MULTIMODAL REFERENCE

STUDIES IN AUTOMATIC GENERATION OF MULTIMODAL REFERRING EXPRESSIONS

IELKA VAN DER SLUIS

IELKA FRANCISCA VAN DER SLUIS

MULTIMODAL REFERENCE

STUDIES IN AUTOMATIC GENERATION OF MULTIMODAL REFERRING EXPRESSIONS

The project of thesis was funded by SOBU (Samenwerkingsorgaan Brabantse Universiteiten; *Organization for cooperation between universities in the Brabant region*)

Published by Uitgevershuis BUG 1907 - 2005 voorheen H.N. Werkman

Address: Hooge der Aa 27, 9712 AD Groningen
Phone: +31 (0)50 312 16 77
Fax: +31 (0)50 314 05 39
Email: bug@boonuitgeverij.nl

Typeset in LaTeX
Cover: Ielka van der Sluis, Tilburg 2005
ISBN 90.75913.443
NUR 984
Keywords: Language generation, Multimodal

Multimodal Reference

Studies in Automatic Generation of Multimodal Referring Expressions

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Tilburg,
op gezag van de rector magnificus,
prof. dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen ten overstaan van
een door het college voor promoties aangewezen commissie
in de aula van de Universiteit
op maandag 19 december 2005 om 16.15 uur

door

Ielka Francisca van der Sluis

geboren op 14 februari 1972

te Assen

Promotor:	Prof. dr. H.C. Bunt
Copromotor:	Dr. E.J. Krahmer

Pete: “Oh dear ... Lucy? .. Lucy, this is Pete Martell,
Lucy .. put Harry on the horn”
Lucy: “Sheriff, it’s Pete Martell up at the mill ...
Uhm, I’m gonna transfer to the phone on the table by the red chair ..
[points in the direction of the phone]
the ... the red chair, against the wall, uh the little table,
with the lamp on it, the lamp that we moved from the corner? ..
the black phone, not the brown phone”
[phone rings]
Harry: “Morning Pete, Harry ...”

Taken from the TWIN PEAKS Pilot, 1990.
TWIN PEAKS (c) Paramount Pictures. Used with permission.

Acknowledgements

A proper expression of my thanks and gratitude for help and support to all people in some way involved in finalizing my Ph.D. project, would make this book at least twice its current size. To keep the attention on the thesis itself, I restrict these acknowledgements to the most important. First of all, my utmost gratitude is due to my supervisors Harry Bunt and Emiel Krahmer, who successfully guided me through the whole process with breathtaking expertise, engagement and encouragement. Both, tirelessly, read numerous earlier versions of this thesis and helped me in structuring and rephrasing the text up to the point where it could be published as the book you have in your hands right now.

Harry offered me a great opportunity to explore the world of computational linguistics, by attracting my interest in the project 'Context-based Natural Language Generation in Multimodal Human-Computer Interaction'. During my years in Tilburg Harry gave me the chance to be involved in several valuable projects and events like the International Workshop on Computational Semantics (IWCS) series, the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information and the Nederlandse Organisatie voor Taal & Spraaktechnologie (The Dutch Organization on Language and Speech Technology, NOTaS). I have enjoyed these experiences very much and am very grateful that I was able to broaden the perspective of my Ph.D. project in such a constructive way.

Emiel has been the best daily supervisor I can imagine. Emiel's enormous enthusiasm, creativity, intellectual and editorial skills as well as his incredible patience and his ever cheery mood lead to innovative and productive output, which is demonstrated by the publications of our joint work at the CLIN, ACL, ENLG, LREC, ICSLP and ST&D workshops and conferences. Emiel put a huge amount of effort into reading and commenting on my ideas and writings, which I appreciate as highly as our regular discussions that helped me to look at things from the bright side when necessary, and stay on the right track to bring this project to a good end.

Several people helped me in various ways to complete this thesis: I have to thank Anton Nijholt, Dafydd Gibbon, Fons Maes, Jan Kooistra, Kees van Deemter, Mandy Schiffrin, Mariët Theune, Robbert-Jan Beun and Walter Daelemans for reading and providing useful comments and rephrasing. Thanks are due as well to the members of the dialogue group: Hans van Dam, Geke Hootsen, Harry Bunt, Huub Prüst, Jeroen Geertzen, Mandy Schiffrin, Rintse van der Werf, Roser Morante, Simon Keizer and Yann Girard. Our weekly meetings provided a profitable platform for intensive brainstorming, collective reading, fruitful discussions, data analysis, and talk rehearsals. Thanks to Bertjan Busser for the software and hardware support. Thanks to Rein Cozijn who helped me out with the statistics. Thanks to Hans van Dam, a walking help desk, who gave me an excellent crash course in graphics and assisted me with speech software. Thanks to the various companies, which made the publication of this thesis financially possible. Thanks to Kees Boon from Uitgevershuis BuG, who was willing to publish my thesis and, as always, did a wonderful job.

Furthermore, I am really pleased with the warm welcome I received at the University of Bielefeld, where I got to meet a wonderful group of sophisticated people who offered me a glance in their impressive kitchen. The way in which Alfred Kranstedt, Andy Lücking, Hannes Rieser, Peter Kühnlein and Thies Pfeiffer combine practice and theory helped me to see the relevance of my own work, which is really nice when one is writing up. I am very grateful for the valuable discussions and comments I received, especially from Alfred, and hope for constructive cooperation in the future. Apart from the people in Bielefeld, there were many people I could count on for help, comments and suggestions, I like to mention: Arjan van Hessen, Jacques Terken, Kees van Deemter, Mariët Theune, Paul Piwek, René Ahn and Rodger Kibble.

At the Department of Computational Linguistics I felt embraced by an outstanding group of colleagues, who contributed to a motivating and agreeable working environment, thanks to: Anne Adriaensen, Antal van den Bosch, Bertjan Busser, Elias Thijsse, Erik Tjong Kim Sang, Erwin Marsi, Hans Paijmans, Huub Prüst, Iris Hendrickx, Jorn Veenstra, Jakub Zavrel, Jeroen Geertzen, Ko van der Sloot, Martin Reynaert, Menno van Zaanen, Olga van Herwijnen, Paul Vogt, Sander Canisius, Reinhard Muskens, Rintse van der Werf, Roser Morante, Simon Keizer, Toine Bogers, Yann Girard, Walter Daelemans and the amiable people of the Faculty of Arts. Here, I owe special thanks to the ones who became trusted friends and who greatly enriched my life in Tilburg: Hans van Dam, Mandy Schiffrin, Piroska Lendvai Rudenko and Sabine Buchholz.

Sabine really made me feel at home in my first years in Tilburg, and I regret that her move to Cambridge made it less easy to pop in now and then. We shared a lot of good times cycling, skating, gaming and cooking meals. Comfort and joy I received in large amounts from Piroska and Yevgen Rudenko. We had great fun visiting art exhibitions, concerts and ballet performances, watching strange films and lots of photos of our respective travels. I feel honored with two beautiful paranymfs at my defense; a noteworthy task which Piroska and Sabine are willing to perform. I am very grateful for the generous support from Hans, which helped me very much in recovering from my physical distress and got me back on track. Although, I only got to meet Mandy in my final months in Tilburg, I feel like I have known her for years. She has been a fantastic companion during a hectic period, in which I was finishing up this thesis and looking for another job.

I am glad that moving far away to the south of the Netherlands, did not hinder my close contacts in keeping our friendship alive. I found a terrific distraction from my work, but also a way to reflect on things in the lengthy and cheery walks through various parts of the Netherlands with our core walking group: Arjan Meijrink, Bert Ommen, Elka Oudenampsen, Jan Kooistra, Hans van Dam, Remkes Kooistra and Wytske Bottema. Thanks, I look forward to exploring Scotland with you! Thanks as well to my dear and faithful friends for their support, understanding, reliance and precious company: Anne Breitbarth, Annemarie Meijers, John Hazeveld, Hyun Mi Kang, Jori Jansen, Marike van Gijsel and Wytske Bottema.

Fundamental and essential loyalty and encouragement I received from my family and all the friends of the family. First of all my deepest thanks to SuperJan Kooistra, for his unlimited trust, love and understanding, as well as for his helpfulness and availability in every respect. Gratefulness to my loving parents, Meent en Trijni van der Sluis, for their solid confidence, devotion and support. Thanks to Vinayak, Willem and Elena for their affection and their interest in my wellbeing. Although, in recent years, we tragically lost too many people, among which my beloved father, my dear grandmother and my favorite aunt, I think we managed to sustain an intimate, supportive and stimulating kinship, which fills me with joy and helps me to move on.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Human-computer interaction (HCI) studies the interaction between people (users) and computers which takes place at the user interface. This includes the hardware, (i.e., input and output devices), as well as the software (e.g., determining which, and how, information is presented to the user or to the system). Advances in HCI provide evidence that the use of multiple modalities, like for instance speech and gesture, in both the input and the output will result in systems that are more robust and efficient to use (Oviatt, 1999). Up until now, however, multimodal systems tend to be somewhat unbalanced (Oviatt, 2003), in that efforts have focused on the interpretation of multimodal input, while multimodal output generation has received considerably less attention. In this thesis the focus is on multimodal output generation. While there are detailed models of multimodal communication (e.g., Maybury (2000)) and of the generation of multimodal presentations (André, 2000; André, 2003), the actual output of multimodal systems relies in general on advances in natural language generation (NLG) combined with other visual modalities like gestures. NLG is the task in natural language processing which involves the generation of natural language from a machine representation, such as a knowledge base or a logical form. NLG as it is implemented in most practical systems often employs elementary constructs such as templates (Theune, 2003), which can be used for simple slot filling dialogues, but for more advanced systems generation should be better adapted to the context. Moreover, the generation part of multimodal systems should also provide cognitively-based directions for the combined generation of multiple modalities (Oviatt, 1999). For instance, systems that use Embodied Conversational Agents (ECAs), lifelike characters which present information to the user, need specifications to combine gesture and language that are obviously more sophisticated in that they should mimic human communication very closely to facilitate the interaction (Byron, 2003). The research that is presented in this thesis focuses on two aspects of the need of more advanced multimodal presentations: (1) In what way is the generation of multimodal utterances directed by the context? and (2) Which factors determine what modality or combination of modalities to use in what conditions?

A task that is addressed in many multimodal systems is that of identifying a certain object in a visual context accessible to both user and system. This can be done for example by blinking or highlighting the object, or by using an ECA that points to the object, possibly in combination with a linguistic referring expression. Especially in situations where a purely linguistic description would be very complex, for example when talking about a domain with many similar objects, highlighting or a pointing gesture may be the most efficient way to single out a target object. Moreover, due to the increased interest in ECAs, researchers have started exploring the possibility of applying NLG to generate spoken language which an ECA can present. Characteristically, this implies the coordinated generation of language and gesture. Figure 1.1 illustrates multimodal reference as occurring in interaction with the SmartKom system (Wahlster et al., 2001) and (Wahlster, 2002; 2003a), where both the user and the system are able to use pointing gestures and speech simultaneously to indicate objects. Figure 1.1 shows a flat screen on which an ECA is displayed that points at a particular object on the screen. At the same time the user also points at an object on the screen. With the design of applications like the SmartKom system, the question arises how such systems should generate descriptions in which linguistic information and gestures are combined, but also how such multimodal referring expressions are produced by humans. In this thesis these questions are addressed.

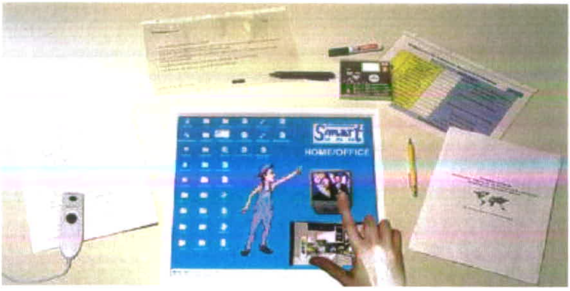


Figure 1.1: Agent and user pointing; interaction with the SmartKom system.

Currently, HCI systems use fairly simple methods for the generation of multimodal referring expressions. The proposed algorithms that generate multimodal referring expressions (e.g., Claassen, 1992; Reithinger, 1992; Huls et al., 1995; Lester et al., 1999) are based on the assumption that a pointing gesture is precise and unambiguous and singles out the intended referent. As a consequence, the generated referring expressions tend to be relatively simple; they usually contain no more than a head noun. Moreover, algorithms tend to be based on fairly elementary, context-independent criteria for deciding whether a pointing gesture should be included or not. Overall these algorithms have four aspects in common:

- The algorithms generate referring expressions irrespective of the context in which they are verbalized, both visually and linguistically;
- The algorithms focus on minimal referring expressions (i.e., the shortest descriptions possible to describe a given referent);
- The algorithms produce only precise pointing gestures, i.e., pointing gestures that uniquely identify the target object;
- The algorithms generate a pointing gesture in all cases, independent of the content of the linguistic part of the referring expression.

However, as noted above, to facilitate the communication between the user and system, algorithms should aim at generating referring expressions similar to the ones produced in human communication. When users are able to communicate with a system in the way they are used to do in human-human communication, a quick and successful interaction is expected. In the following discussion, three important notions that underly the human production of referring expressions are considered in slightly more detail: salience, effort and certainty.

In human communication, referring expressions which include pointing gestures are rather common (Beun and Cremers, 1998). The context that plays a role in identifying objects in a multimodal environment can basically be split into the discourse context (i.e., what is said) and the perceptive context (i.e., what can be perceived).[1] In general, **salient** objects can be referred to in a concise way. For instance, less linguistic information is needed to identify an object that has been talked about recently, than to identify an object that is not in the discourse context. An object that has a notable property which the other objects in the domain lack can easily be identified in only linguistic terms (Beun and Cremers, 1998). Similarly, an object that is located close to the speaker might be identified just by touch (i.e., by means of a pointing gesture that can unambiguously be interpreted by the hearer). In the situation in which the target is located further away, the speaker can still decide to point to the object, but then some linguistic description might be needed as well, especially if there are more (similar) objects located in the scope of the pointing gesture. An important factor in these cases is the *principle of minimal effort* (Clark and Wilkes-Gibbs, 1986), which states that in cooperative dialogue a speaker tries to minimize both her own and the

[1] See Bunt (1997), Bunt and Black (2000a) and Bunt and Girard (2005) for a more elaborated notion of context

hearer's **effort**. Consequently a speaker's goal is to make identification by the hearer as easy as possible by providing enough but not too much information. At the same time the speaker also wants to minimize her own effort in producing the referring expression. Besides balancing the amount of information, the principle determines the kind of information that is used as well: as suggested above, in some cases a pointing gesture is the optimal way to refer to an object, whereas in others a linguistic description is more appropriate, or a combination of the two. Contrastive to the minimization of effort is the speaker's objective to make sure that the hearer can interpret the referring expression. This notion is formalized in the *principle of distant responsibility* (Clark and Wilkes-Gibbs, 1986), which says that a speaker must be **certain** that the information provided in an utterance is understandable for the hearer. Correspondingly, especially in domains with many similar objects, or with objects that do not have easily perceptible features, the speaker might be tempted to overspecify a referring expression or use a very precise pointing gesture, in order to gain certainty of correct identification by the hearer.

To summarize, when considering the production of referring expressions in human communication in more detail, the following observations can be made:

- Speakers produce referring expressions dependent on the context, e.g., speakers tend to refer to objects that have already been mentioned in an abbreviated form (Grosz and Sidner, 1986; Hajičová, 1993) and speakers use salient features to identify an object (Beun and Cremers, 1998);

- Speakers tend to overspecify their referring expressions, i.e., rather than using minimal descriptions, they often provide more information than necessary to indicate the target (Arts, 2004; Maes et al., 2004; Pechmann, 1989);

- Speakers not only use precise pointing gestures, they also produce underspecified pointing gestures to indicate objects that are located at a certain distance (Kranstedt et al., 2005);

- Instead of using gestures and speech separately, speakers integrate their use of pointing gestures and linguistic material in a compositional way (Lücking et al., 2004; Hintikka, 1998; ter Meulen, 1994; Mc Neill, 1992).

In this thesis these observations are taken as a starting point in the development of a more advanced multimodal algorithm that intends to provide natural communication between the user and HCI systems. As a result, the algorithm proposed in this thesis generates possibly overspecified referring expressions that may include various kinds of pointing gestures, in which the linguistic material and the pointing gestures are combined in a compositional way with respect to the linguistic and visual context.

1.2 Generating Multimodal Referring Expressions

The model for pointing that will be proposed in this thesis provides for a close coupling between linguistic information and pointing gestures used. The algorithm in which this model will be formalized generates various pointing gestures, precise and imprecise ones. The type of pointing gesture is closely linked to the perceptual context in that the scope of an imprecise pointing gesture contains more objects than the scope of a precise pointing gesture. This proposition is modeled as illustrated in Figure 1.2, where a pointing gesture is likened to the cone of a flashlight. If one holds a flashlight just above a surface, it covers only a small area (the target object). Moving the flashlight away enlarges the cone of light (shining on the target object but probably also on one or more other objects). A direct consequence of this Flashlight Model for pointing is that the amount of linguistic properties required to generate a distinguishing multimodal referring expression is predicted to covary with the kind of pointing gesture used.

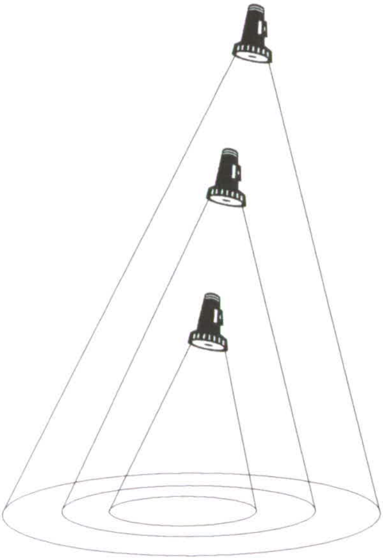


Figure 1.2: Flashlight cones.

