen 20 zinnen volgen die de Virtuele Gids zou kunnen zeggen in een gesprek. Een gesprek zou kunnen gaan over het aanwijzen van bijvoorbeeld de zaal op een kaartje en antwoorden op vragen zoals ”hoe kan ik bij de wc komen?”. Verder komen ook nog begroetingen en andere korte reacties aan bod. Het hangt in dit experiment dus niet af van het gebruik van ”je” of ”u”, maar vooral van hoe net of alledaags je het woordgebruik inschat.

Hoe formeel schat jij het woordgebruik in de volgende zinnen op de schaal van zeer informeel tot zeer formeel? Selecteer je keuze onder iedere zin.

Bij elke vraag mag commentaar toe voegen. Dit is echter niet verplicht!

Vul hier je naam en leeftijd in. Deze zullen vertrouwelijk worden behandeld.

1. Je hebt waarschijnlijk woorden gezegd die ik niet snap.

2. Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven.
3. Je zult het anders moeten zeggen.
4. Adieu.
5. Als je het niet erg vind, kun je op het plattegrondje kijken waar ik de zaal heb aangestipt.
6. Er zijn woorden gezegd die ik niet snap.
7. Geen probleem.
8. Je locatie is gemarkeerd door het kruis.
9. Hey. (begroeting)
10. Tot je dienst.
11. Ik heb je niet helemaal begrepen.
12. Je zult het op een andere wijze moeten verbaliseren.
13. Je zou op de plattegrond moeten kijken waar ik de zaal heb gemarkeerd.
14. Het indiceren van de route naar toilet is gereed.
15. Doei.
16. Je zult het op een andere wijze moeten verwoorden.
17. Goedenacht.
18. Het aanstippen van de weg van het toilet naar de zaal is klaar.
19. Je zou op het plan moeten kijken waar ik de zaal heb gemarkeerd.
20. Als je het nog eens probeert, lukt het vast wel.

Einde deel 1

Deel 2: Beleefdheid

Er zijn veel verschillende manieren waarop je iemand iets kunt vragen. Je kunt iets aardig vragen of op een botte manier. Dat heeft niet altijd iets te maken met iemand met "je" of met "u" aan te spreken, maar destemeeer met de manier waarop je een vraag formuleert. Hierna zal een twintigtal zinnen volgen en de bedoeling is dat jij op een schaal aangeeft hoe beleefd je een zin vindt. Stel je voor dat de Gids deze zinnen gebruikt in een gesprek met jou, en beoordeel hoe je de uitspraak van de Gids op zou vatten. Je kunt kiezen uit zeer onbeleefd, onbeleefd, niet beleefd en niet onbeleefd (neutraal), beleefd en zeer beleefd.

1. Is het mogelijk dat je op het kaartje kijkt waar ik de zaal heb gemarkeerd?
2. Je vind het vast niet erg om het nog een keer te proberen.
3. Je moet het anders zeggen.
4. Je zou het nog een keer moeten proberen.
5. Ik vroeg me af of je op het kaartje wilt kijken waar ik de zaal heb gemarkeerd.
6. Is het mogelijk dat het nog een keer geprobeerd kan worden?
7. Iemand zou het nog een keer moeten proberen.
8. Nog een keer proberen?
9. Zou je het anders kunnen formuleren?
10. Kun op het kaartje kijken waar ik de zaal heb gemarkeerd?
11. Probeer het nog eens, vriend
12. Kun je eventjes op het kaartje kijken waar ik de zaal heb gemarkeerd?
13. Wil je het nog een keer proberen?
14. Kijk op het kaartje waar ik de zaal heb aangegeven
15. De vraag is of je het anders wilt formuleren
16. Als je het niet erg vind, kun je op het kaartje kijken waar ik de zaal heb gemarkeerd.
17. Kun je het misschien anders formuleren?
18. Zullen we het nog een keer proberen?
19. Zou je op het kaartje willen kijken waar ik de zaal heb gemarkeerd?
20. Als je het anders formuleert, lukt het vast wel