The model for pointing will be implemented as a multimodal extension of a new algorithm for the generation of referring expressions. This algorithm, proposed by Krahmer et al. (2003), approaches the generation of referring expressions as a graph construction problem using subgraph isomorphism. It will be shown that the generation of multimodal referring expressions can be facilitated by combining linguistic graphs with gesture graphs. The decision to point is made on the basis of cost functions which are grounded in Fitts' law (Fitts, 1954). Fitts defined a fundamental law about the human motor system, which states that the difficulty of reaching a target is a function of the size of the target and the distance to the target. The output of the algorithm is based on a trade-off between the cost of a pointing gesture and the cost of the linguistic information needed to single out a target object. As such, minimal referring expressions are generated on the basis of a notion of effort, which balances the kind of information that should be presented in order to identify the target at the lowest cost.

The proposed algorithm is in more than one sense context-sensitive. The algorithm generates referring expressions that contain solely linguistic information or that consist of combinations of pointing gestures and linguistic information, based on a three-dimensional notion of salience, which acknowledges the linguistic and the perceptual context. On the one hand, to determine the linguistic context, the discourse history with a notion of recency is taken into account. On the other hand, the perceptual context is determined by two factors: (1) the inherent salience of certain objects, that stand out because they have a particular property that is not present in the rest of the domain; and (2) the visual focus of attention, which centers around the last mentioned target in the discourse, where the scope of possibly generated pointing gestures is incorporated as well. By integrating such a multimodal notion of salience, the algorithm is capable of determining the context in which a target is to be identified very precisely. This leads to the generation of adequate referring expressions, in other words, more concise referring expressions can be generated when the target has already been mentioned in the discourse and locative expressions can be used that describe the target in terms of its relation with another salient object.

Evaluation of this kind of NLG algorithms is difficult, because in linguistic corpora, the objects and their properties that are referred to are not known. Evaluation of multimodal referring expressions is even harder, because multimodal corpora are scarce and the basis on which speakers decide which modality to use is concealed. In this thesis it will be shown that these problems can be circumvented by using production experiments in which participants identify items by speech and gesture. In this way, spontaneous multimodal data is gathered on controlled input. This thesis will present a report of two studies in which participants refer to objects that differ in shape, size and color. One study has a very strict setting; pointing is forced and no feedback is given. The other study is performed in a more natural and interactive setting. The participants in the two studies are divided into two groups: one group located close to the object domain (i.e., the subjects can touch the targets by using precise pointing gestures) and one group located further away (i.e., the subjects can only use pointing gestures that vaguely indicate the location of the target). A detailed analysis of the multimodal referring expressions resulting from these studies is used to evaluate the output of the multimodal algorithm.

The multimodal algorithm that so far only generates minimal referring expressions is revised in this thesis in order to generate overspecified referring expressions. A detailed survey of both unimodal and multimodal overspecification has been carried out with respect to the data resulting from the production experiments as well as findings in cognitive linguistics. Two questions are considered: (1) Why and when do speakers overspecify? and (2) How do speakers overspecify? In correspondence with the answers to these questions, the algorithm will be adapted in such a way that overspecified referring expressions can be generated on the basis of an estimation of the likelihood that a user will be able to correctly interpret the referring expression in the current context. Both the pointing gestures and the linguistic information that can be included in a referring expression are enriched with certainty scores that estimate their effect on the referring expression as a whole in terms of certainty. The degree of overspecification necessary in any particular situation is based on discourse and context factors. As a result the algorithm selects linguistic information and pointing gestures by balancing their costs and certainty scores, in order to find the referring expression that satisfies the responsibility to make sure that the user can identify the target at the lowest cost.

1.3 Overview

This thesis is structured as follows. Chapter 2 will discuss the background for the research reported on in this thesis. From a broad perspective on the field of HCI the scope of this chapter is narrowed down from multimodal interaction, dialogue systems, aspects of NLG and of multimodal presentations, and finally to the generation of multimodal referring expressions both by humans and by machines. Chapter 3 will provide the background of the multimodal algorithm proposed in this thesis. The chapter gives a critical discussion of earlier algorithms for the generation of referring expressions. Comparisons between the algorithms are facilitated by means of a uniform presentation format. The focus in the discussion is on the context-sensitive generation of referring expressions, which includes a new proposal for a three-dimensional notion of salience. This notion incorporates linguistic salience, inherent salience and a demarcation of the focus of attention. In Chapter 4 the new model for pointing will be introduced, together with a detailed description of the graph-based algorithm in which it is implemented. The algorithm uses Fitts' law as a measure of effort to determine when to generate a pointing gesture. The notion of salience presented in Chapter 3 is included in the algorithm to account for context-sensitive descriptions. The workings of the algorithm are illustrated with extensive worked examples. In Chapter 5 the empirical studies conducted to evaluate the multimodal algorithm will be presented. The linguistic referring expressions and the gestures the participants produced to indicate the targets are analyzed and the results for various linguistic and gestural features are reported. Chapter 6 will address overspecification in multimodal referring expressions. Based on an overview of the work on overspecification in (cognitive) linguistics and a detailed analysis of the experiment data from Chapter 5, an algorithm that generates overspecified multimodal referring expressions is proposed and evaluated. Finally in Chapter 7 a thorough discussion will be given of the most interesting aspects in this thesis as well as objectives to be pursued in future work.

Chapter 2

Multimodal Language Generation

2.1 Introduction

This chapter presents the background for the research reported in the following chapters. Section 2.2 starts with a general introduction in the field of multimodal dialogue systems, in which it is discussed what multimodal dialogue systems are, why these systems are interesting and how they work. From this general view the focus is narrowed to the generation side of multimodal systems. Section 2.3 focusses on the presentation of the different modalities in a multimodal environment. Firstly, an architecture for the generation of natural language is presented. Secondly, the generation of multimodal presentations is discussed. Then the attention is further restricted to the generation of multimodal referring expressions. Section 2.4 concerns the generation of multimodal object descriptions in human-human communication. In Section 2.5 a brief overview is given of existing algorithms for the generation of multimodal descriptions. Section 2.6 concludes this chapter with a discussion.

2.2 Multimodal Interaction

2.2.1 Multimodality in HCI

In the field of human computer interaction (HCI) there has been an increased interest in multimodal systems. **Multimodal systems** are systems that allow combinations of two or more modalities to communicate with the user, both on the input and the output side (c.f., Gibbon et al., 2000). The term modality is used in different ways by different researchers. For example, (André, 2003) uses the term modality for the input and the term media for the output of multimodal systems, whereas Maybury and Lee (2000) define modality, or mode, in relation to the human senses that process for instance visual, auditory and tactile information, while the term media is reserved for the means of communication, for example natural language or graphics. In this thesis Beun and Bunt (2001) are followed in their definitions of modality and media. Beun and Bunt use **modality** to denote the form in which the information is presented, like spoken or written language and gestures. The term **media** is then saved for the channels and carriers of information like the human perceptual channels or video or audio streams etc.

There are several reasons for the interest in multimodal systems. One reason is that human communication is inherently multimodal (e.g., Duncan, 1972; Heritage, 1984), it always involves some combination of sight, hearing and touch (e.g., Goodwin, 1981; Mc Neill, 1992; Sacks, 1992). Gestures appear in human communication very often; Mc Neill et al. (2002) even argue that gestures are part of the cognitive processes involved in communication. For instance, when looking at situations in which people do not know how to express themselves using speech, they appear to use more gestures (Butterworth and Hadar, 1989; Kraus et al., 1991). From a technological point of view, systems that combine several modalities are believed to be more suitable for more demanding applications. Multimodal systems are expected to be more robust, because the different modalities can complement each other in communication with the user. Another important reason for the interest in multimodal dialogue systems is that these systems are believed to be easier and more efficient to use. Users should be able to interact more naturally with multimodal systems, precisely because human-human communication is by nature multimodal. Experimental studies reveal that users accomplish their tasks in a multimodal environment faster and with less errors (e.g., Oviatt and Cohen, 1991; Oviatt et al., 1997; Cohen et al., 1998). Finally, it is expected that multimodal systems may be helpful to people with disabilities (e.g., Baljko, 2001a; Baljko, 2001b; Edwards, 2002).

In the design of multimodal systems, it is not beneficial to add just modalities. Instead, multimodality should be adjusted to human cognitive and perceptual processing (e.g., Bunt, 1998). Accordingly, with the advance of multimodal systems challenging issues arise like (1) When to interact uni- or multimodally: users do not interact multimodally all the time (Oviatt, 1997); (2) Which modality to use: which modality is accessible or the most suitable in which situation (c.f., Oviatt and Cohen, 1991; Cohen and Oviatt, 1995); and (3) How to integrate the different modalities: which part of the content should be transmitted with what modality at what time, (c.f., André and Rist, 1996; Gaiffe et al., 2000). To be able to answer the above mentioned research issues appropriately, it is important to collect data about how people synchronize and fuse spoken information with gestural information concerning content and timing (c.f., Levinson, 1983, chapter 6, on the need for empirical studies). This data can be collected by observing human-human conversation, or by setting up experiments in which people perform certain tasks in HCI. Another way to collect data is to let computers mimic human discourse, for instance with the use of embodied conversational agents and improve the computer output based on user evaluation. In the experiments conducted so far, it appears that the combined usage of speech and gesture puts new constraints on the interpretation and generation modules in multimodal spoken dialogue systems. Oviatt (1999) points out, for instance, that the spoken part of multimodal language tends to be simpler than unimodal language. Furthermore, in multimodal expressions, the different modalities do not always overlap in content and often do not co-occur simultaneously in time.

Multimodal systems come in various types; a historic overview of multimodal system design is given by Oviatt (2003). In this thesis the focus is on multimodal dialogue systems (i.e., multimodal systems with language as one of the input and output modalities) as a subgroup of multimodal systems. On the input side, a number of multimodal systems allow the user to single out a target object in a visual interface using gestures (touch pointing) accompanied with speech (as in the SmartKom system, e.g., Wahlster, 2003a). Examples of multimodal systems that combine gestures and linguistic output are applications that involve embodied conversational agents (ECAs) (Cassell et al., 2000) or systems that use language in combination with the highlighting of objects like the DenK system (Ahn et al., 1995; Bunt et al., 1998) or the MATIS project (Soudzilovskaia and Jansen, 2001) and the LIVE system (Kelleher and van Genabith, 2003; Kelleher et al., 2005). In the next section multimodal dialogue systems as an instance of multimodal systems are inspected in more detail.

2.2.2 Multimodal Dialogue Systems

With the recent and fast development of multimodal systems, there has been an increased interest in multimodal dialogue systems as a subgroup of such systems. The goal of a dialogue system is to listen to and understand a typed or spoken user request and to generate a suitable response. Multimodal dialogue systems process information from different types of input and output modalities in parallel. Because of the need for parallel processing of different modalities, multimodal dialogue systems usually make use of multi-agent architectures. Multi-agent systems like for example the Open Agent Architecture (Cohen et al., 1994; Martin et al., 1999) and the Adaptive Agent Architecture (Kumar and Cohen, 2000), provide a flexible infrastructure for the different information flows employed by multimodal dialogue systems. With Multi-agent architectures the different tasks in processing the multimodal input and output are coordinated by the Facilitator. The Facilitator is an interface that routes the different tasks and subtasks to the appropriate modules in a distributed fashion, (c.f., the Hub module in the DARPA architecture as presented by Levin et al., 2000).

In Figure 2.1 a general architecture of a multimodal dialogue system is presented (others exist as well).

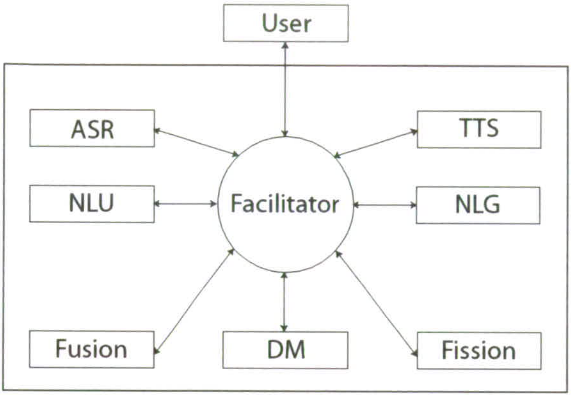


Figure 2.1: Architecture of a multimodal dialogue system.

A multimodal dialogue system can roughly be split up into three parts: (1) The input side focussing on understanding and interpretation of the user input, which can be typified by **hypothesis management** (i.e., selecting the most suitable interpretation for given input); (2) The output side addressing language generation, which can be characterized as a **process of choice** (i.e., what to respond and how to formulate it, given the available means), following the terminology of (Mc Donald, 1992); and (3) Dialogue management taking care of the coordination between the input and output of the system. Starting with the input side, the user input in a multimodal dialogue system consists of language (spoken or written) in combination with in most cases one other modality like touch (i.e., pointing gestures on a touch screen) or pen input or face and gesture recognition etc. In the case of a spoken dialogue system as depicted in Figure 2.1, the speech input of the user is dealt with by the automatic speech recognition module (ASR). The ASR module converts speech into word hypotheses, often in the form of an N-best list or a wordgraph. The strings of words resulting from ASR are taken as input for the natural language understanding module (NLU), which takes care of linguistic processing. Now the Fusion module combines the results of NLU with the data coming in from the other modalities. Within a multimodal dialogue system architecture there are two ways in which the different modalities can be integrated, early fusion and late fusion (Oviatt, 2003). With **early fusion** the modalities are integrated at the feature level, which is suitable for modalities that display a strong temporal connection such as speech and facial expressions or gestures. In contrast, **late fusion** integrates the modalities at the semantic level. Late fusion is therefore applicable to modalities that contain complementary information that is not strictly temporally bound, like speech and pen input. Systems that use late fusion can consequently apply unimodal recognizers in NLU. The fused input is interpreted by the dialogue management module (DM) considering the semantic content, the dialogue act and dialogue history.[1] The DM module handles the **communicative goal**; it computes a response which is accurate and cooperative in the current dialogue context and adapted to both the user and the current intentions and beliefs of the system. Thus, the dialogue manager determines what to respond. On the architecture's output side, the realization of the DM response is handled by the Fission module. The Fission module splits and synchronizes the response according to modality, speech or other. For instance with a plan-based approach for communication as suggested for example by Maybury (2000), the output modalities can be chosen with respect to the nature of the content of the response, (c.f., Vernier and Nigay, 2000). Analogous to the process of fusion, fission can be either early or late. With **early fission** the different modalities are combined at the semantic level, which is suitable for modalities that present complementary information. For example object highlighting in combination with corresponding linguistic object descriptions. With **late fission** the modalities are integrated at the feature level, which may result for instance in more adequate speech and gesture correlations to be presented by embodied conversational agents. In both cases of fission the different modalities are time stamped to provide for synchronized output. The natural language generation module (NLG) generates the text for the speech output. The text to speech module (TTS) produces the speech that matches the words and their mark up. This thesis focusses on the output side: multimodal information presentation. Section 2.3 discusses natural language generation and the generation of multimodal presentations.

2.3 Multimodal Output

2.3.1 Natural Language Generation

Natural language generation (NLG), in general, is the process of converting a communicative act (i.e., as produced by a dialogue manager) into natural language (Dale and Reiter, 2000; van Linden, 2000; Evans et al., 2002; Bateman and Zock, 2003). Stent (1998) formulates NLG as a knowledge-intensive, goal-driven process, which should address the following issues:

[1] See Bunt and Romary(2002; 2004) and Landragin et al. (2004) for formal multimodal meaning representation for multimodal systems. See also the work on the repository of dialogue act definitions as currently undertaken by the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information, instigated by the international standards organization ISO. See http://let.uvt.nl/research/ti/sigsem/wg and http://www.tc37sc4.org

- Content determination addressing the communicative goal of the system;
- Content presentation in accordance with the discourse context;
- Modality choice adapted to content;
- Output suitable for specific users.

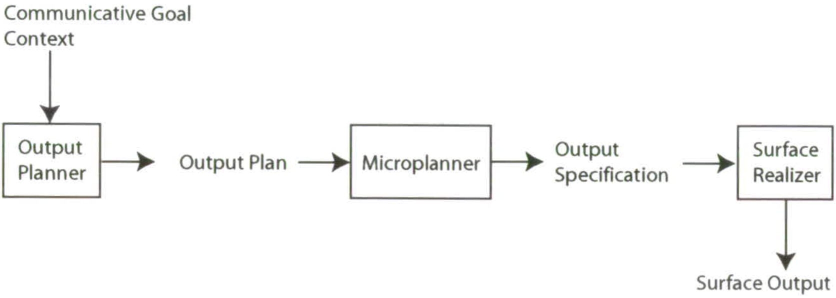


Figure 2.2: NLG System Architecture.

This section focusses on NLG as discussed by Dale and Reiter (2000). Dale and Reiter introduce a pipelined architecture for text-based NLG systems. This architecture is adapted to dialogue systems in general as depicted in Figure 2.2. The architecture distinguishes three modules that carry out different tasks. The first is the Output Planner, which is provided with a Communicative Goal and its context. As indicated in Section 2.2.2 the Dialogue Manager provides this goal as an accurate and cooperative response with respect to the context, user and application. To reach this goal, the Output Planner executes two subtasks: (1) It selects the information that should be communicated (content determination); and (2) It decides how the content should be organized (content structuring). This process results in an Output Plan, which is sent to the Microplanner. The Microplanner transforms the Output Plan into a detailed Output Specification by carrying out three subtasks: (1) It decides on the linguistic structures and their ordering, which are the most suitable to present the content (aggregation); (2) It generates the expressions that identify the entities contained in the content (referring expression generation); and (3) It selects the words to express the content (lexicalization) (c.f., Stone et al., 2003 on a uniform approach on microplanning in the SPUD system). The Microplanner outputs an Output Specification, which is turned into actual output by the Surface Realizer. The Surface Realizer covers two types of realization: (1) It enriches the Output Specification with punctuation symbols, takes care of word order and morphological issues etc. (i.e., linguistic realization); and (2) It inserts structuring mark-up symbols that guide the presentation (structure realization). The Surface Realizer at last produces the Surface Output, being the final output of the NLG module.

Most practical spoken dialogue systems use template-based generation (Theune, 2003), where statistical methods might be employed for output planning (e.g., Bangalore and Rambow, 2000a; Bangalore and Rambow, 2000b; Oh and Rudnicky, 2000; Walker, 2000). In principle, such techniques can be as advanced as real NLG (see, van Deemter et al., 2005), but often templates are rather simple due to the limited output capacities of current systems. The demand for more advanced generation methods as for example suggested by Galley et al. (2001), is likely to increase with the development of more complex dialogue systems, as observed by Oviatt (2003). More complex systems ask for improved output techniques that use natural language and also other modalities. In the next section the generation of multimodal presentations is discussed as an extension to the architecture for NLG presented here.

2.3.2 Multimodal Presentations

In this section the processes and planning that play a role in the generation of multimodal presentations are briefly presented as described by André (2000; 2003). The architecture for multimodal presentation systems suggested by André is presented in Figure 2.3. The approach taken to generate multimodal presentations is similar to the architecture for NLG presented in Section 2.3.1. The main difference is that all modules are now handling multiple modalities. In the architecture, all modules are connected to a knowledge base which is familiar with the application, user, context and design. The architecture consists of a knowledge base and five layers that are responsible for the tasks and processes involved in the generation of multimodal presentations. In the following discussion, the functions of these components are described.

The task of the Control layer is to direct the presentation process in conformance with the presentation goals. The Content Layer covers content selection, content structuring and modality allocation. The output of the Content Layer specifies design tasks for the different modalities together with their underlying relations. The Design Layer consists of Microplanners for each of the modalities, that convert the tasks provided by the Content Layer into specified output plans while considering temporal and spatial coordination. The Realization Layer encodes the information per modality into specific surface presentations. The Presentation Display Layer sends the output of the Realization Layer to the appropriate output media in a time-coordinated manner. Finally, the Knowledge Base contains the information about the application, user, context and design that is necessary to the presentation process.

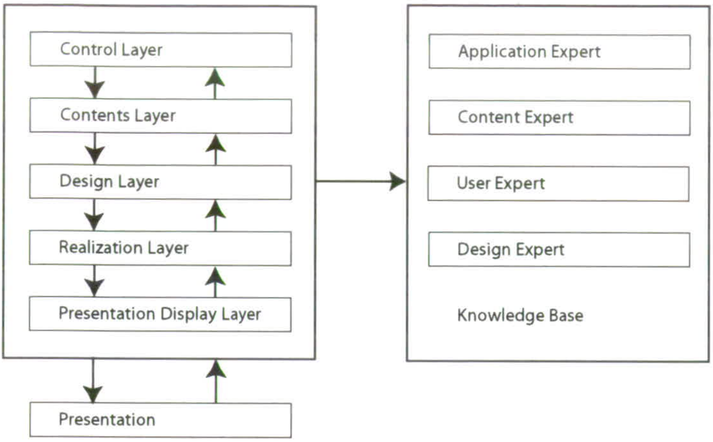


Figure 2.3: Multimodal NLG System Architecture according to Andre(2003).

The integration of more than one modality as carried out by the Fission module in a multimodal system as presented in Section 2.2.2, covers three subtasks: (1) The selection and organization of information; (2) The allocation of the different modalities; and (3) The content-specific modality encoding. This thesis is mainly concerned with **modality allocation**. André (2000) characterizes modality allocation as follows: Given an Output Plan and a set of output modalities, find a combination of modalities that conveys the communicative goal adequately in the current context. The factors to respect in this process are, consequently, the nature of the content and the nature of the modalities, the communicative goal, the user model, the task to be performed and the application itself. With respect to modality allocation, André (2000), Maybury and Lee (2000) and Oviatt et al. (2003), among others, advocate that the integration of different modalities should happen dynamically, instead of considering all modalities individually with respect to appropriateness in composing a multimodal expression. Consequently, the integration of different modalities into a multimodal expression should be based on a theory of communication as a whole. Maybury(1993; 2000), formalizes communication as several related classes of action which cover Physical, Linguistic and Graphical Acts, that are all considered multifunctional and context dependent. In the taxonomy proposed by Maybury, Physical Acts are divided into three groups: (1) Deictic, like pointing or circling; (2) Attentional, like snapping fingers or clapping hands; and (3) Body language, like facial expressions or gestures. Allwood(2002; 2002) discusses bodily language and its place in human communication. Using the terminology of Searle (1969) and Appelt and March (1982), Maybury (2000) splits the Linguistic Acts into (1) Referential or attentional acts, like 'the large block' or 'wake up!' ; (2) Illocutionary acts, addressing the communicative function, like inform or request; and (3) Locutionary, as surface speech acts like asking for information or commanding an action to be performed. Maybury (2000) considers dialogue acts as a special case of Linguistic Acts, because of their context dependency, (c.f., Bunt 1997, 2000a; Bunt and Black, 2000b; Beun, 2001; Bunt and Girard, 2005 on the role of context in information dialogues). Finally Graphical Acts, using graphical media, are also divided into three groups (1) Deictic or attentional acts, like highlighting, blinking; (2) Display control acts, like zooming or panning; and (3) Depict acts, like depict image, draw or animate action. Since graphics are hard to define compositionally, Maybury and Lee (2000) propose to define their semantics in a way that is partly analogical and partly symbolic. On top of the Physical, Linguistic and Graphical Acts, Maybury (2000) presents the class of Rhetorical Acts (c.f., Rhetorical Structure Theory Mann and Thompson, 1987). The Rhetorical Acts form a medium- and modality-independent level of communication, that can be used to integrate Linguistic and Graphical Acts by considering the content and the effect of these acts in communication.

Currently in multimodal NLG little work has been done on the integration and synchronization of multiple output modalities. Most of it is applied in embodied conversational agents (ECAs) such as REA (e.g., Cassell et al. (2000) Cassell et al. (2000)), which are able to produce context-sensitive speech combined with representational gestures and nonverbal gestures (e.g., beat gestures, gaze and posture). Other examples are the agent Greta (Pelachaud et al., 2002), in which facial gestures are adapted to the linguistic output and the VMC project (e.g., Nijholt and Heylen, 2002; Theune et al., 2005), where an agent provides route descriptions that integrate speech and gestures. Projects that address the choice and integration of output modalities are the ANGELICA project Theune (2001), and the NECA project (c.f., André and Rist, 2000; Krenn et al., 2002). The integration of the various output modalities commonly takes place by first determining the linguistic output and subsequently inserting the gestures at appropriate positions in the verbal output. This results in non-complementary output presentations, which may display unnatural redundancies, for example when a precise pointing gesture is performed to indicate a single object that is at the same time distinguished by an elaborate linguistic referring expression (Theune et al., 2005). In contrast, Theune et al. (2005) propose a general architecture of the generation process in which language and nonverbal signals are combined. This architecture, displayed in Figure 2.4 can be interpreted as a multimodal variant of the architecture for NLG proposed by Dale and Reiter (2000) (see Section 2.3.1, Figure 2.2). The Microplanner's subtasks, the generation of referring expressions and the lexicalization are enriched with respectively the generation of deictic gestures and the generation of representational gestures. The Surface Realizer is extended with discourse structuring signals that include prosody. The architecture lacks backtracking, which has the effect that once a gesture has been added to the output, it cannot be removed. In this respect Theune et al. propose an ordering of the subtasks of the Microplanner, where aggregation precedes the generation of referring expressions, which in turn precedes lexicalization (c.f., Kopp et al., 2004 for a unified approach on language and iconic gesture planning based on the SPUD system). In the subsequent phases of the architecture, gestures can only be added if not in discord with the ones already contained in the output; the deictic gestures have preference over representational gestures, which are again preferred over discourse structuring signals. As such, gestures are composed during the different phases in the generation process. For instance, a deictic gesture that also indicates some characteristic of the referent, like a pointing gestures that includes a circular movement to refer to the round shape of the target is generated as follows: First, while generating a referring expression a pointing gesture is included. Then, on a second note in the lexicalization phase the pointing gesture is enriched with a representational gesture, (i.e., circular movement). In this thesis the architecture as proposed by Theune et al. (2005) is adopted. The remainder of this thesis focusses on the subtask of the Microplanner involving the generation of multimodal referring expressions (i.e., referring expressions that are combined with deictic gestures). This topic is approached in the next sections by an account of how people produce multimodal referring expressions, followed by a discussion of the automatic generation of referring expressions.

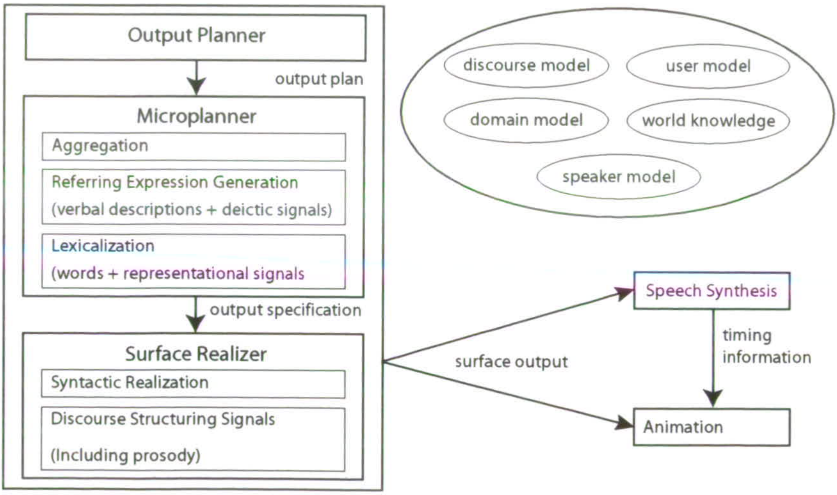


Figure 2.4: Integrated architecture for generation of language and nonverbal signals taken from Theune et al. (2005).

2.4 Human Generation of Multimodal Referring Expressions

This section discusses multimodal referring expressions produced in human communication by first considering the two modes, language and gestures, separately. In Section 2.4.1, aspects that play a role in the production of verbal referring expressions are considered and in Section 2.4.2 deictic gestures in particular pointing gestures are discussed. Finally in Section 2.4.3, how these two modes are to be used together is considered.

2.4.1 Referring Expressions

Referential acts and referring expressions have been extensively studied from various perspectives in linguistics and psychology (e.g., Karttunen, 1976; Clark and Marshall, 1981; Cohen, 1984; Appelt, 1985; Gundel et al., 1993; Wilson, 1992). A **referring expression** distinguishes a referent from the objects in its context by a specification of properties, relations and deictic gestures that provide sufficient information for identification. This section focusses on linguistic referring expressions. In human communication linguistic referring expressions appear in various forms: indefinite noun phrases and definite noun phrases, including proper names and pronouns. In general, indefinite noun phrases are used to refer to objects that have not been mentioned before (i.e., initial reference), whereas definite noun phrases can also be used as a subsequent reference, for instance to refer to objects that have been introduced in a discourse. This thesis focusses on **distinguishing referring expressions**, referring expressions that uniquely single out a referent from the other objects in the domain. This notion is illustrated with the definite noun phrases presented in Figure 2.6, that can be uttered to indicate object d_1 in the simple block domain depicted in Figure 2.5.

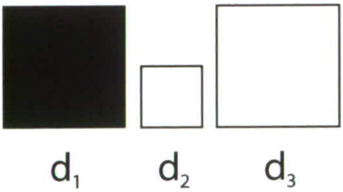


Figure 2.5: Example Domain.

(1) 'the large black block to the left of the white one'
(2) 'the black block'
(3) 'the block'
(4) 'it'

Figure 2.6: Possible realizations for d_1 in Figure 2.5.

Referring expression (1) is a distinguishing referring expression in the form of an extended noun phrase. Since there are no other large black blocks in the domain, the inclusion of the locative information, *to the left of the white one* is superfluous in distinguishing the target. Referring expression (2) is also distinguishing, since *black* is an outstanding property of the target, (i.e., all other blocks in the domain have a different color). Referring expression (3) is not distinguishing, it is only suitable as a subsequent reference. In this case, one can refer to the object in a shortened way using a head noun, but a pronoun like expression (4) is also possible.

In the generation of referring expressions by humans, there are at least four aspects to consider, which are illustrated with the referring expressions in Figure 2.6. The first aspect is the **principle of minimal cooperative effort** (Clark and Wilkes-Gibbs, 1986), which states that the total effort of both speaker and hearer should be minimal. In cooperative dialogue this means that a speaker's goal is to make identification by the hearer as easy as possible by providing enough but not too much information (c.f., Reiter and Mellish, 1992). At the same time the speaker wants to minimize her own effort in producing the referring expression. Balancing the amount of information in this way is closely related to the notion of relevance in the sense of providing maximal information with minimal processing effort (Wilson and Sperber, 1984; Matsui, 1998). With respect to the example above, if object d_1 has been talked about before, expression (1) takes too much effort to produce and to interpret. A second aspect that plays a role in the formulation of a referring expression is the **accessibility** (Ariel, 1991; 2001) of an object in its context. An object is accessible by both speaker and hearer if it has recently been mentioned in the discourse. In these cases a reduced anaphoric description like (2) or (3) or a pronoun like expression (4) is more suitable to indicate d_1. A third aspect to consider is the **salience** of an object (Cremers, 1996), i.e., objects are salient when they stand out in comparison to other objects in the domain of conversation. Objects can be salient for several reasons, for instance an object can be more prominent than others because of a certain property, color or shape (i.e., inherent salience). In the example domain in Figure 2.5 the target is salient because it is the only black block. In general, the identification of salient objects demands less effort, because the group of candidate objects which can cause confusion is relatively smaller. Accordingly, a salient target can be identified with a simpler expression than a non-salient object (Cremers, 1996). Krahmer and Theune (2002) provide a detailed discussion on salience in the linguistic context, which is addressed in more detail in Chapter 3, Section 3.5.1. Finally, a fourth aspect is the **principle of distant responsibility** (Clark and Wilkes-Gibbs, 1986), which states that a speaker or writer must be certain that the information provided in an utterance is understandable for the user. In cooperative dialogue, a speaker may provide more information than necessary to help the hearer in identifying the target; for instance, in cases in which the target is not easy to distinguish, because it has no prominent features, or in cases in which it is very important that the hearer understands the referring expression. As such this principle may lead to overspecified referring expressions, as the one exemplified in description (1). Note that the principle of minimal effort and the principle of distant responsibility can be contrastive in selecting the proper amount of information for identification.

2.4.2 Deictic Gestures

This section focusses on deictic acts as a subgroup of the physical acts in the taxonomy presented in Section 2.3.2. In the classification scheme of Mc Neill (1992), deictic acts, or deictic gestures, are typically pointing gestures. Although an object can also be indicated by placing it in the focus of attention (c.f., Clark, 2003, who investigates deictic gestures in terms of directing attention and the many appearances of indication), this thesis focusses on identification by means of pointing gestures. In contrast to linguistic referring expressions, pointing gestures are generally not used on their own to identify an object. Although there are exceptions (c.f., Haviland (2003), on the complexity of pointing gestures), in most cases people combine a pointing gesture with a referring expression to establish a unique identification of a target object. In the following discussion, three aspects that certainly play a role in such a combined reference are discussed and illustrated with reference to the referring expressions for object d_1 in the example domain in Figure 2.5 presented in the previous section.

As with linguistic referring expressions, an important aspect of identification by means of a pointing gesture is the **accessibility** of an object in its context (e.g., Levelt, 1989). As opposed to linguistic referring expressions that can be used to identify objects that are not currently visible or even abstract, the referent of the pointing gesture has to be located in the perceptual context of both speaker and hearer. In cases where both speaker and hearer can see and maybe even touch the target object, a pointing gesture together with a determiner and maybe a head noun might be sufficient for identification (i.e., expression (3), in combination with a pointing gesture). If a pointing gesture is omitted, distinguishing the target object might need a more elaborate linguistic description, for instance including some adjectival properties of the object, like in expression (1) or (2). Accordingly, as a second aspect, the inclusion of a pointing gesture can be viewed as being closely linked to **cooperativity** as an important factor in human dialogue (Grice, 1975), (c.f., Bunt, 1998; Beun and Bunt, 2001). A deictic pointing gesture may shorten a complex description in case a target is difficult to describe. In the example in Figure 2.5 this would be the case if the large black block is not uniquely characterized by expression (1) (i.e., there are more large black blocks in the domain that are located to the left of white blocks). In such cases, identifying the target with a pointing gesture minimizes the effort of both speaker and hearer. As a third aspect the **precision** of the pointing gesture is considered. Intuitively, a pointing gesture is generated by the speaker as a precise pointing gesture, no matter the distance to the target (i.e., a straight line can be projected to the target). However, interpretation of these pointing gestures by the hearer may get more complicated when the distance to the target becomes larger (i.e., when the pointing gesture can be projected as a cone capturing an area). In return, the speaker might acknowledge this and provide more linguistic information to distinguish the target when the pointing gesture can be interpreted as directed towards the area in which the referent is located among other objects. In this way the speaker makes sure that the hearer can interpret the referring expression correctly. In the same vein, in cases where a pointing gesture that uniquely points out the referent is used, for instance by touching the object, less linguistic information is needed. This intuition is discussed in more detail in Chapter 4.