21. Sorry, kun je het nog eens proberen?

Einde deel 2

Deel 3: Aanpassen aan de gesprekspartner

Sommige mensen passen zich eerder aan hun gesprekspartner aan dan anderen. Hieronder zullen enkele conversaties volgen. Elk conversatie stelt een gesprek voor tussen de bezoeker en de Gids. De Gids is zo geprogrammeerd dat ze haar taalgebruik kan aanpassen aan dat van de bezoeker. Hierna zullen 9 dialogen van een bezoeker met de Virtuele Gids volgen. Nu is aan jou te taak om te beoordelen hoe de gids zich volgens jou aanpast aan de bezoeker.

1. BEZOEKER: hi
GIDS: Goeienavond.
BEZOEKER: toon me de zaal
GIDS: Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven.
BEZOEKER: doe
GIDS: Dag.
2. BEZOEKER: hey
GIDS: Hallo daar.
BEZOEKER: ik wil graag naar de zaal
GIDS: Is het mogelijk dat er op het plattegrondje gekeken wordt waar ik de zaal heb aangestipt.
BEZOEKER: bedankt
GIDS: Geen dank.
BEZOEKER: later
GIDS: Doeg.
3. BEZOEKER: goedendag
GIDS: Goeienavond.
BEZOEKER: weet u misschien waar het toilet is
GIDS: Als je het niet erg vind, kun je op de kaart kijken waar ik het toilet heb aangegeven.
BEZOEKER: dank u wel
GIDS: Graag gedaan.

APPENDIX D. QUESTIONNAIRE

4. BEZOEKER: hi
GIDS: Hi.
BEZOEKER: toon me de zaal
GIDS: Kijk op het kaartje waar ik de zaal heb aangestipt.
BEZOEKER: doi
GIDS: Doei.
5. BEZOEKER: hey
GIDS: Hey daar.
BEZOEKER: ik wil graag naar de zaal
GIDS: ik vroeg me af of je op het kaartje kunt kijken waar ik de zaal heb aangestipt.
BEZOEKER: bedankt
GIDS: Geen dank.
BEZOEKER: later
GIDS: Later.
6. BEZOEKER: goedendag
GIDS: Goeienavond.
BEZOEKER: weet u misschien waar het toilet is
GIDS: Zou u op de kaart willen kijken waar ik het toilet heb aangegeven?
BEZOEKER: dank u wel
GIDS: Het is mijn taak.
7. BEZOEKER: goedendag
GIDS: Goedendag.
BEZOEKER: weet u misschien waar het toilet is
GIDS: kunt u misschien op het kaartje kijken waar ik de zaal heb gemarkeerd?
BEZOEKER: dank u wel
GIDS: Het is mijn taak.
8. BEZOEKER: hi
GIDS: Hallo daar.
BEZOEKER: toon me de zaal
GIDS: Is het mogelijk dat je op het plattegrondje kijkt waar ik de zaal heb aangestipt?
BEZOEKER: doi
GIDS: Tot een andere keer.

9. BEZOEKER: hey
GIDS: Goeie nacht.
BEZOEKER: ik wil graag naar de zaal
GIDS: Je vind het vast niet erg om op de kaart te kijken waar ik de zaal heb aangegeven.
BEZOEKER: bedankt
GIDS: Geen dank.
BEZOEKER: later
GIDS: Tot ziens.
-

Hieronder staan 3 versies van een gesprek van een gebruiker met de Gids.

Versie 1

BEZOEKER: Hi
GIDS: Hallo daar.
BEZOEKER: Toon me de zaal
GIDS: Is het mogelijk dat je op het plattegrondje kijkt waar ik de zaal heb aangestipt?
BEZOEKER: Doe
GIDS: Tot een andere keer.