Pointing gestures are unique to human behavior and almost inevitable in human communication (Kita, 1993; Kendon, 1994). Mc Neill (1992) distinguishes between abstract and concrete pointing gestures. Abstract pointing gestures are directed at objects in a metaphorically used space in front of the speaker, whereas concrete pointing gestures, which are considered in this thesis, are used to indicate objects, locations or directions. The referent of a pointing gesture is not always easy to identify. For instance, if the pointing gesture indicates a direction, the **origo** or reference point (Bühler, 1934; Mc Neill, 1992, page 173-174) determines the interpretation of the gesture (c.f., Haviland, 2003). Instead, if the referent is an object, the pointing gesture can refer to the object itself, some property of the object, or it can just state that the object is located at the indicated position (Clark, 2003). Such pointing gestures can be achieved with various body parts, for example with the head, hand, lips, elbow, or even with a foot (Kendon and Versante, 2003; Kendon, 2004). Pointing gestures produced solely by the hand are already complicated; they appear in multiple forms with different interpretations (c.f., Calbris, 1990; Kendon and Versante, 2003; Wilkins, 2003; Kendon, 2004). The discussion in this thesis is therefore limited to pointing gestures that are performed by a hand with an extended index finger that cause a projection of a straight line from the tip of the index finger to the intended referent. Such pointing gestures may still combine features. For example, a pointing gesture can be used to indicate the location of the referent together with a characteristic of the referent. For instance, some kind of movement can be made while pointing, whereby the shape of an object is indicated as well as its location Kendon (2004). The appearance of pointing gestures can be defined in terms of the gesture hierarchy proposed by Mc Neill et al. (1990), which is copied and presented in Figure 2.7. The top level in the hierarchy proposed by Mc Neill et al. describes the arm and body movements the speaker uses in performing a gesture, while the second level specifies the related head movements. Here, the focus is on the hierarchy below Gesture Unit, which is defined as the time in which the body performs a gesture. Within the Gesture Unit the gesture is described from initiation to finish. Thus, a pointing gesture can be described in terms of the time interval which starts the moment the hand starts to move and which ends as soon as the hand comes to rest again. Within such a Gesture Unit one or more Gesture Phases may occur, which consist each of one or more movement phases. The Preparation phase is optional; here the hand moves to the position where the stroke begins, while anticipating the linguistic part of the communicative action. Then there is an optional Pre-stroke Hold, where the hand stays still until the stroke begins, (but c.f., Kita, 1990). Subsequently there is the obligatory Stroke itself, displaying the maximum of effort within the gesture and expressing the meaning of the gesture. At the end of the stroke the hand may optionally stall briefly in its position, a Post-stroke Hold, before the Retraction, in which the hand returns to its rest position. Retraction phases are omitted when the gesture proceeds right away to the next gesture. Apart from the hierarchy presented here, other hierarchies exist as well (c.f., Kendon, 1972; Kendon, 1980; Schegloff, 1984).

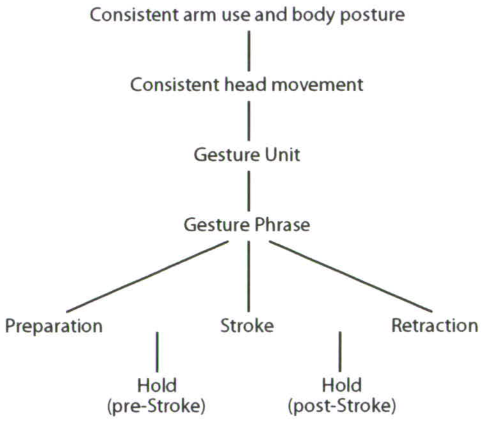


Figure 2.7: Gesture Hierarchy according to Mc Neill(1992, page 82).

2.4.3 Integration of Referring Expressions and Deictic Gestures

In combining pointing gestures with linguistic referring expressions at least three aspects need to be considered: (1) Timing; (2) The appearance of the linguistic expression; and (3) The interaction of the pointing gesture and the linguistic expression in constituting their joint meaning. In this section all three aspects are addressed.

The first aspect, timing, concerns the synchronization of gesture and speech. Mc Neill(1992, page 26) formulates the Phonological synchrony rule which expresses that the stroke of a gesture precedes or ends at, but does not follow, the phonological peak syllable of speech, (c.f., Kendon, 1980). The synchronization rule accounts for the fact that speakers tend to keep a gesture and the accompanying speech close together (c.f., Wachsmuth, 1999 on rhythm of speech and gestures that aid the communication process). In order to model speech and gesture production and to account for their synchronization in time, de Ruiter(1998; 2000) extends the architecture for speech production proposed by Levelt (1989) with a model for the production of gestures. In contrast to another extension of Levelt's model proposed by Krauss et al. (1996), de Ruiter constructs his model on the assumption that both speech and gesture are planned by the same process and derived from the same representation of the communicative intention, (c.f., Mc Neill (1992), for a contrastive approach on this issue). Accordingly, de Ruiter proposes an architecture that permits detailed modeling of the synchronization of speech and gesture. However, since the gestures are planned before the speech, the architecture does not permit any influences from the information contained in the speech on the gestures (c.f., Kita and Özyürek (2003)). In general the synchronization of speech and gesture is hard to define in terms of time intervals. For pointing gestures, however, synchronization with speech seems relatively simple, because pointing gestures are relatively easy to interpret in relation to the accompanying spoken information (c.f., Clark, 2003). Various experimental studies have addressed this issue, so far with mixed results. Empirical evidence by Levelt et al. (1985), Feyereisen (1997) and de Ruiter (1998) reveals that the stroke of pointing gestures directed towards concrete objects is temporally very close to the noun onset. In contrast, the experiments conducted by Kranstedt et al. (2003) and Lücking et al. (2004) do not reveal such a smooth synchronization of pointing gestures and speech.

The second aspect, the appearance of linguistic expressions in combination with pointing gestures, is studied a lot on the input side of multimodal systems. For instance, Oviatt and Kuhn (1998), Oviatt et al. (1997) and Oviatt et al. (1994), report on how users formulate referring expressions using speech and pen input: users tend to use shorter descriptions and less complex spatial descriptions in multimodal interaction. de Angeli et al. (1999) report on experiments in which users identify objects by speech and pen input with respect to the perceptive context. In this study, it appeared that users spontaneously plan their linguistic referring expressions based on their knowledge of the visual context. Furthermore, there is a correlation between the complexity of the linguistic expressions and of the accessibility of the visual information. Simple pointing gestures directed towards a clearly identifiable referent join simple verbalizations, whereas complex verbalizations occur especially if the referent is difficult to identify with a pointing gesture (Wolff et al., 1998). It should be noted that the results of studies on human-computer communication may not be fully representative of the way human speakers provide multimodal referring expressions. The tasks employed in most of these studies appear to be relatively easy and might not cover the complete range and diversity of human language (Kehler et al., 1998). Moreover, several studies reveal that users are very much affected by the language used by a system. For instance, Zoltan-Ford (1991) shows that users adapt the length of their sentences to the sentence length the system uses, and Brennan (1996) argues that users also copy the system's vocabulary. From the experiments conducted by Skantze (2003), that aim to investigate the extent in which multimodal referring expressions produced by users are affected by the multimodal referring expressions used by the system, it can be concluded that in their use of speech and gestures, users copy the way in which the system refers to the objects but not the way in which the system refers to locations (c.f., Bell et al., 2000, for similar results). Consequently, for the generation of multimodal referring expressions to be used by embodied conversational agents, experiments on human-human communication, as presented by Beun and Cremers (1998), may be preferable. Beun and Cremers (1998) performed several tests with Dutch subjects in which one participant (the instructor) had to instruct another participant (the builder) to make certain changes in a block building that was located in a shared workspace. The experiment was set up so that participants could both talk about and point to the blocks in front of them.

The third aspect connected to the proper integration of speech and gestures focusses on how gestures and language are related in constituting their joint meaning. In this thesis a compositional view is followed, in which the meaning of a multimodal referring expression is constructed from both the gestures and the language. This choice can be accounted for by the fact that in human communication, the number of words needed to identify an object in combination with a pointing gesture tends to be much less than the number of words used in a purely linguistic description of the same object, (c.f., Lücking et al., 2004). Also the semantic synchrony rule of (Mc Neill, 1992, page 27-29), which states that co-occurring speech and gestures display the same meaning supports the compositionality of speech and gestures. Furthermore, as investigated by Oviatt et al. (1997), Sharma et al. (2000) and Kettebekov et al. (2002)), prosodic features underline the semantic parallels between speech and gestures as well. Based on empirical findings in different languages, the relationship between the production of gestures and the production of speech is modeled by Kita and Özyürek (2003) as presented in Figure 2.8. In this model, the Communication Planner determines what modalities to use, but not exactly how these modalities should be used (c.f., de Ruiter, 2000; Krauss et al., 2000 for different views). In contrast to existing models that determine the content of a gesture solely on the basis of a communication model (e.g., de Ruiter, 2000; Mc Neill, 1992), Kita and Özyürek propose a model in which the content of a gesture is specified by a general notion; the Action Generator. The Action Generator specifies the content of a gesture as being jointly determined by three factors: (1) The communicative intention as defined by the Communication planner; (2) The spatial features of the context in which the gesture is to be performed; and (3) Direct feedback from the Formulator via the Message Generator. The latter factor, the bi-directional information exchange between the Action Generator and the Formulator, provides the gestural content to be shaped in correspondence with the linguistic content.

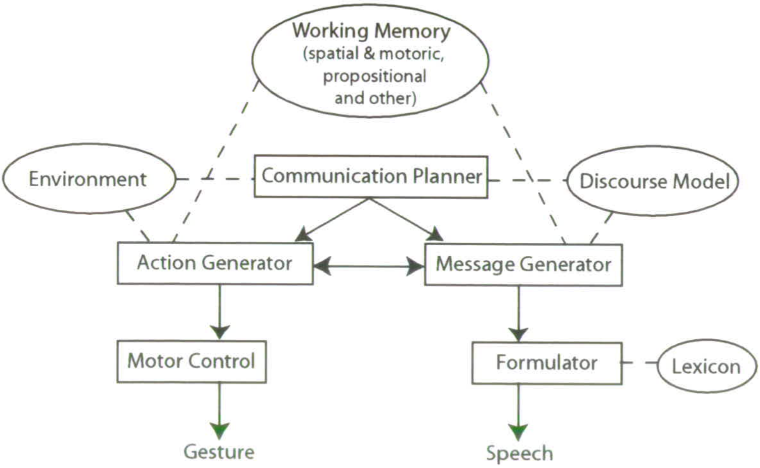


Figure 2.8: Model for human production of speech and gestures proposed by Kita and Özyürek (2003).

A framework that accommodates the cooperative combination of multiple modalities in a compositional way is proposed by Martin et al. (1998). Below this framework is adapted more specifically to multimodal referring expressions. In contrast to frameworks proposed by Hutchins et al. (1986) and Nigay and Coutaz (1993), who use spatial and temporal dimensions to combine multiple modalities, Martin et al. define five types of cooperation between modalities: (1) Transfer, in cases where one modality uses the information provided by another modality (i.e., the blinking or highlighting of objects, that are talked about); (2) Equivalence, when the same information can be generated by more modalities (i.e., the utterance 'this block' or a precise pointing gesture); (3) Specialization, when information can only be generated by one particular modality (i.e., a pointing gesture specifically demarcates an area, while the accompanying speech clarifies the type of object); (4) Complementarity, in case information can be split in segments, which are generated by separate modalities (i.e., an utterance like 'shall we move this?' while producing a pointing gesture directed at an object during pronunciation of 'this'; and (5) Redundancy, when information presented by different modalities overlaps, which is not always easy to differentiate from complementarity (i.e., circular movement of a pointing finger while uttering *round* as a property of the target). The user studies conducted by Gupta and Anastasakos (2004) provide similar observations on semantic integration patterns of multiple modalities. To be able to represent these kinds of cooperation for pointing gestures combined with linguistic referring expressions, a semantic representation of pointing gestures is needed. Until recently, however, there did not exist a compositional semantic representation scheme for pointing gestures. The first suggestion in this direction is proposed by Rieser (2004), who presents a type logic account for pointing gestures integrated in the framework of Logical Description Grammar, (c.f., Muskens, 2001). The interface Rieser proposes accommodates pointing gestures directed at both objects and locations, which possibly occur at different positions within the linguistic descriptions.

2.5 Automatic Generation of Multimodal Referring Expressions

In this section the automatic generation of multimodal referring expressions is discussed. Section 2.5.1 considers what type of referring expressions a multimodal system should be able to produce. Section 2.5.2 discusses various algorithms that generate multimodal referring expressions which combine linguistic referring expressions with pointing gestures. Their similarities and differences are discussed in Section 2.5.3.

2.5.1 Referring Expressions in Multimodal Contexts

In the model of multimodal communication presented in Section 2.3.2, two types of deictic acts are identified. One type is identified as a Physical Act, like pointing gestures, which are discussed in detail in Section 2.4.2. The other type of deictic acts addresses the blinking or highlighting of objects and falls thereby in the scope of Graphical Acts. In contrast to humans, multimodal systems can use both types. Algorithms that generate physical deictic gestures can be used in for example the design of embodied conversational agents (e.g., Cassell et al., 1994; Rickel and Johnson, 1999; Lester et al., 1999; Cassell et al., 2000; Theune, 2001; Sowa and Wachsmuth, 2001; Kopp and Wachsmuth, 2002; Jörding and Wachsmuth, 2002;

Wahlster, 2002; Theune et al., 2005). Algorithms that generate graphical deictic gestures are implemented in systems like for instance EDWARD (Bos, 1993), DenK (Ahn et al., 1995), CHAMELEON (Mc Kevitt, 1998), SmartKom (Wahlster et al., 2001) and Matis (Soudzilovskaia and Jansen, 2001). While focussing on algorithms for the automatic generation of multimodal referring expressions both types of deictic gestures are considered together with the accompanying linguistic expressions.

Apart from the fact that the deictic gestures contained in referring expressions can be both graphical and physical in nature, there is also a difference in the way the referring expressions can be used in human communication versus computer applications, in other words, in multimodal dialogue systems, referring expressions can also provide for coreferential links between the different media employed. André (2000) differentiates between three types of referring expressions occurring in multimodal discourse (c.f.,Byron, 2003): (1) Evoking multimodal referring expressions, that refer to world objects for the first time in a discourse by a combination of at least two modalities, for instance natural language expressions combined with pointing gestures; (2) Exophoric or cross-media referring expressions, that refer to objects in the perceptive context, for instance to other presentation media like 'in the upper left part of this figure'. In most cases these referring expressions serve to direct the attention to the intended referent; (3) Anaphoric referring expressions, that refer to world objects in an abbreviated form; the referents are already introduced (i.e., the referring expression has an antecedent). As an extra complication, the modalities in which the anaphora and the antecedents are communicated are not necessarily the same in multimodal communication. As an example consider 'the figure you have just been shown', where the linguistic expression has a graphical antecedent. For the three kinds of referring expressions, it should be decided whether to use a pronoun or a noun phrase (including proper names). As discussed in Section 2.4.1, there are four important factors on which this decision should be based: (1) The principle of cooperative effort; (2) Accessibility; (3) Salience; and (4) The principle of distant responsibility. This thesis focusses on the generation of noun phrases. For an account of the generation of pronouns see Dale and Reiter (2000, page 149-151).

To generate automatically a noun phrase, properties and relations have to be selected with which the intended referent can be identified by the user. With respect to cooperativity, accessibility, salience and distant responsibility, the generation of referring expressions means balancing between ambiguity and redundancy; to provide enough but not too much information to the hearer in order to be able to identify the target object, (i.e., to provide a cooperative and relevant description). Two aspects that should be taken into account in this process are the perceptive context and the discourse context. The discourse context represents all the objects in the domain that are in the current focus of attention of the hearer (Grosz and Sidner, 1986). The perceptive context is much more difficult to model. As indicated in Section 2.4.1, the perceptive context can be modeled by using the notion of accessibility. As such, the most extensive view on the perceptive context results in a set that contains all objects that are in the current view of both speaker and hearer. However, recent studies may lead to more specific characterizations of the perceptual context (c.f., Thorisson, 1994; Wolff et al., 1998; de Angeli et al., 1999; Landragin et al., 2001; Kelleher and van Genabith, 2003; Kelleher et al., 2005).

Thorisson (1994) proposes a gestalt-based measurement for perceptual grouping by taking into account the proximity and the similarity of different objects in a multi-dimensional space. Landragin et al. (2001) use this notion of perceptual grouping to construct a semantic representation of the visual context. Within this framework Landragin et al. integrate an extensive notion of visual salience. Visual salience in their approach is composed by four factors that are ordered according to preference: category, functionality, physical characteristics and orientational aspects. In this composition the factor 'category' denotes the *shape* of the object, which seems a rather domain dependent factor that might be reconsidered as a physical characteristic. Moreover, empirical findings by Maes et al. (2004) signal that functional aspects are not commonly used in objects descriptions. Physical characteristics and orientational aspects, however, seem usable factors in modeling visual salience as also advocated by Kelleher et al. (2005) (c.f., Kelleher and van Genabith, 2003).

Kelleher et al. (2005) present a different approach for modeling visual salience based on the relationship of focus of attention and physiological aspects of human visual perception. Salience is measured for each visible object in a virtual reality system depending on its color and size with the use of the graphical method called false coloring (c.f., Noser et al., 1995). The focus of attention can be seized from the visual information presented to the user by ranking the objects according to their salience. Kelleher et al. (2005) combine this visual salience measure with linguistic salience, in order to handle all three types of referring expressions as mentioned above: evoking, exophoric and anaphoric. Accordingly, Kelleher et al. formulate a context model that is updated in case of changes in both the linguistic and the visual context considering a notion of recency.

Another model in which the discourse context and the perceptual context are combined is proposed by Salmon-Alt and Romary (2000). The model integrates information transmitted by various channels. The context is defined as a subset of all objects in the domain that are dependent on the discourse, gestures and the perceptual environment. To determine the objects situated in the focus of attention, notions of linguistic salience proposed by Hajičová (1993) and Grosz and Sidner (1986) are applied (in a similar way as done for linguistic expressions by Krahmer and Theune, 1998; 2002). The context model uses the notion of perceptual grouping proposed by Thorisson (1994) to compute the salient objects with respect to the perceptual environment. Furthermore, the model takes into account whether an object has been manipulated recently. The question of how the integration of such sources of information may be used for the actual generation of multimodal descriptions is not addressed.

2.5.2 Approaches

Various algorithms for generating multimodal referring expressions have been proposed (e.g., Reithinger, 1992; Claassen, 1992; André and Rist, 1996; Lester et al., 1999). These algorithms all operate on domains which are in the direct visual field of both speaker and hearer. Throughout this thesis this assumption is made as well. This section presents a concise overview of the existing algorithms for the automatic generation of multimodal referring expressions that include deictic, graphical, or both types of gestures. The algorithms are summarized in a way that facilitates their comparison on several linguistic and gestural features.

Referring Expressions that include Graphical Gestures

The domain which is applied in the CUBRICON system (Neal and Shapiro, 1988, 1991; Neal et al., 1998) concerns the planning of tactical Air Force missions which is communicated via various modalities including maps, tables, forms, printed text and spoken language that may contain graphical and deictic gestures. A screen shot of the CUBRICON system is displayed in Figure 2.9. The system uses a user model, a discourse model, a knowledge base and a unified multimodal language for the communication between user and system. To be able to process different modalities simultaneously, like humans do, input and output are handled as compound data streams that combine the units corresponding to the various modalities. The discourse model of the CUBRICON system uses a notion of attentional focus space (Grosz and Sidner, 1986) to keep track of the objects that have been expressed in the discourse as well as of the objects that are currently visible on the screen. In contrast to, for example, the XTRA system (Allgayer et al., 1989), CUBRICON allows for the output of multiple multimodal referring expressions per utterance. The targets of these expressions are always referred to by blinking or highlighting combined with a noun phrase. In case a target is not visible, CUBRICON displays a window that contains a graphical representation of the target. The noun phrases

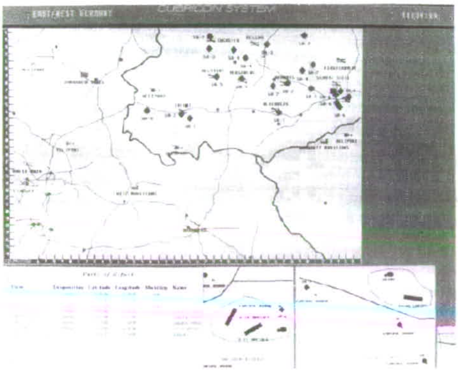


Figure 2.9: Screen shot of the CUBRICON system taken from Neal et al. (1998).

that accompany the highlighting of the target contain a demonstrative determiner and a proper name or a class name that indicates the object.

The DenK system is a generic architecture, which is applied to offer help in using an electron microscope as displayed in Figure 2.10 (a)[2]. The interface of the DenK system consists of several windows, among them one in which the dialogue takes place and one with a graphical representation of the microscope. In contrast to both the CUBRICON system and the CHAMELEON system, the DenK system (Ahn et al., 1995; Bunt et al., 1998; Kievit et al., 2001) employs a two way interaction between user, system and application domain which allows communication between the user and system through both linguistic communication and graphical operations. The user can ask questions like 'what is this?' while clicking on an object on the screen, give orders like 'increase magnification', to which the system responds with an alteration of the rays and the microscope. The system identifies objects by generating referring expressions or highlighting the objects, similar to the CUBRICON system. While much effort has been put into the interpretation of multimodal input, on the output side of the DenK system, the different modalities have not been integrated. At a user's request to show for example the C2-lens, the system answers by highlighting the specified lens without further linguistic explanation. This might lead to confusion in cases where the user is not paying attention to the window where the microscope is depicted.

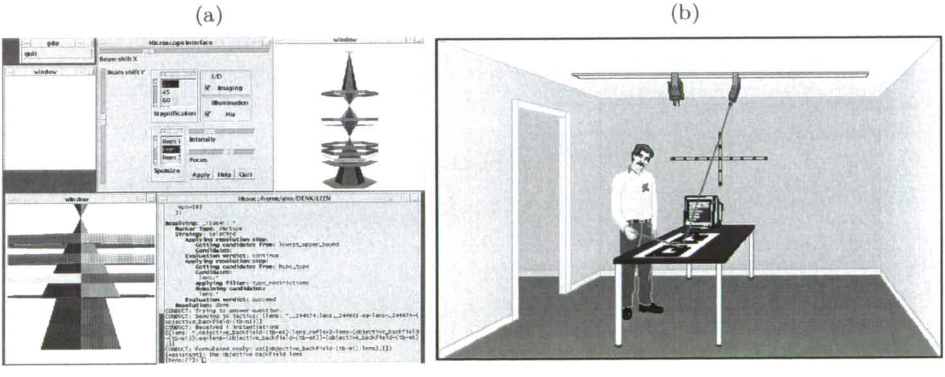


Figure 2.10: (a) Screen shot of the DenK system and (b) the CHAMELEON workbench taken from Brøndsted et al. (1999).

Like CUBRICON and DenK, CHAMELEON (Mc Kevitt, 1998; Brøndsted, 1999; Brøndsted et al., 1999) is a generic architecture that can accommodate various applications, for instance a building information system that provides information about tenants, locations of offices and routes to offices within a building. As shown

[2] Provided by Paul Piwek

in Figure 2.10 (b), CHAMELEON uses a building plan situated on a physical table, on which the system can point out locations or draw routes with a laser beam while answering the user in spoken natural language. The system's answers are generated using templates which are completed with variables selected from a finite list that contains names of objects and persons. Accordingly, simple references are generated that always contain a noun phrase and a deictic gesture produced with a laser beam. In case of requested route descriptions the laser beam moves over the building plan to the location that is asked for.

Referring Expressions that include Deictic Gestures

André and Rist (1996) propose an algorithm for the generation of temporally coordinated multimodal referring expressions, used in a plan-based presentation system named PPP that generates instructions for maintenance, service and repair of technical devices. The PPP system is also used as a web service to retrieve and present various kinds of information to the user (André, Rist, and Müller 1997, 1998). A screen shot of the output of the PPP system is given in Figure 2.11. The presentation system involves a character that presents information to the user. The PPP system may distribute the information to be presented over several windows, which demands exophoric referring expressions to direct the user's attention to the appropriate target. The presentation planner of the PPP system (André and Rist, 1993) is responsible for three tasks (1) Determining the material to be presented; (2) Selecting the appropriate media combination for the content; and (3) Designing a presentation script. This presentation script is an ordered list of timed actions to be performed. The actions are generated by a graphics generator, a text generator and a gesture generator. The output of such a presentation script in case of a multimodal exophoric referring expression is presented as follows: The system first displays a window in which the object of the conversation, for example a modem's circuit board, is displayed. Next the ECA is moved to a position on the screen from which it can easily point to various elements located on the circuit board. Finally, the ECA performs a very precise pointing gesture using a stick, directed at one of the elements on the circuit board, while naming the element by producing a spoken referring expression, for example 'this is a transformer'.

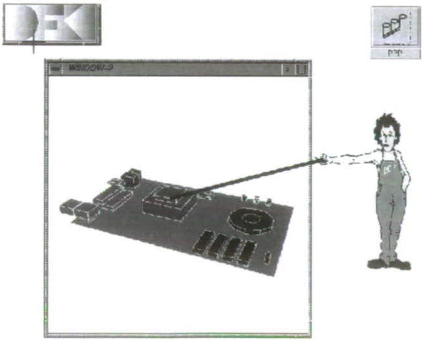


Figure 2.11: Screen shot of the PPP system taken from André and Rist (1996).

The ECA COSMO (Lester et al., 1999) supplies help in a plan-based learning environment applied to the domain of internet packet routing. The interface of the system is displayed in Figure 2.12. COSMO generates multimodal referring expressions on the basis of a spatial deictic framework founded on work by Cassell et al. (1994), Stone and Lester (1996) and Lester and Stone (1997). To generate the appropriate behavior for COSMO, the system uses a world model, a curriculum information network (i.e., a representation of problem-solving techniques for the given domain and task), a user model, the current problem state and a gestural and spoken focus history. In the system, a so-called explanation planner determines the content and structure of the output, which is passed on to the behavior planner. The behavior planner thus receives a communicative act, a topic, a spoken referent and a gestural referent, which are realized in speech, gesture and locomotion. In the case that a referring expression needs to be generated, the deictic planner is invoked with the intended referent. The deictic planner first determines the potential ambiguity of a referring expression for the target, dependent on the current focus. Subsequently, dependent on the potential ambiguity, it is determined if COSMO should point to the target. A pointing gesture is included if a target is not mentioned in the two previous utterances, or if other objects are mentioned in the previous utterances as well. In the case that a pointing gesture is required, there are three rules upon which the ECA should first move closer to the target before it can perform a pointing gesture: (1) The target is not located near to the ECA; (2) Other salient objects are located close to the target; or (3) The target is a relatively small object. The generated linguistic referring expression to accompany the pointing gesture is realized as a pronoun when the target is more salient than the other objects in the domain. If the target is potentially ambiguous, proximal or distal demonstrative determiners are generated, depending on the distance between COSMO and the target after any required movement of the ECA has taken place. If the target is of the same type as the other salient objects, the system generates the phrase 'this one'; in other cases the *type* property of the target is generated instead of the word 'one'. The result of the deictic planner is returned to the behavior planner to be presented to the user in a synchronized way. The generated referring expressions are always clear and unambiguous.

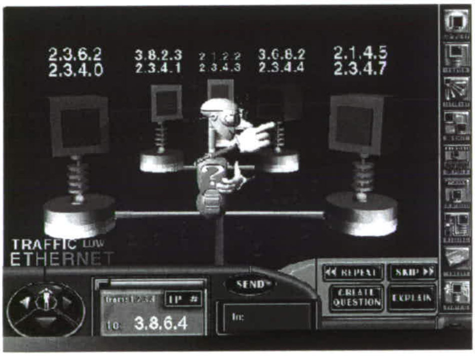


Figure 2.12: Screen shot of the COSMO system, taken from Lester et al. (1997).

The Faculties of Linguistics and Technology at the University of Bielefeld perform joint research in a project involving an ECA named Max, inhabiting a virtual environment for helping the user with assembly procedures in construction tasks (Sowa and Wachsmuth, 2001; Sowa et al., 2001; Wachsmuth and Kopp, 2001; Kranstedt et al., 2003; Kranstedt et al., 2005). Figure 2.13 illustrates the interaction with Max. Max' behavior in identifying objects is based on human performance during production experiments that involved identification tasks. The system makes use of a knowledge base that uses three kinds of information structures: (1) A tree expressing spatial relations of user movement and objects; (2) A semantic representation of the objects, which incorporates the linguistic information; and (3) A graph that represents the shape properties of the objects in the domain. The planning of the multimodal utterance incorporates the generation of verbal and nonverbal parts and the coordination of the two. The system uses a XML-based specification language to express multimodal utterances in a given context, in which the linguistic output is time stamped in order to produce gestural behavior accordingly. Most emphasis in the work on Max has been on the synchronization of speech and gesture based on work by de Ruiter (1998) and Mc Neill (1992). Max is able to produce various pointing gestures directed towards objects and areas together with simple noun phrases. Definite descriptions and pointing gestures are generated by a multimodal variant of the incremental algorithm by Dale and Reiter (1995) (Kranstedt and Wachsmuth, 2005). The proposed algorithm produces a pointing gesture in case both dialogue partners can see the target object. Subsequently, dependent on the scope of the pointing gesture Max can perform from his static position, there are two types of pointing that can be generated: (1) Object pointing, i.e., a pointing gesture that unambiguously identifies the target; and (2) Region pointing, i.e., a pointing gesture in whose scope more objects are located besides the target. In the first case no other linguistic information is generated, while in the second case the target is distinguished from the other objects in the scope of the pointing gesture by its properties, for example 'die lange Leiste' (the long bar) + region pointing.

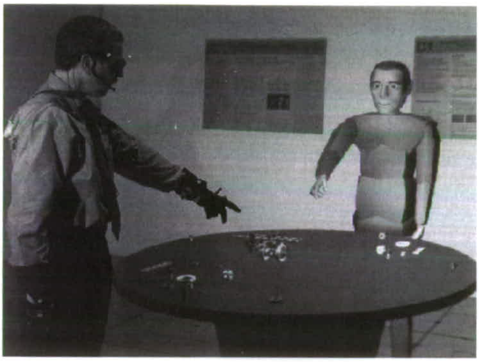


Figure 2.13: Interaction with Max, taken from Kopp et al. (2003).

Referring Expressions that include Deictic and Graphical Gestures

POPEL (Reithinger, 1992) is the generation component of the XTRA system (Allgayer et al., 1989), which helps a user to fill out a tax form. The XTRA system uses a dialogue memory, a user model and a form-hierarchy that keeps track of the graphical representations on the screen. Instead of using highlighting or blinking as in CUBRICON, XTRA tries to mimic human pointing gestures in a flexible manner. In this way not every item in the graphical representation on the screen has to be indexed, which is required in the case of highlighting objects. Moreover, XTRA also allows for pointing gestures to parts of the referent or to a place just below the referent (instead of covering the target with the pointing device while pointing), which is difficult to accomplish by means of highlighting. As displayed in Figure 2.14, pointing gestures are visualized in various ways dependent on the utterance in its context (e.g., a hand with a pointing finger or a hand that holds a pencil). POPEL can only include a pointing gesture if the current target can be associated with a node in the form hierarchy. If there exists such a connection, a pointing gesture is included if the target has not been talked about or if a linguistic target description is too complex. Thus the gesture might complement or even replace the linguistic description. When a pointing gesture is generated, the linguistic material is selected to accompany the gesture in an incremental and parallel way.

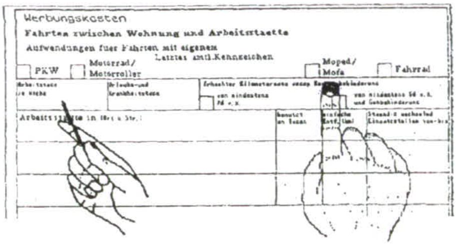
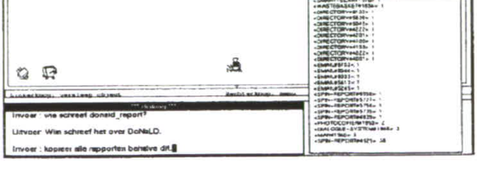


Figure 2.14: Pointing gestures as generated in POPEL, taken from Schmauks and Reithinger (1988).

As opposed to the CUBRICON system and the POPEL generation module, the EDWARD system (Claassen, 1992; Huls et al., 1995) uses one general context model that integrates both the linguistic and the non-linguistic information in the dialogue. The application domain of EDWARD is a file system environment with a graphical interface that users can manipulate by typed and mouse input. A screen shot of the interface is given in Figure 2.15, where the agent is sitting in a central position beneath the representation of the files. EDWARD uses a graph editor and a Dutch natural language dialogue system called DoNaLD (Claassen and Huls, 1991). Furthermore EDWARD uses a knowledge base and context model that employs a notion of salience derived from Alshawi (1987), which



Invoer : wie schreef donald_report?

Uitvoer: Wim schreef het over DoNaLD.

Invoer : kopieer alle rapporten behalve dit.

Figure 2.15: The interface of EDWARD, taken from Huls et al. (1995).

integrates various context factors and a notion of recency. The referring expressions generated are unimodal (i.e., pointing gestures or written linguistic referring expressions) or multimodal (i.e., a pointing gesture and a written linguistic referring expression). The linguistic output and the graphical output of the system involve two different windows in the interface. The pointing gestures evolve from a stationary agent that is located at the bottom of the graphical representation of the filesystem. The agent directs the attention to the target with an arrow that 'grows' in the direction of the target. As soon as this arrow has reached the target, the target is identified with comparatively little arrows that surround its location. As such, EDWARD uses a combination of pointing and highlighting. Unimodal pointing gestures are used in the case of user commands like 'remove that file', after which a pointing gesture is generated to indicate the target before the ordered action is executed. Multimodal referring expressions in EDWARD always consist of a linguistic description of the target plus a pointing gesture which is generated right after the head noun that identifies the target has been produced. In these cases the linguistic descriptions can express proper names, noun phrases that express a basic level category (e.g., the report), and modified noun phrases (e.g., the report about parsing). The noun phrase modifiers in EDWARD are relative clauses and prepositional phrases, (i.e., adjectives are not included). EDWARD generates multimodal referring expressions when: (1) The target is not very salient; (2) The system has no distinguishing information about the target; and (3) The target is visible. In all other cases a purely linguistic referring expression is generated.

The SmartKom system (Wahlster et al., 2001; Wahlster, 2002; Wahlster, 2003a) is a multimodal dialogue system that applies the use of speech, gestures and facial expressions in both input and output to a wide range of electronic devices that employ audio tools, touch screens, projectors and cameras. Figure 2.16 presents a possible interface of the SmartKom system. Similar to CUBRICON, SmartKom makes use of a unified multimodal representation language for all input and output. The system is applied to several application domains, in which for instance information about the use of electronic devices, help with the selection of TV programs or travel guidance can be obtained from an ECA named Smartakus. Similar to the actions of the PPP agent (André and Rist, 1996), Smartakus may for example present a city map by pointing at a map on which the names of the cinemas requested are highlighted at the appropriate lo-

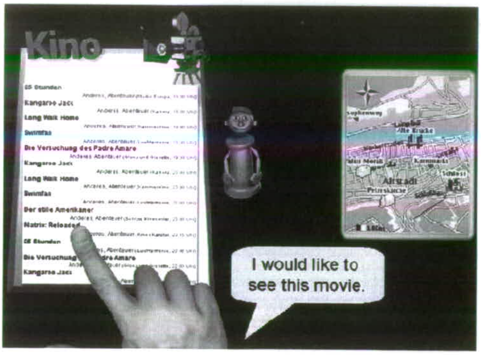


Figure 2.16: The interface of SmartKom, taken from Wahlster (2003b).

cations in the city (Wahlster et al., 2001). In another application Smartakus can present the scheduled TV programs by pointing at a listing and uttering 'here is a listing of tonight's tv broadcasts' (Wahlster, 2003a). SmartKom uses a multimodal discourse representation that accommodates the domain, the discourse and the modalities. In this representation the linguistic, visual and gestural modalities are in turn related to the objects in the discourse, that are related to the domain of conversation. The actual output of the system is coordinated by a presentation planner that applies predefined presentation strategies that separate the presentation goal into various presentation tasks (Reithinger et al., 2003). In the fission process a template-oriented approach based on Tree Adjoining Grammars (Abeillé and Rambow, 2000) for NLG is employed.

2.5.3 Differences and Similarities

For the sake of readability the various algorithms discussed above are summarized in two tables; one presenting the gestural part and one addressing the linguistic part of the referring expressions. In Table 2.1 the differences and similarities between the algorithms are shown across several features concerning the generated gestures. The first two columns indicate whether the algorithm generates graphical or deictic gestures, or both. The third column indicates whether an ECA is used to present information to the user. The fourth column, shows the precision of the generated pointing gestures. Except for the Max system, all algorithms discussed in this section generate pointing gestures that are precise and unambiguous. As soon as a pointing gesture is included, it singles out the intended referent from the other objects in the domain. Note that unlike the other systems, the COSMO agent is able to produce a pointing gesture directed at an object located at a certain distance, but only in case there are no other objects located near the target. Consequently, COSMO's pointing gestures are not ambiguous but precise. In contrast, the Max system is able to generate various pointing gestures depending on the objects in the scope of the gesture, while the agent itself keeps a static position.