Versie 2

BEZOEKER: Hi
GIDS: Goeienavond.
BEZOEKER: Toon me de zaal
GIDS: Je zou op de kaart moeten kijken waar ik de zaal heb aangegeven.
BEZOEKER: Doe
GIDS: Dag.

Versie 3:

BEZOEKER: Hi
GIDS: Hi.
BEZOEKER: Toon me de zaal
GIDS: Kijk op het kaartje waar ik de zaal heb aangestipt.
BEZOEKER: Doe
GIDS: Doe.

Kies de versie van het dialoog uit die je het beste bevalt, en geef aan in het tekstvak waarom:

APPENDIX D. QUESTIONNAIRE

Hieronder staan 3 versies van een gesprek van een gebruiker met de Gids.

Versie 1:

BEZOEKER: Goedendag

GIDS: Goeienavond.

BEZOEKER: Weet u misschien waar het toilet is

GIDS: Als je het niet erg vind, kun je op de kaart kijken waar ik het toilet heb aangegeven.

BEZOEKER: Dank u wel

GIDS: Graag gedaan.

Versie 2

BEZOEKER: Goedendag

GIDS: Goeienavond.

BEZOEKER: Weet u misschien waar het toilet is

GIDS: Zou u op de kaart willen kijken waar ik het toilet heb aangegeven?

BEZOEKER: Dank u wel

GIDS: Het was me een genoegen.

Versie 3:

BEZOEKER: Goedendag

GIDS: Goedendag.

BEZOEKER: Weet u misschien waar het toilet is

GIDS: Kunt u misschien op het kaartje kijken waar ik de zaal heb gemarkeerd?

BEZOEKER: Dank u wel

GIDS: Het was me een genoegen.

Kies de versie van het dialoog uit die je het beste bevalt, en geef aan in het tekstvak waarom:

Hieronder staan 3 versies van een gesprek van een gebruiker met de Gids.

Versie 1:

BEZOEKER: Hey

GIDS: Goeiemiddag.

BEZOEKER: Ik wil graag naar de zaal

GIDS: Je vindt het vast niet erg om op de kaart te kijken waar ik de zaal heb aangegeven.

BEZOEKER: Bedankt

GIDS: Geen dank.

BEZOEKER: Later

GIDS: Tot ziens.

Versie 2:

BEZOEKER: Hey

GIDS: Hallo daar.

BEZOEKER: Ik wil graag naar de zaal

GIDS: Is het mogelijk dat er op het plattegrondje gekeken wordt waar ik de zaal heb aangestipt.

BEZOEKER: Bedankt

GIDS: Geen dank.

BEZOEKER: Later

GIDS: Doeg.

Versie 3:

BEZOEKER: Hey

GIDS: Hey daar.

BEZOEKER: Ik wil graag naar de zaal

GIDS: Ik vroeg me af of je op het kaartje kunt kijken waar ik de zaal heb aangestipt.

BEZOEKER: Bedankt

GIDS: Geen dank.

BEZOEKER: Later

GIDS: Later.

Kies de versie van het dialoog uit die je het beste bevalt, en geef aan in het tekstvak waarom:

Einde deel 3

Dit is het einde van het onderzoek. Heb je nog op- of aanmerkingen over de Virtuele Gids of deze vragenlijst? Geef deze dan in het vak hieronder.

Commentaar:

APPENDIX E

Questionnaire Results

APPENDIX E. QUESTIONNAIRE RESULTS

Formality results																				
Test subject	Questions																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	29	20
1)	5	3	4	1	4	4	4	2	5	1	4	1	3	1	4	2	1	4	2	4
2)	4	3	3	5	3	2	3	2	4	2	3	1	3	2	4	2	3	4	2	4
3)	5	5	5	3	4	4	4	4	5	4	4	4	4	2	4	4	2	4	4	4
4)	3	3	3	3	3	3	3	3	3	2	3	1	3	1	3	2	2	2	3	3
5)	2	2	2	2	2	3	3	3	3	3	3	2	2	1	3	2	3	2	3	3
6)	3	5	3	5	5	2	2	2	3	3	3	1	3	1	4	3	2	3	3	3
7)	1	4	3	5	3	3	4	3	5	3	3	2	3	1	4	3	2	3	3	3
8)	3	4	4	5	5	4	4	2	5	3	2	1	4	1	4	3	2	4	3	3
9)	4	3	4	2	3	4	3	3	5	4	4	1	3	2	4	3	2	2	3	3
10)	3	3	3	5	4	2	4	4	5	3	4	3	4	2	5	3	3	4	4	4
11)	5	3	4	5	3	4	3	2	5	2	2	1	1	1	5	2	2	4	2	3
12)	2	2	2	4	2	3	4	1	4	2	3	1	2	1	4	1	3	2	2	3
13)	5	4	5	4	5	4	3	3	5	3	3	1	3	2	5	1	2	3	3	3
14)	4	5	2	4	2	2	2	2	4	2	2	2	2	2	3	2	2	2	2	2
15)	3	5	5	3	3	2	1	2	4	1	2	2	5	2	3	2	2	4	3	4
16)	2	2	3	2	4	2	2	2	3	2	2	1	2	2	2	2	2	2	2	2
17)	4	4	5	5	3	2	4	2	4	2	3	1	3	1	4	2	2	3	3	5
18)	4	5	3	1	2	4	2	4	5	2	1	1	3	1	5	2	2	2	4	4
19)	2	2	3	3	3	2	2	2	4	2	2	3	2	4	4	2	2	2	2	2
20)	2	2	4	5	4	2	4	1	4	4	2	1	4	1	4	2	2	4	4	4
21)	4	5	5	5	3	4	3	3	5	3	4	3	3	2	5	2	2	3	3	3
22)	4	4	4	3	2	2	2	3	5	2	2	2	4	2	5	2	2	2	3	3
23)	2	3	3	3	4	4	4	4	5	1	3	3	3	3	5	3	2	3	3	4
24)	3	1	2	2	4	3	4	5	3	4	2	3	3	4	4	4	5	4	4	3
25)	4	4	4	5	4	2	3	4	5	4	4	4	4	1	5	4	1	4	4	5

Table E.1: Questionnaire results for formality of language

Politeness results																				
Test subject	Questions																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	29	20
1)	1	4	5	3	1	2	5	3	2	4	5	1	2	4	4	1	3	4	1	4
2)	2	4	4	3	3	2	3	4	2	3	5	2	2	4	4	2	2	2	2	4
3)	2	4	4	4	2	2	3	4	2	2	4	2	2	5	4	2	2	2	2	4
4)	2	3	4	3	2	2	3	5	3	2	4	3	3	4	5	3	3	2	2	5
5)	2	2	3	3	2	2	2	3	2	2	2	2	3	3	2	2	2	3	2	4
6)	2	4	5	4	2	2	4	4	2	2	4	2	2	4	3	2	2	1	2	5
7)	2	3	4	3	2	2	3	4	2	2	4	3	2	4	3	2	2	2	2	5
8)	2	2	3	3	2	3	3	5	1	3	5	3	2	4	4	3	1	1	3	5
9)	4	3	4	2	2	2	4	5	2	3	4	3	2	4	4	3	2	3	2	4
10)	2	4	4	4	2	2	4	3	2	2	3	2	2	4	3	2	2	2	2	4
11)	2	3	4	3	2	3	2	5	1	1	4	3	3	4	5	3	2	4	2	5
12)	2	3	3	2	2	2	2	4	2	2	3	3	2	4	4	2	2	2	2	5
13)	1	3	4	3	3	2	4	3	2	2	4	3	2	3	4	2	3	3	2	2
14)	2	2	4	3	2	2	4	4	2	2	3	3	2	4	2	2	2	2	2	2
15)	4	3	4	3	4	2	2	2	2	3	4	4	3	2	3	2	1	2	2	2
16)	2	3	4	2	2	2	3	4	2	2	4	2	2	2	3	2	2	2	2	2
17)	2	3	5	3	3	2	4	5	1	2	4	4	3	4	3	2	1	3	3	4
18)	3	5	4	4	1	1	3	4	1	2	2	2	3	5	4	2	2	2	2	1
19)	3	3	4	2	3	1	2	3	1	3	4	2	2	3	1	1	1	2	2	2
20)	2	3	4	2	2	2	4	3	2	3	3	2	2	4	2	2	2	2	2	2
21)	2	4	3	2	2	2	3	4	2	2	4	3	2	3	3	4	2	2	2	2
22)	2	3	3	2	1	1	3	4	2	2	3	2	2	4	2	2	2	3	2	3
23)	2	2	4	4	3	2	2	3	3	3	4	3	3	3	3	2	2	2	2	2
24)	3	2	4	3	3	2	3	3	2	3	3	3	3	3	2	3	2	2	2	2
25)	2	2	3	3	3	2	2	4	3	2	4	1	1	4	4	3	2	1	1	4

Table E.2: Questionnaire results for politeness

APPENDIX E. QUESTIONNAIRE RESULTS

Alignment results												
Test subject	Questions											
	1	2	3	4	5	6	7	8	9	1	2	3
1)	1	3	3	4	3	4	4	3	1	3	2	1
2)	2	3	4	4	4	3	3	1	2	2	2	2
3)	3	1	4	4	3	3	3	1	1	3	3	3
4)	2	3	4	4	3	4	4	2	3	2	1	2
5)	2	4	3	4	3	4	4	3	3	3	1	3
6)	4	4	4	4	4	3	3	3	4	1	2	3
7)	2	2	4	4	3	4	4	3	3	2	1	1
8)	3	3	2	1	4	3	3	1	1	1	2	2
9)	2	3	4	3	4	3	2	2	2	2	2	2
10)	2	2	2	3	2	2	3	2	2	3	1	3
11)	3	3	1	4	4	3	4	2	1	3	3	3
12)	3	1	1	4	2	2	2	1	1	2	-	-
13)	2	3	4	4	4	4	3	3	4	2	1	1
14)	2	2	3	1	1	4	4	3	3	2	2	3
15)	3	1	2	3	3	4	4	3	3	2	2	2
16)	2	2	3	4	3	3	4	1	3	2	1	2
17)	2	3	4	4	4	3	3	3	3	3	2	2
18)	1	4	3	4	3	2	2	2	2	2	1	2
19)	3	3	2	4	3	3	3	3	3	2	2	1
20)	2	3	4	4	4	4	4	3	1	2	2	1
21)	3	3	4	4	3	4	4	2	2	3	3	3
22)	2	3	3	4	3	4	4	3	3	1	3	1
23)	1	2	4	4	3	4	3	2	1	3	2	3
24)	1	2	3	4	4	1	2	1	1	3	1	3
25)	3	2	4	4	3	4	4	1	4	3	3	1