In Table 2.2, the linguistic part of the output of the algorithms is split into three features that show the differences: (1) The output modalities, spoken or written; (2) The kind of referring expressions that the systems can produce; and (3) An example of a referring expression the system is able to generate. All algorithms, except for the one used in the Max system, use templates and canned text. In most cases the algorithms generate a noun phrase consisting of a determiner and a head noun, the latter of which indicates the *type* of the target. EDWARD and SmartKom are able to generate an extension in the form of a prepositional phrase. Some of the algorithms can also generate proper names. A special rule provides COSMO with the ability to utter pronouns and 'the one'. In contrast to these template-based systems, the Max system uses a simplified version of the algorithm proposed by Dale and Reiter (1995). This algorithm, which is explained in detail in Section 3.3.2, generates a referring expression by selecting properties that distinguish the target from the other objects in the domain. Accordingly, Max is the only system that generates adjectives for identification, dependent on the objects in the scope of the pointing gesture.

System	DG	GG	ECA	TG
CUBRICON	no	yes	no	precise
CHAMELEON	no	yes	no	precise
DenK	no	yes	no	precise
PPP	yes	no	yes	precise
COSMO	yes	no	yes	precise
Max	yes	no	yes	various
POPEL	yes	yes	no	precise
EDWARD	yes	yes	yes	precise
SmartKom	yes	yes	yes	precise

Table 2.1: Overview of the gestural part of the multimodal referring expressions generated by multimodal algorithms. The algorithms are presented by the name of the system in which they are used. The abbreviations in the table are defined as follows: *DG* = deictic gestures, *GG* = graphical gestures, *ECA* = embodied conversational agent and *TG* = type of gestures.

System	S/W	RE	Example
CUBRICON	S/W	NP→Det+*type*	'this SAM' = (surface-to-air missile system)
CHAMELEON	S	NP→Det+*type*	'Tom's office'
DenK	W	NP→Det+*type*	'the C2-lens'
PPP	S	NP→Det+*type*	'a transformer'
COSMO	S	NP→Det+*type*,Det+'one',prn	'this router/one', 'it'
Max	S	NP→Det+[Adj]+*type*	'the [lange] leiste'
POPEL	W	NP→Det+*type*	'this amount'
EDWARD	W	NP→Det+*type*+[PP]	'the report [about parsing]'
SmartKom	S/W	NP→Det+*type*+[PP]	'a listing [of tonight's tv broadcasts]'

Table 2.2: Overview of the linguistic part of the multimodal referring expressions generated by multimodal algorithms. The algorithms are presented by the name of the system in which they are used. The abbreviations in the table are defined as follows: *S/W* = spoken or written output, *RE* = type of generated referring expressions. In the column *RE*, *NP* = noun phrase, *Det* = determiner, *PP* = prepositional phrase, *Adj* = adjective. Everything between square brackets [ ] is optional.

As already noted, most algorithms described in this section assume that a pointing gesture is precise and unambiguous; it singles out the intended referent from the other objects in the domain. As a consequence, the generated expressions tend to be relatively simple and usually contain no more than a head noun in combination with a pointing gesture. An exception is the Max system, which is able to generate a distinguishing referring expression in the case when the gesture is too imprecise for the identification of a target. But all algorithms tend to be based on relatively elementary, context-independent criteria for deciding whether a pointing gesture should be included or not. For instance, CUBRICON, CHAMELEON and the PPP system always include a gesture. POPEL, the DenK system and the SmartKom system include gestures whenever possible. EDWARD only generates a pointing gesture when referring to an object for which no distinguishing linguistic description can be produced. COSMO produces pointing gestures for all objects which cannot be referred to with a pronoun. And finally Max produces a pointing gesture whenever the target is visible to both discourse participants.

Although NLG is a key component of a dialogue system, the focus in most systems developed so far is primarily on interpretation, while the generation part is often covered using simple, straightforward solutions. With the development of more complex dialogue systems, such as for example the SmartKom system, together with the enhanced human-like behavior of ECAs like Max, the demand for more advanced generation methods is likely to increase. In the generation of multimodal referring expressions, some progress can be made by focussing on: (1) The incorporation of the discourse and the perceptual context in order to derive an adequate exploitation of modalities; (2) A flexible inclusion of various pointing gestures as occurring in human communication; and (3) Fission improvement, i.e., merging the linguistic and the gestural part of the multimodal referring expression in a natural way. As seen in Section 2.4, in human communication a greater variability of multimodal referring expressions is observed than the current algorithms are able to generate. Hence in this thesis an algorithm is proposed that mimics the human production of referring expressions by integrating a multimodal notion of context and the ability to generate various kinds of pointing gestures. The algorithm combines the gestural and linguistic modality aspect of the referring expressions in a natural and complementary fashion. Of the algorithms presented in this section, the approach advocated in this thesis is closest to the work in POPEL and that on Max, in that it focusses on multimodal generation in the way that models human behavior.

2.6 Discussion

By discussing the generation of referring expressions in multimodal systems, this chapter presented the background for the next chapters of this thesis. The development of multimodal systems was addressed, with multimodal dialogue systems as a specific instance of such systems. Within multimodal dialogue systems the focus was narrowed to the generation part, which was approached as a plan-based process that results in multimodal presentations. The attention was further restricted to the generation of multimodal referring expressions as a microplanning task in such a process. A detailed discussion was presented on the generation of multimodal referring expressions both by humans and by machines, where purely linguistic descriptions, deictic gestures and their integration were considered. A concise overview of existing algorithms for the generation of multimodal referring expressions using deictic gestures shows that the generation side of multimodal dialogue systems receives less attention than required, as the development of such systems is to advance to more complex applications. From a comparison of object identification as occurring in human communication to the current algorithms that generate multimodal referring expressions, it can be inferred that modeling automatic generation after human production requires more attention for context-sensitivity, the generation of various kinds of pointing gestures and a proper merging of modalities. In an attempt to meet these requirements, the objective of this thesis is to design a context-sensitive algorithm that is able to generate multimodal referring expressions where the pointing gestures and linguistic referring expressions complement each other in a natural way. As a starting point, Chapter 3 gives a thorough overview of the algorithms developed for the generation of linguistic referring expressions.

Chapter 3

Generating Referring Expressions

3.1 Introduction

This chapter provides an overview of the state of the art in the generation of referring expressions.[1] It is organized as follows: In Section 3.2, the different terms used in the generation of referring expressions are explained and defined. In Section 3.3, two basic algorithms for the generation of referring expressions are presented. An algorithm for the generation of minimal descriptions is presented in Section 3.3.1, and the Incremental Algorithm of Dale and Reiter, which result in more natural descriptions in Section 3.3.2. In Section 3.4, several extensions to the Incremental Algorithm are examined that concern completeness. Comparisons between the algorithms are facilitated by means of a uniform presentation format (c.f., (Bohnet and Dale, 2004) for a different approach to compare and contrast the different algorithms). In Section 3.5, a three-dimensional notion of salience is defined that provides for the context-sensitivity of the algorithm. Section 3.6 discusses what is missing in the state of the art, and from where to proceed in the following chapters in this thesis.

3.2 Basic Notions

The generation of referring expressions (GRE) is one of the primary tasks in NLG (Dale and Reiter, 2000, section 5.4). It is arguably also one of the most clearly defined ones: given a target object r and its properties, decide what is the most

[1] An annotated bibliography of the research in GRE can be found at the web site of the TUNA project: http://www.csd.abdn.ac.uk/~agatt/tunabibl/index.html

accurate way to refer to r in the current context. In this chapter it is shown that different algorithms interpret the term 'accurate' in different ways. The current context can be defined as the **context set** C that consists of r and the objects in D from which r has to be distinguished, which are called the **distractors** (terminology from Mc Donald (1981)). The goal of a typical GRE algorithm is to single out the target r by selecting a set of properties L that is only applicable to r and not to any of its distractors. This process is usually referred to as **content determination** for referring expressions.

Once a GRE algorithm has selected a set of properties L, this set can be realized as a natural language expression. Note that a GRE algorithm does not itself output a linguistic expression, rather it feeds the selected properties to a linguistic realizer in the lexicalization module (see Section 2.3.1). In this thesis the focus is specifically on content determination, not on linguistic realization. It is assumed that a set of properties L can be realized using standard realizers such as FUF/SURGE, (Elhadad, 1993; Elhadad and Robin, 1998).[2]

A GRE algorithm can be illustrated with a simple example in a block domain. For example in Figure 3.1 the context set C consists of: d_1, d_2 and d_3. The objects can be represented in a knowledge base (KB). Each object d can be characterized with a set of properties P_d. The properties p in a set P_d can be represented as attribute-value pairs, $\langle A, V \rangle$. To represent the objects in Figure 3.1 the following attributes can be used: *type*, *color*, *shape*, and *size*. Assigning values to these attributes for every object in Figure 3.1 results in the KB presented in Figure 3.2.

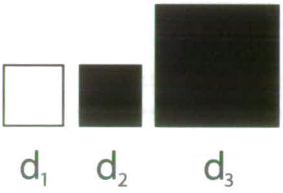


Figure 3.1: Example Domain I.

$$
\begin{array}{ll}
P_{d_1} & = \{ \langle \textit{type}, \textit{block} \rangle, \langle \textit{color}, \textit{white} \rangle, \langle \textit{shape}, \textit{square} \rangle, \langle \textit{size}, \textit{small} \rangle \} \\
P_{d_2} & = \{ \langle \textit{type}, \textit{block} \rangle, \langle \textit{color}, \textit{black} \rangle, \langle \textit{shape}, \textit{square} \rangle, \langle \textit{size}, \textit{small} \rangle \} \\
P_{d_3} & = \{ \langle \textit{type}, \textit{block} \rangle, \langle \textit{color}, \textit{black} \rangle, \langle \textit{shape}, \textit{square} \rangle, \langle \textit{size}, \textit{large} \rangle \}
\end{array}
$$

Figure 3.2: KB for Example Domain I in Figure 3.1.

[2] Throughout this thesis the terms *description* and *referring expression* are used to refer to both the set of properties and its linguistic realization.

Note that all objects in Figure 3.1 are assigned the same value for the attributes *type* and *shape*, respectively *block* and *square*. The differences between the objects in Figure 3.1 are represented by the values assigned to the attributes *color* and *size*. The attribute-value pairs provide the building blocks to generate referring expressions. A GRE algorithm searches for a set L of attribute-value pairs, or properties, in C that provides a unique semantic description of r. As defined below, the denotation of a property p in C is the set of objects for which p is true. Accordingly, the denotation of a set of properties is the intersection of the denotations of the properties in the set.

$$[\![\, p \,]\!]_C = \{d \in C \mid d \text{ has property } p\}$$
$$[\![\, \{p_1, \ldots, p_n\} \,]\!]_C = [\![\, p_1 \,]\!]_C \cap \ldots \cap [\![\, p_n \,]\!]_C$$

The denotation of the set L then contains the properties that are true for r. Thus the denotation of L equals the singleton set that contains r: $[\![\, L \,]\!]_C = \{r\}$. For now, let's assume that there is such a set L, which implies that the set of properties of r, P_r, is not empty and that there are no objects in C that have exactly the same set of properties as r.[3] There are several possibilities to describe an object. As an example consider the selections of attribute-value pairs for d_3 in Figure 3.3. These sets can be realized respectively as the definite NPs presented in Figure 3.4. Of these descriptions (1) and (2) are ambiguous; (1) can refer to any of the objects in C and (2) can refer to both d_2 and d_3. Although the descriptions (3), (4) and (5) differ, they are all **distinguishing**: they only apply to the target and not to any other object in C. Description (5) includes all properties defined for d_3, whereas (3) and (4) only use a subset. Not all these descriptions are equally suitable; (1) and (2) fail to single out the target, and (4) and (5) contain some redundant properties. In general a GRE algorithm has to find a subset of the properties with which the target can be described **accurately**. For this purpose several GRE algorithms have been developed in the last decade. In the next section two of these algorithms are addressed. It will be shown that there is more than one possible interpretation of the term 'accurate'.

(1) $\{ \langle$ *shape*, *square* $\rangle \}$
(2) $\{ \langle$ *type*, *block* $\rangle, \langle$ *color*, *black* $\rangle \}$
(3) $\{ \langle$ *type*, *block* $\rangle, \langle$ *size*, *large* $\rangle \}$
(4) $\{ \langle$ *type*, *block* $\rangle, \langle$ *size*, *large* $\rangle, \langle$ *color*, *black* $\rangle \}$
(5) $\{ \langle$ *type*, *block* $\rangle, \langle$ *size*, *large* $\rangle, \langle$ *color*, *black* $\rangle, \langle$ *shape*, *square* $\rangle \}$

Figure 3.3: Five potential sets of attribute-value pairs L for d_3.

[3] Where this can be done without creating confusion subscripts are omitted

(1) 'the square'
(2) 'the black block'
(3) 'the large block'
(4) 'the large black block'
(5) 'the large black square block'

Figure 3.4: Possible realizations for d_3.

3.3 Basic Algorithms

3.3.1 Full Brevity Algorithm

One approach to generate accurate referring expressions is to look for a **minimal description**: the shortest distinguishing description for a target object r (for examples see Dale, 1988; Gardent, 2002; Varges, 2004). To generate minimal descriptions Dale (1989) proposed the **Full Brevity Algorithm**. This GRE algorithm is based on two principles: a principle of adequacy and a principle of efficiency, which together account for referring expressions that are distinguishing and minimal. The principles are derived from the Gricean conversational maxims of *quantity*, *relevance* and *brevity* (Grice, 1975) and are made applicable for referring expressions. Below the principles are quoted from Dale (1989).

- **Principle of Adequacy**: "a referring expression should identify the intended referent unambiguously, and provide sufficient information to serve the purpose of the reference"
- **Principle of Efficiency**: "the referring expression used must not contain more information than is necessary for the task at hand"

To generate a minimal description the algorithm has to find the smallest set of properties L that unambiguously represents r. Informally this works as follows. In a first iteration, the algorithm inspects whether a single property is sufficient to single out r. If there is such a property in P_r, the algorithm is finished and returns this property for realization. If none of the individual properties suffices, the algorithm inspects every combination of two properties. A set L consisting of two properties is returned only if the two properties together are exclusively applicable to r. Again, if no such set is available, the algorithm tries every combination of three properties, and so on until the set that consists of all properties of r is tried. The Full Brevity Algorithm finishes if it encounters a set L with which r can be distinguished from its distractors (success) or if all combinations of the properties in P_r have been tried (failure).

The pseudocode for the function *GenerateMinimalDescription* presented in Figure 3.5 makes this more precise. In line (1) *GenerateMinimalDescription* is called with the parameters r (the target) and C (the context set). The properties of all objects in C, including P_r, are assumed to be globally accessible from a KB. In line (2) L is initialized as the empty set. In line (3) a counter i is initialized as one. With line (4) a search for the smallest set L is initiated. As long as i is not larger than the cardinality of the set properties of r, $|P_r|$, the algorithm inspects all possible subsets of P_r starting with the sets consisting of one property, line (5). If the denotation of L in C only describes r and no other object in C, the algorithm returns L, line (6). If there is no set consisting of i properties in P_r with which all distractors can be ruled out the algorithm increases i with one in line (7) and repeats the same procedure starting at line (4). If the algorithm did not come across a successful set L and i is larger than the number of properties in P_r the algorithm returns failure in line (8).

```
(1) GenerateMinimalDescription($r, C$)
(2)   $L := \emptyset$
(3)   $i := 1$
(4)     while $i \leq | P_r |$ do
(5)       foreach $L \subseteq P_r$ where $| L | = i$ do
(6)         if $[\![ L ]\!]_C = \{r\}$ then return $L$
            endif
          end foreach
(7)       $i := i + 1$
        end while
(8)   return failure
```

Figure 3.5: Pseudocode Full Brevity Algorithm.

As an example the Full Brevity Algorithm is applied to Example Domain I in Figure 3.1. Let's take d_2 as a target. The properties of d_2 are represented in P_{d_2} as: $\{\langle$ *type*, *block* $\rangle$, $\langle$ *color*, *black* $\rangle$, $\langle$ *shape*, *square* $\rangle$, $\langle$ *size*, *small* $\rangle\}$. The distractors are d_1 and d_3. The algorithm inspects for each single property in P_{d_2} if it can rule out both distractors. The properties $\langle$ *type*, *block* $\rangle$ and $\langle$ *shape*, *square* $\rangle$ are not distinguishing at all, they apply to all objects in C. The property $\langle$ *color*, *black* $\rangle$ only rules out d_3 and the property $\langle$ *size*, *small* $\rangle$ only rules out d_1. In both cases not all distractors are ruled out. Now the algorithm has to inspect all possible subsets in P_r that contain two properties. This leads to success, if the algorithm encounters the set $\{\langle$ *color*, *black* $\rangle$, $\langle$ *size*, *small* $\rangle\}$. The denotation of this set L only describes d_2. Both the distractors d_1 and d_3 are ruled out and L is returned. Subsequently L can be realized for example as 'the small black block'.

The Full Brevity Algorithm finds the shortest description for the target object (if there is one). In case there is more than one minimal description the algorithm generates the first one it encounters, whereby the output is dependent upon the ordering of the properties that are tried. The algorithm is tailored to principles inferred from the Gricean conversational maxims to generate referring expressions in an understandable way for humans. The focus is primarily on the quantity of information, for which the Gricean maxims are very strictly applied: the referring expression should neither include more, nor less properties than necessary to distinguish the intended referent. However, it might not be practical to generate referring expressions as minimal descriptions. According to the empirical evidence provided by for example Pechmann (1989) and Beun and Cremers (1998), people tend to produce overspecified instead of minimal referring expressions, i.e., they include more properties than necessary to identify the target. Dale and Reiter (1995, page 248) note that: "For example, in a typical experiment a participant is shown a picture containing a white bird, a black cup, and a white cup and is asked to identify the white bird; in such cases participants generally produce the referring expression 'the white bird', even though the simpler form 'the bird' is sufficient". In the case of the example presented above, instead of the generation of the minimal description 'the small black block', a more redundant description, for example 'the small black block next to the large one', could be more suitable if the goal is to model human production.