Table E.3: Questionnaire results for alignment

BIBLIOGRAPHY

- [ABL99] L. Ardissono, G. Boella, and L. Lesmo. Politeness and speech acts. In *Proceedings Workshop on Attitude, Personality and Emotions in User-Adapted Interaction*, pages 41–55, Banff, Canada., 1999.
- [AC97] James Allen and Mark Core. *Draft of DAMSL Dialog Act Markup in Several Layers*, 1997.
- [Ace05] AceBit. Winsurvey 2005. <http://www.winsurvey.com/>, 2005.
- [ARMB04] Elisabeth Andre, Matthias Rehm, Wolfgang Minker, and Dirk Buhler. Endowing spoken language dialogue systems with emotional intelligence. In *Lecture Notes in Computer Science*, pages 178–187, University of Augsburg, Germany, 2004.
- [Bat94] Joseph Bates. The role of emotion in believable agents. *Commun. ACM*, 37(7):122–125, 1994.
- [BIOW05] Carsten Brockmann, Amy Isard, Jon Oberlander, and Michael White. Modelling alignment for affective dialogue. In *Proceedings of the Workshop on Adapting the Interaction Style to Affective Factors at the 10th International Conference on User Modeling (UM-05)*, Edinburgh, UK, 2005.
- [BK04] M. Bouman and R.C.P. Korthuis. Gesproken routebeschrijvingen in een virtual environment. Master’s thesis, Universiteit Twente, Enschede, 2004.
- [BL87] Penelope Brown and Stephen C. Levinson. *Politeness - Some universals in language usage*. Cambridge University Press, Cambridge University, Cambridge, 1987.

- [BMJPJC06] H.P. Branigan, J. Pearson M. J. Pickering, J.F. McLean, and C.I. Nass. Syntactic alignment between computers and people: The role of belief about mental states. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*, pages 186–191, 2006.
- [Bus05] D. Bussink. Building a corpus and testing speech recognition. Universiteit Twente, Enschede, 2005.
- [dJO07] Prof. dr. J. Odijk. Identification and representation of multi-word expressions proposal. <http://www-uilots.let.uu.nl/irme/wordpress>, 2007.
- [exe07] Cogentex, inc. - exemplars framework. <http://www.cogentex.com/>, 2007.
- [FH02] M. Fleischman and E. Hovy. Towards emotional variation in speech-based natural language generation. In *Proceedings of the Second International Natural Language Generation Conference (INLG'02)*, New York, USA, 2002.
- [FJM98] H. Francis and D. Jean-Marc. Variation in the contextuality of language: an empirical measure. In *Foundations of Science. Ide, Nancy and Vronis, Jean (1998) Word sense disambiguation: The state of the art.*, 1998.
- [GH03] Curry Guinn and Rob Hubal. Extracting emotional information from the text of spoken dialog. In *9th International Conference on User Modeling*, Johnstown, PA, USA, 2003.
- [GP04] Simon Garrod and Martin J. Pickering. Why is conversation so easy? In *Trends in Cognitive Sciences, Volume 8, Number 1*, pages 8–11. Elsevier, 2004.
- [GRW05] Swati Gupta, Daniela M. Romanor, and Marilyn A. Walker. Politeness and variation in synthetic social interaction. In *Human-Animated Character Interaction Workshop*, Edinburgh, 2005.
- [GWR07] Swati Gupta, Marlilyn A. Walker, and Daniela M. Romano. Generating politeness in task based interaction: An evaluation of the effect of linguistic form and culture. In Eduard Hovy, editor, *Proceedings of the Eleventh International Workshop on Natural Language Generation*, pages 57–64, Sheffield, UK, S1 4DP, 2007. Association for Computational Linguistics.
- [HD99] Francis Heylighen and Jean-Marc Dewaele. Formality of language: definition, measurement and behavioral determinants. In *Internal Report, Center Leo Apostel*, Free University of Brussels, 1999.
- [HodAN03] D. H. W. Hofs, H. J. A. op den Akker, and A. Nijholt. A generic architecture and dialogue model for multimodal interaction. In P. Paggio, K. Jokinen, and A. Jönsson, editors, *Proceedings of the 1st Nordic Symposium on Multimodal Communication, Copenhagen, Denmark*, volume 1, pages 79–91, Copenhagen, September 2003. CST Publication, Center for Sprogteknologi.