Apart from the accuracy of the generated referring expression another criterion to consider is the complexity of the task: to generate a minimal description means to perform an exhaustive search. The smallest subset L is found by checking all properties and also all combinations of all properties until a success is encountered. In the field of complexity analysis such a task belongs to the class of NP-hard problems (Garey and Johnson, 1979). Looking at the Full Brevity Algorithm, it is easy to see that for larger context sets the number of combinations of properties that have to be checked soon becomes infeasible. If the target object cannot be described with the use of a single property, the time it takes to generate a referring expression increases exponentially with the cardinality of the set of properties to be checked. Moreover, the higher the number of distractors, the longer it takes the algorithm to find a solution. For a more detailed discussion of NP-hard tasks and computational efficiency in relation to GRE (Dale and Reiter, 1995, section 3). Based on the observations described above about computational efficiency and accuracy Dale and Reiter (1995) proposed the Incremental Algorithm, which is discussed in the next section.

3.3.2 Incremental Algorithm

The aim of Dale and Reiter's **Incremental Algorithm** is to efficiently generate a distinguishing description. Instead of concentrating on a minimal description, the algorithm also generates overspecified descriptions just as people do. For the sake of psychological realism and computational efficiency Dale and Reiter (1995) discard the strict interpretation of the Gricean maxims and propose a simple and fast algorithm that incrementally selects attributes to describe a target. To pick the best attributes first, the Incremental Algorithm uses a list of **preferred attributes**. In this list the properties relevant for the domain are ordered according tọ the preference that human speakers and hearers have when discussing objects in that particular domain. The exact order of properties for a particular domain is an empirical matter (c.f Mangold and Pobel, 1988; Arts, 2004). However, some general trends exist. For instance, usually speakers first try to distinguish an object by its *type* property (c.f., Sonnenschein, 1982). In addition, speakers have a general preference for **absolute** properties such as *color* and *shape*, over **relative** properties such as *size*. This may be explained by the fact that relative properties are less easily observed and always require inspection of other objects in the domain (e.g., Pechmann, 1989; Levelt, 1989; Beun and Cremers, 1998 for empirical evidence). Moreover, relative properties are often subjective and more prone to misinterpretation.

The input of the Incremental Algorithm consists of the target object r and a context set C, where r is the object to be described and where the context set contains all objects in the domain. The algorithm essentially iterates through the list of preferred attributes $\mathcal{A}$, adding a property to the description of r only if it rules out one or more distractors in the context set that are not previously ruled out. Dale and Reiter make the assumption that the property *type* must always be included in a distinguishing description even if it has no discriminating power (i.e., even if it did not rule out distractors).[4] The Incremental Algorithm terminates when all distractors are ruled out (success) or when all properties of r have been checked (failure).

The pseudocode for the function *GenerateDistinguishingDescription* presented in Figure 3.6 makes this more precise. In line (1) the function is called with the parameters r (the target) and C (the context set). The properties of the objects in C are assumed to be globally accessible from a KB. In line (2) L is initialized as the empty set. In line (3) the search for a distinguishing description for r is initiated. For each attribute A of r in the ordered list of preferred attributes $\mathcal{A}_r$ it is checked in line (4) if the denotation of $\langle\ A,\ V\ \rangle$ holds for r and if it rules out any distractors in C. If this is the case two things happen. First, in line (5) the property $\langle\ A,\ V\ \rangle$ is added to L. Second, in line (6) all distractors outside $[\![\langle\ A,\ V\ \rangle]\!]$ are removed from C. In line (7) it is tested whether all distractors are ruled out. If r is the only member left in C and if the *type* property of r is not already included in L, line (8), *type* is added to L in line (9). Successively, in line (10) L is returned. If all attributes A in $\mathcal{A}_r$ are checked and there are still distractors in C the algorithm returns failure in line (11).

[4] Dale and Reiter(1995, section 4.1) make use of a subsumption taxonomy in the KB with **basic level values** and **more specific values** (see also Reiter, 1991). The selection of the **best value** for a preferred attribute depends on the view of the system and of the view of the current user. For the attribute *type*, the basic level value is used, if *type* does not rule out any distractors. For simplicity reasons the subsumption taxonomy is ignored in this thesis.

```
GenerateDistinguishingDescription($r, C$)
  $L := \emptyset$
    foreach $A \in \mathcal{A}_r$ do
        if $r \in [\![ \langle A, V \rangle ]\!]$ and $C \not\subseteq [\![ \langle A, V \rangle ]\!]$ then
          $L := L \cup \{ \langle A, V \rangle \}$
          $C := C \cap [\![ \langle A, V \rangle ]\!]$
        endif
        if $C = \{r\}$ then
          if $\{ \langle type, V_r \rangle \} \notin$ L then
            $L := L \cup \{ \langle type, V_r \rangle \}$
          endif
        return $L$
        endif
    end foreach
  return failure
```

Figure 3.6: Pseudocode Incremental Algorithm.

As an example let's reconsider Example Domain I in Figure 3.1, and apply the Incremental Algorithm to refer to d_3. This implies that the distractors are the other two objects in this particular domain, d_1 and d_2. In line with the observations made above, let's assume that the ordered list of preferred attributes is $\langle$ *type*, *color*, *shape*, *size* $\rangle$. The algorithm finds that the attribute *type* is not suited to distinguish d_3: it rules out no distractors and is therefore discarded. By including the property $\langle$ *color*, *black* $\rangle$ in L, one distractor can be removed: the white block d_1. But the attribute *color* is not sufficient to distinguish d_3, d_2 still remains as a distractor. Since d_2 and d_3 have the same shape, the attribute *shape* has no effect. To rule out d_2, the relative property $\langle$ *size*, *large* $\rangle$ can be used. Finally, the Incremental Algorithm inspects whether the *type* attribute is included, and since this is not the case, it is added to the set of distinguishing properties of d_3 after all. The set of selected properties $\{\langle$ *color*, *black* $\rangle$, $\langle$ *size*, *large* $\rangle$, $\langle$ *type*, *block* $\rangle\}$ can now be realized linguistically by the distinguishing description 'the large black block'.

The two main advantages of the Incremental Algorithm are its efficiency and its psychological realism. The efficiency of the algorithm is illustrated by its complexity, which is polynomial in time (Dale and Reiter, 1995, page 247). Within the Incremental Algorithm there is no backtracking, hence the term **incremental**: once a property p has been selected, it is realized in the final description, even if a property which is added later renders the inclusion of p redundant. In the example in Figure 3.1 'the large block' would be distinguishing but the relative property *size* is only included after the absolute property *color* is added. Because there is no backtracking, the Incremental Algorithm is fast and efficient. In addition, Dale and Reiter claim that the algorithm is psychologically realistic because human speakers also often include redundant modifiers in their referring expressions (where they refer to Pechmann, 1989).

3.3.3 Discussion

Both the Incremental Algorithm and the Full Brevity Algorithm have a number of restrictions. For instance, the algorithms only provide for singular descriptions that can be generated by the conjunction of a finite number of properties. Since references to sets of objects are very common in human conversation, the generation of plural descriptions is a natural and important step in the development of GRE algorithms. But both basic algorithms only use conjunctions of properties to describe objects. This implies that, when referring to a set of objects, the target objects need to have enough properties in common to distinguish them as a group from their distractors. The fewer the properties that the target objects share, the greater the likelihood that search for a distinguishing expression will fail. The basic algorithms fail to describe sets of objects or singular objects that cannot be distinguished by conjunctive combinations of properties, whereas in human communication, a distinguishing reference is often possible using a negation or a disjunction of properties. Clearly, such a boolean extension that includes disjunctions and negations in the generation of referring expressions is very beneficial especially for the generation of plural descriptions. It adds to the **completeness** of the algorithm. An algorithm is complete if it generates a description whenever there is one. Nevertheless, even in cases where the target object(s) can be uniquely described, it is still possible that the Incremental Algorithm is not able to generate a distinguishing description. These failures are due to the lack of backtracking when properties overlap (van Deemter, 2002, section 3) or are caused by infinite domains where there are too many distractors.

Another restriction of the basic algorithms is that they lack a sense of context in a number of ways. For example, the treatment of relative attributes is rather unsatisfying. Relative attributes like *size* are context dependent: the size of an object is rated large or small dependent on the absolute sizes of the other objects in the context set. In contrast, the algorithms use a KB in which these relative properties are stored, without any consideration of the other objects in the context set, the values of for example *size* are *small* or *large*. For the context set presented in Figure 3.1, the KB in Figure 3.2 might be adequate, but it is easy to imagine another context set in which d_3 is not that large because there are other, larger, objects around. Except for the validation of relative properties, the other objects in the context set can also be used to describe a target object. Although references that include spatial relations between the objects are rather common in human conversation, the described algorithms just fail if the target cannot distinguishingly be described by its own properties. For example none of the objects in Example Domain II in Figure 3.7 can unambiguously be described without a spatial relation. In this case 'the white square block' describes both

d_2 and d_3. If d_2 is the target, it can for example be identified with use of its relation to d_1: 'the white block to the right of the black one'.[5] In this realization d_1 is called the **relatum** of the target object following Levelt (1989). Thus, in GRE the objects in the context set that stand in a close spatial relation to the target object can be used as a relatum to distinguish the target. Yet neither the Incremental Algorithm nor the Full Brevity Algorithm have the means to include a locative relatum in a referring expression. In Section 3.4.4 this issue returns.

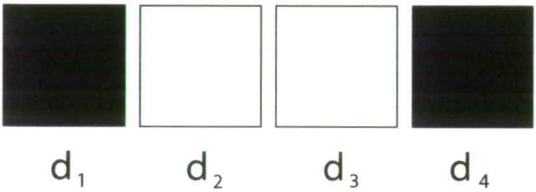


Figure 3.7: Example Domain II.

Besides the contextual information that is actually expressed in the generated referring expressions, there is another kind that is not used by the algorithms: **salience**. So far the algorithms distinguish the target from all objects in the domain and thereby generate only suitable descriptions for objects that have not been mentioned before. In human conversation it is considered tedious to repeat the same distinguishing referring expression for an object more than once in a short time. An object that has already been talked about is often referred to in a more concise way, because speakers prefer to use a simpler form to minimize their own and the hearers effort (e.g., Beun and Cremers, 1998; Clark and Wilkes-Gibbs, 1986; Zipf, 1949). With a notion of **linguistic salience** the algorithms can restrict the context as a subset of the domain, thereby tailoring the generated referring expressions to the discourse. There are more reasons for which an object can be salient. For instance, the **focus space**: objects that are located in the neighborhood of an object that has just been mentioned inherit some salience (Beun and Cremers, 1998). Furthermore, there are objects that are **inherently salient**: objects that stand out compared to the other objects in the domain, because they have a certain property that the other objects do not have.

To conclude, neither the Incremental Algorithm nor the Full Brevity algorithm include all the kinds of context that may be relevant in the GRE process. Moreover the generation of boolean combinations of properties and the generation of plural descriptions is not possible. Of the two algorithms, the Incremental Algorithm is generally accepted as the state of the art in the generation of referring expres-

[5] For the sake of simplicity, Dale (1988, section 5.3) is followed here in assuming that 'one' is used instead of a full head noun N when the context of a description contains another NP whose head is also N.

sions and various extensions have been proposed that resolve one or more of the above mentioned restrictions. In the following two sections some extensions are discussed that explicitly aim at keeping the attractive properties of the Incremental Algorithm, in particular, speed, low complexity and psychological plausibility. Section 3.4 discusses the extensions that account for plural descriptions, relative attributes, negations, disjunctions, and locative related objects. In Section 3.5, a three-dimensional notion of salience is defined to enrich a GRE algorithm with a more elaborated sense of context.

3.4 Extensions

This section discusses several extensions to the Incremental Algorithm. In Section 3.4.1 a variation of the Incremental Algorithm, that accounts for plural descriptions is presented. Section 3.4.2 deals with a context-sensitive inclusion of relative properties. The last two sections both concern algorithmic completeness. Two methods are discussed that increase the success of the algorithm in cases where the target object cannot be described in terms of its own properties. In Section 3.4.3 the expansion of the generated descriptions by means of boolean combinations of properties is considered. Section 3.4.4 discusses the generation of referring expressions including spatial relations.

3.4.1 Plurals

The Incremental Algorithm only generates singular referring expressions and no plurals. To generate plural NPs van Deemter (2000) transforms the Incremental Algorithm into an algorithm that accepts sets of objects as a target. The algorithm is called with two sets: (1) a target set S that contains all objects to be described and (2) a context set C that contains the members of S and their distractors. While iterating through an ordered list of preferred attributes, the algorithm adds to L those properties that describe S and at the same time rule out the distractors from C. The distractors that are ruled out by a property that is only applicable to the objects in S are removed from C. The algorithm is successful if C equals S and if so, it returns the selected properties. To describe an individual object the algorithm is called with a singleton set $\{r\}$.

In Figure 3.8 the pseudocode of the Incremental Algorithm presented in Figure 3.6 is slightly altered according to van Deemter (2000) to handle sets of objects. In Figure 3.8, line (1) calls the function with the parameters S (the target set) and C (the context set). The properties of the objects in C are assumed to be globally accessible from a KB. In line (2) the search for a distinguishing description for S is initiated. For each attribute A of the objects in S in the ordered list of preferred attributes $\mathcal{A}_S$, it is checked in line (3) if the denotation of $\langle\ A,\ V\ \rangle$ holds for all objects in S and if it rules out any distractors in C. In line (4) the code is given to test whether all distractors are ruled out. If S equals C then if necessary, in line (6) the *type* property of the objects in S is added to L and L is returned. Otherwise if all attributes A in $\mathcal{A}_S$ are checked and there are still distractors in C the algorithm returns failure.

```
(1)  GenerateDistinguishingDescription($S, C$)
       $L := \emptyset$
(2)      foreach $A \in \mathcal{A}_S$ do
(3)         if $S \subseteq [\![ \langle A, V \rangle ]\!]$ and $C \not\subseteq [\![ \langle A, V \rangle ]\!]$ then
              $L := L \cup \{ \langle A, V \rangle \}$
              $C := C \cap [\![ \langle A, V \rangle ]\!]$
            endif
(4)         if $C = S$ then
(5)           if $\{ \langle type, V_S \rangle \} \notin$ L then
(6)             $L := L \cup \{ \langle type, V_S \rangle \}$
              endif
              return $L$
            endif
         end foreach
       return failure
```

Figure 3.8: Pseudocode Plural Algorithm.

To illustrate the functioning of this variant of the Incremental Algorithm, henceforth called the **Plural Algorithm**, the objects in Figure 3.9 are used as a context set C. Suppose no object has been mentioned and the target set $S = \{d_4, d_5, d_6\}$ has to be referred to. Consequently, the distractors from which the objects in S have to be distinguished are d_1, d_2 and d_3. The algorithm iterates through the ordered list of preferred attributes as before. Each attribute is checked to see whether it holds for all objects in S and whether it rules out any distractors. The attribute *type* which has the same value for all objects in S does not rule out any distractors and is therefore discarded. Since the objects in S cannot be described with a single value for the attribute *color*, the attribute cannot be used to rule out any distractors. The attribute *shape* has again the same value for all objects in S, but it does not rule out any distractors. The algorithm then inspects the attribute *size*, which has one suitable value to describe all objects in the target set and which at the same time rules out all distractors. The *type* property is included and $L = \{\langle$ *size*, *large* $\rangle$, $\langle$ *type*, *block* $\rangle\}$ can be realized as 'the large blocks'.

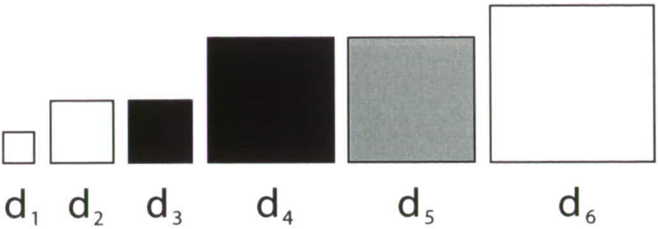


Figure 3.9: Example Domain III.

There are two constraints on the success of the Plural Algorithm: (1) The objects in the target set need to have at least one property in common, which is of course not shared by the distractors, (disjunctive combinations of different values of the same attribute like 'the white and the black blocks' are not possible, but see Section 3.4.3); and (2) The value of the attribute *type* must hold for every object in the target set, because there is only room for one head noun in a referring expression. To relax this constraint, van Deemter gives up the systematic inclusion of the *type* property and suggests to choose *type* in the realization phase. In Section 3.4.3 this suggestion is taken up. Furthermore, to generate collective plural NPs, (i.e., NPs such as 'the committee', which apply to a group of referents but not to any of the individual members) van Deemter (2002) proposes some changes: the context set C is initialized as the powerset[6] of the domain D and both the target set and the list of properties are defined as sets of sets. See also Stone(1999; 2000) on the description of collective and distributive plural NPs.

3.4.2 Context and Relative Properties

In Section 3.4.1 the generation of a referring expression for $S = \{d_4, d_5, d_6\}$ resulted in $L = \{\langle\ size, large\ \rangle, \langle\ type, block\ \rangle\}$. So far all objects in S are represented with the property $\langle\ size, large\ \rangle$. Actually, d_6 is a bit larger than d_4 and d_5 and therefore may be better represented as *extra large*. But what if Example Domain III in Figure 3.9 is extended with another large block, even larger than d_6? Then probably d_4 and d_5 are not that large anymore. To avoid the need for an update of the KB for different domains, Dale and Reiter(1995, section 5.1.2) suggest to use absolute values to define relative properties. Accordingly, the KB in Figure 3.10 defines the objects in Figure 3.9, where the values for the attribute *size* are defined in centimeters as suggested by van Deemter (2000).

[6] The set of possible subsets of C.

$$
\begin{aligned}
P_{d_1} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ white\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 1cm^2\ \rangle\ \} \\
P_{d_2} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ white\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 2cm^2\ \rangle\ \} \\
P_{d_3} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ black\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 2cm^2\ \rangle\ \} \\
P_{d_4} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ black\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 4cm^2\ \rangle\ \} \\
P_{d_5} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ grey\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 4cm^2\ \rangle\ \} \\
P_{d_6} &= \{\ \langle\ type,\ block\ \rangle,\ \langle\ color,\ white\ \rangle,\ \langle\ shape,\ square\ \rangle,\ \langle\ size,\ 5cm^2\ \rangle\ \}
\end{aligned}
$$

Figure 3.10: KB for Example Domain III in Figure 3.9 with absolute values.