- [HRBS95] B. Hayes-Roth, L. Brownston, and E. Sincoff. Directed improvisation by computer characters. Technical report, Stanford Univ. Knowledge Systems Laboratory, Stanford Univ., Palo Alto, Calif., 1995.
- [IAB⁺06] Isard, Amy, Brockmann, Carsten, and Oberlander. Individuality and alignment in generated dialogues. In *Proceedings of the 4th International Natural Language Generation Conference (INLG-06)*, pages 22–29, Sydney, Australia, 2006.
- [JMAR05] L. Johnson, R. Mayer, E. Andr, and M. Rehm. Cross-cultural evaluation of politeness in tactics for pedagogical agents. In *Proc. of the 12th Int. Conf. on Artificial Intelligence in Education (AIED)*, University of California, Santa Barbara, 2005.
- [JRB⁺04] L.W. Johnson, P. Rizzo, W.E. Bosma, M. Ghijsen, and H. van Welbergen. Generating socially appropriate tutorial dialog. In E. Andr, L. Dybkjr, W. Minker, and P. Heisterkamp, editors, *ISCA Workshop on Affective Dialogue Systems*, volume 3068 of *LNAI*, pages 254–264, Berlin Heidelberg New York, 2004. Springer Verlag. ISBN=3-540-22143-3.
- [KB06] Simon Keizer and Harry Bunt. Multidimensional dialogue management. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, page pages 3745, Department of Language and Information Science, Faculty of Arts, Tilburg University P.O. Box 90153, 5000 LE Tilburg, The Netherlands, 2006. Association for Computational Linguistics.
- [KodAN02] S. Keizer, H. J. A. op den Akker, and A. Nijholt. Dialogue act recognition with bayesian networks for dutch dialogues. In K. Jokinen and S. McRoy, editors, *Proceedings 3rd SIGdial Workshop on Discourse and Dialogue, Philadelphia, Pennsylvania*, pages 88–94, Philadelphia, July 2002. ACL SIGDIAL.
- [LK82] Willem J.M. Levelt and Stephanie Kelter. *Surface Form and Memory in Question Answering*, pages 78–106. Universeit Nijmegen, 1982.
- [LL94] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.
- [MO03] D. Morand and R. Ocker. Politeness theory and computer-mediated communication: A sociolinguistic approach to analyzing relational messages. In *Proceeding of the 36th Hawaii International Conference on System Sciences. HICSS’03.*, 2003.
- [NH00] A. Nijholt and G. H. W. Hondorp. Communicerende agents en avatars in virtuele werelden. *RE-Cursief*, 8(2):13–23, December 2000.

- [PE04] Daniel S. Paiva and Roger Evans. A framework for stylistically controlled generation. In *Lecture Notes in Computer Science*, pages 120–129, University of Sussex, Brighton, UK, 2004.
- [PLT06] Ronakkumar Patel, Anton Leuski, and David Traum. Dealing with out of domain questions in virtual characters. Technical report, Institute for Creative Technologies University of Southern California Marina del Rey, CA 90292, USA, 2006.
- [PPM04] Kaska Porayska-Pomsta and Chris Mellish. Modelling politeness in natural language generation. In Anja Belz, Roger Evans, and Paul Piwek, editors, *Lecture Notes in Computer Science*, pages 141–150, Edinburgh EH8 9LW, UK, 2004. Springer Berlin / Heidelberg.
- [Taa07] Van Dale Taalweb. Onlinewoordenboek. <http://www.vandale.nl/opzoeken/woordenboek/>, 2007.
- [Vis94a] Roel Vismans. Beleefdheid, nederlandse modale partikels en het partikelloze engels. *Colloquium Neerlandicum*, 12:269–291, 1994.
- [Vis94b] Roel Vismans. *Modal Particles in Dutch directives: a study in Functional Grammar*. IFOTT, Vrije Universiteit Amsterdam, 1994.
- [WC98] Michael White and Ted Caldwell. EXEMPLARS: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 266–275, New Brunswick, New Jersey, 1998. Association for Computational Linguistics.
- [WCW96] M. Walker, J. Cahn, and S. Whittaker. Linguistic style improvisation for lifelike computer characters. In *Entertainment and AI / A-Life, Papers from 1996 AAAI Workshop. Available as AAAI Technical Report WS-96-03.*, 1996.