With this KB the Incremental Algorithm can output property sets L like $\{\langle\ color,\ black\ \rangle,\ \langle\ size,\ 4cm^2\ \rangle,\ \langle\ type,\ block\ \rangle\}$ which may be realized as: 'the $4cm^2$ black block'. The resulting referring expressions are not so appealing, because they are not very natural and therefore difficult for the hearer to interpret. To solve this van Deemter (2000) proposes an extension to the Incremental Algorithm which uses the proportional information that can be extracted from the exact numerical values in the KB. In Figure 3.11 the inequalities that can be derived for the property *size* for the objects in the database of Figure 3.10 are presented in the form $\mathbf{Size}(x) > n$.

(1) $\mathbf{Size}(d_6) > 4cm^2$

(2) $\mathbf{Size}(d_4),\ \mathbf{Size}(d_5),\ \mathbf{Size}(d_6) > 2cm^2$

(3) $\mathbf{Size}(d_2),\ \mathbf{Size}(d_3),\ \mathbf{Size}(d_4),\ \mathbf{Size}(d_5),\ \mathbf{Size}(d_6) > 1cm^2$

Figure 3.11: Derived proportions for KB in Figure 3.10.

The Incremental Algorithm itself need not be modified to deal with these derived proportions. As an illustration of the generation process that uses these inequalities, let's take d_6 in Example Domain III of Figure 3.9 as the target object. To describe d_6 the Incremental Algorithm cannot use the properties *type*, *color* or *shape* to rule out all distractors. The property $\langle\ color,\ white\ \rangle$ is included to rule out d_3, d_4 and d_5, but still the distractor set is not empty. When the algorithm encounters the property $\langle\ size,\ 5cm^2\ \rangle$ all remaining distractors are removed from C. The implication of the corresponding proportional derivation $size(d_6) > 4cm^2$ is that d_6 is the only object in C larger than $4cm^2$ which can be characterized by means of a superlative in the final set $L = \{\langle\ color,\ white\ \rangle,\ \langle\ size,\ largest\ \rangle,\ \langle\ type,\ block\ \rangle\}$, in which also *type* is included. A possible realization is 'the largest white block'.

van Deemter (2000) combines the proportional information with the Plural Algorithm (Figure 3.8) with the effect that plural NPs can be generated that contain at least one relative property. For example, suppose the set $S = \{d_4, d_5, d_6\}$ has to be described, where the blocks are not of exactly the same size. From line (2) of Figure 3.11 it can be inferred that all objects in S have the property $> 2cm^2$. From both line (1) and (2) it can be inferred that there is no block in C larger than any of the members of S. The algorithm provides the set $L = \{\langle$ *size*, *largest* $\rangle$, $\langle$ *type*, *block* $\rangle\}$ as described for the singular example above. To describe S, L can be realized as 'the three largest blocks' or 'the large blocks' depending on pragmatic principles van Deemter(2000, section 5.2).

The automatic conversion from absolute to relative values, results in the generation of context-sensitive descriptions. This method, however, can cause ambiguity. For instance the more relative properties are included in the descriptions, the harder the interpretation. For example to interpret 'the heavy large block' leads to uncertainty on what is meant: a block that is heavy compared to large blocks or a block that is large compared to the heavy blocks. See also the example above in which the scope of the property *largest* is ambiguous. The target can be the largest block in the domain, or the largest block of only the white blocks. However, in Example Domain III the description denotes d_6 in both cases. Ambiguity can also arise in combination with the notion of salience (Krahmer and Theune, 2002) introduced in Section 3.5.1. If the target is large compared to the salient objects in the context set but not really large compared to the objects in the domain then 'the large block' is perhaps not such an adequate description.

3.4.3 Negations and Disjunctions

As a demonstration of the need for negations and disjunctions, consider the set $\{d_3, d_4, d_5\}$ in Figure 3.9 as the target set. These three objects only share their *type* property, which is not distinguishing in the context set. The Incremental Algorithm fails here, although S is referable in human communication. There are at least two possibilities to describe S: (1) Using a negation; 'the blocks that are not white'; and (2) using a disjunction of properties, 'the black and the grey blocks'. van Deemter (2002) shows with the **Boolean Algorithm**, as an extension of the Plural Algorithm, that negation and disjunction can be used to find adequate referring expressions in cases where the Incremental Algorithm fails. The use of negations and disjunctions is especially relevant in the generation of plural NPs, since sets of objects that do not have distinguishing properties in common, cannot be described with conjunctive combinations.

For the generation of referring expressions with negations, the KB has to be modified. Negated values are already implicitly available in the KB. For instance, if for an object the value for the attribute *color* is represented as *black*, it can be inferred that the object is *not white* and *not grey*. In Figure 3.12, a KB for Example Domain III in Figure 3.9 is presented that makes these implicit negations explicit by including negated values (indicated by $\neg$). For reasons of presentation the property *size* is left out in the KB. A consequence of this alteration of the KB is that the list of preferred attributes $\mathcal{A}_{\mathcal{S}}$ as used in the Incremental Algorithm is not directly applicable, because attributes are possibly represented more than once for each object. van Deemter (2002) proposes to use an ordered list of values, $\mathcal{V}_{\mathcal{S}}$. On top of the preferred order of the attributes (absolute before relative) in $\mathcal{V}_{\mathcal{S}}$, van Deemter (2002) suggests a preference of positive values over negative values. But what order of preference should be chosen if there are more possible values per attribute? Is black preferred over white or white over black, and what about grey? This question is left unanswered and an arbitrary order of the values is chosen for the attribute *color*.

$$\begin{aligned}
P_{d_1} &= \{\,\langle type,\, block\rangle,\ \langle color,\, white\rangle,\ \langle color,\, \neg black\,\rangle,\ \langle color,\, \neg grey\ \rangle,\ \langle shape,\, square\rangle\,\}\\
P_{d_2} &= \{\,\langle type,\, block\rangle,\ \langle color,\, white\rangle,\ \langle color,\, \neg black\,\rangle,\ \langle color,\, \neg grey\ \rangle,\ \langle shape,\, square\rangle\,\}\\
P_{d_3} &= \{\,\langle type,\, block\rangle,\ \langle color,\, black\,\rangle,\ \langle color,\, \neg white\rangle,\ \langle color,\, \neg grey\ \rangle,\ \langle shape,\, square\rangle\,\}\\
P_{d_4} &= \{\,\langle type,\, block\rangle,\ \langle color,\, black\,\rangle,\ \langle color,\, \neg white\rangle,\ \langle color,\, \neg grey\ \rangle,\ \langle shape,\, square\rangle\,\}\\
P_{d_5} &= \{\,\langle type,\, block\rangle,\ \langle color,\, grey\ \rangle,\ \langle color,\, \neg black\,\rangle,\ \langle color,\, \neg white\rangle,\ \langle shape,\, square\rangle\,\}\\
P_{d_6} &= \{\,\langle type,\, block\rangle,\ \langle color,\, white\rangle,\ \langle color,\, \neg black\,\rangle,\ \langle color,\, \neg grey\ \rangle,\ \langle shape,\, square\rangle\,\}
\end{aligned}$$

Figure 3.12: KB with negated values for Example Domain III Figure 3.9.

In cases where the algorithm does not succeed in distinguishing a target with conjunctive combinations of positive and negative values, disjunctions are used to rule out the distractors. The Boolean Algorithm uses an ordered list of disjunctions of positive and negative values. Not all possible disjunctions have to be tried. For example disjunctions of the type $\neg black \cup black$, $\neg black \cup white \cup black$ and $\neg black \cup white \cup \neg black$ can logically be skipped, since a simple check can avoid contradictory or redundant occurrences of values within one disjunction. Furthermore, in the block domain used here, disjunctions of the type $\neg black \cup white$ can be omitted because instead objects can be described with the single value *white* or the disjunction $white \cup grey$, which are both more preferred because they do not contain a negation. The problem of determining a preferred order of the multiple values of one attribute of course reappears and increases in the determination of a preferred order of disjunctions. Figure 3.13 presents a possible order of all combinations of two properties that can be used by the Boolean Algorithm while describing a target set in the block domain. Since for the Boolean Algorithm the inclusion of the property *type* is postponed to the realization phase, *block* is not included in the preferred list of attributes or any of its disjunctions.

$$
\begin{aligned}
\mathrm{P} = \langle\ & \mathit{white} \cup \mathit{black},\ \mathit{white} \cup \mathit{grey},\ \mathit{black} \cup \mathit{grey},\ \mathit{white} \cup \mathit{square}, \\
& \mathit{black} \cup \mathit{square},\ \mathit{grey} \cup \mathit{square},\ \mathit{white} \cup \mathit{small},\ \mathit{white} \cup \mathit{large}, \\
& \mathit{black} \cup \mathit{small},\ \mathit{black} \cup \mathit{large},\ \mathit{grey} \cup \mathit{small},\ \mathit{grey} \cup \mathit{large}, \\
& \mathit{square} \cup \mathit{small},\ \mathit{square} \cup \mathit{large},\ \mathit{white} \cup \neg\mathit{square},\ \mathit{black} \cup \neg\mathit{square}, \\
& \mathit{grey} \cup \neg\mathit{square},\ \mathit{white} \cup \neg\mathit{small},\ \mathit{white} \cup \neg\mathit{large},\ \mathit{black} \cup \neg\mathit{small}, \\
& \mathit{black} \cup \neg\mathit{large},\ \mathit{grey} \cup \neg\mathit{small},\ \mathit{grey} \cup \neg\mathit{large},\ \mathit{square} \cup \neg\mathit{small}, \\
& \mathit{square} \cup \neg\mathit{large},\ \mathit{small} \cup \neg\mathit{square},\ \mathit{large} \cup \neg\mathit{square},\ \mathit{square} \cup \neg\mathit{white}, \\
& \mathit{square} \cup \neg\mathit{black},\ \mathit{square} \cup \neg\mathit{grey},\ \mathit{small} \cup \neg\mathit{white},\ \mathit{large} \cup \neg\mathit{white}, \\
& \mathit{small} \cup \neg\mathit{black},\ \mathit{large} \cup \neg\mathit{black},\ \mathit{small} \cup \neg\mathit{grey},\ \mathit{large} \cup \neg\mathit{grey}, \\
& \neg\mathit{white} \cup \neg\mathit{square},\ \neg\mathit{black} \cup \neg\mathit{square},\ \neg\mathit{grey} \cup \neg\mathit{square}, \\
& \neg\mathit{white} \cup \neg\mathit{small},\ \neg\mathit{white} \cup \neg\mathit{large},\ \neg\mathit{black} \cup \neg\mathit{small},\ \neg\mathit{black} \cup \neg\mathit{large}, \\
& \neg\mathit{grey} \cup \neg\mathit{small},\ \neg\mathit{grey} \cup \neg\mathit{large},\ \neg\mathit{square} \cup \neg\mathit{small}, \\
& \neg\mathit{square} \cup \neg\mathit{large}\ \rangle
\end{aligned}
$$

Figure 3.13: An ordered list of disjunctive combinations of two properties.

Following van Deemter (2002), the Plural Algorithm is used as a basis for the boolean extensions. The Boolean Algorithm searches for a distinguishing description for a set of objects, S, using a globally accessible KB with positive and negative values for all objects in the context set C. In comparison to the Plural Algorithm, the Boolean Algorithm involves an extra recursive step with which the algorithm iterates through different lists of values and disjunctive combinations of values. The algorithm adds properties or disjunctions of properties to L only if they describe the objects in S and at the same time rule out distractors. In each iteration it is checked whether the target is distinguished. If this is the case, L is returned; otherwise the set of preferred values is modified. In the first iteration the list of preferred values, $\mathcal{V}_S$, is used to distinguish S. In the second iteration the list of preferred values is transformed into a list of all possible disjunctions of two values of $\mathcal{V}_S$. In the third iteration the list of preferred values consists of all possible disjunctions of three values of the values of $\mathcal{V}_S$ and so on, until the disjunctions consist of the number of values that equals the number of attributes has been tried (failure) or the target object is distinguished (success).

In Figure 3.14 the Boolean Algorithm is presented as an extension of the Plural Algorithm as shown in Figure 3.8 (minus the inclusion of the type property line (5) and (6)). For the recursion a counter i is initialized in line (1) and a new list variable P is introduced and initialized as $\mathcal{V}_S$ in line (2). In line (3) the iteration is started on the condition that the value of i does not exceed the number of values in $\mathcal{V}_S$. The Plural Algorithm operates as before. If no distinguishing description is found with the use of single values, the function *MakeNewDisjunctions*, is called in line (4), which assigns a new list of disjunctions of values to P. *MakeNewDisjunctions* is presented in line (6) to (12). A new list, *NewP* is initialized in line (7). Each single value X of P (line (8)) is combined with all the values in $\mathcal{V}_S$ (line (9)), as long as contradictions and redundancies of the values within the disjunctions are avoided. The disjunctions of each two values formed in line (10), are added to *NewP* in line (11). If all possible disjunctions are generated, *NewP* is returned in line (12). The counter i is increased with 1 in line (5). Successively, a second iteration of the Boolean Algorithm is started with a list of disjunctions of two values. If in this iteration no distinguishing description is found a new list of disjunctions is generated. Now disjunction X of P is combined with each of the preferred values of $\mathcal{V}_S$, which results in a list of disjunctions each containing three values. The Boolean Algorithm keeps extending the disjunctions in P until a distinguishing description is found or until the counter i is larger than the number of values in $\mathcal{V}_S$ and all combinations have been tried.

```
        GenerateDistinguishingDescription($S, C$)
          $L := \emptyset$
(1)       $i := 1$
(2)       $P := \mathcal{V}_S$
(3)         while $i \leq |\mathcal{V}_S|$ do
              foreach $V \in P$ do
                if $S \subseteq [\![ \langle A, V \rangle ]\!]$ and $C \not\subseteq [\![ \langle A, V \rangle ]\!]$ then
                  $L := L \cup \{ \langle A, V \rangle \}$
                  $C := C \cap [\![ \langle A, V \rangle ]\!]$
                endif
                if $C = S$ then
                  return $L$
                endif
              end foreach
(4)           $P :=$ MakeNewDisjunctions($P$)
(5)           $i := i + 1$
            end while
          return failure

(6)     MakeNewDisjunctions($P$)
(7)       $NewP := \emptyset$
(8)         foreach $X \in P$ do
(9)           foreach $Y \in \mathcal{V}_S$ do
(10)            $Disjunct := X \cup Y$
(11)              $NewP := NewP \cup Disjunct$
              end foreach
            end foreach
(12)      return $NewP$
```

Figure 3.14: Pseudocode Boolean Algorithm.

As an example, take the set $\{d_3, d_4, d_5\}$ as the target S. The algorithm starts in the first iteration with the set of preferred values $\mathcal{V}_S = \langle$ *white*, *black*, *grey*, *square*, *small*, *large*, $\neg$*white*, $\neg$*black*, $\neg$*grey*, $\neg$*square* $\neg$*small*, $\neg$*large* $\rangle$. When the algorithm encounters the value $\neg$*white* it finds that this value describes all objects in S and at the same time rules out all distractors. Successively this value is added to L. In the realization phase the *type* property is added, which results in a possible description: 'the blocks that are not white'.

In the beginning of this section it was suggested that the set $\{d_3, d_4, d_5\}$ in Figure 3.9 can be referred to using a negation, 'the blocks that are not white' or a disjunction of properties, 'the black and the grey blocks'. In Example Domain III any set of objects can be described by single properties or the single negated properties and their conjunctions. Since single negated properties are preferred over disjunctions of properties, the Boolean Algorithm always encounters success in the first iteration in describing any set in the domain. But is the set $\{d_3, d_4, d_5\}$ really best described using negation? At least in this case a disjunction, 'the black and the grey blocks', seems very natural.

A drawback of the Boolean Algorithm is that if a distinguishing description is not found in the first iteration, the number of disjunctive properties per disjunction increases per iteration and the Boolean Algorithm rapidly becomes intractable. To derive a polynomial runtime, van Deemter(2002, page 48) suggests pruning the algorithm at some point. The algorithm enriched with boolean extensions is still incremental; once a property or a disjunction is added to L it is realized in the final description. But, along with the increase of the number of disjunctive properties, the resulting descriptions become increasingly more redundant. To avoid these long descriptions, Gardent (2002) proposes a constraint-based algorithm that produces minimal descriptions with the inclusion of disjunctions and negations (see also Gardent et al., 2003). Constraint-based programming is used to efficiently solve the NP-hard problem of finding the shortest description; but according to Horacek(2003, page 104) it still takes a lot of time for rather simple problems. Horacek (2003) suggests a best-first search algorithm that uses a reduced search space and linguistic preferences (c.f., (Horacek, 2004)). Another solution is offered by Varges and van Deemter (2005) based on the algorithm by Varges (2004). A breadth-first search strategy is performed on a tree representation of all possible referring expressions in the domain. This tree is built by applying inference rules on a domain representation that extend the object descriptions to descriptions involving quantification.

3.4.4 Locative Relata and Physical Context

The Incremental Algorithm provides only for referring expressions that describe objects that can be distinguished in terms of their properties. As shown in Section 3.3.3, in cases where the target object cannot be described with its properties, the physical context might be used to single out a target. In this section, the possibility of an extension of the Incremental Algorithm is examined that is able to include locative relata in referring expressions. Objects that are located within a 'small distance' of each other are determined to be **spatially related** (e.g., Horacek, 1995; Krahmer and Theune, 2002). For the current Example Domain III this 'small distance' is interpreted in such a way that two objects are spatially related only if there is no other object located in between these two objects. A refined definition is given in Section 3.5.2, but see Tenbrink (2004) for a more elaborated approach on spatial relations.

Krahmer and Theune (2002) suggest extending the Incremental Algorithm with locative relata by enriching the list of preferred attributes with spatial relations. In adding spatial relations to the ordered list of preferred attributes, Krahmer and Theune make two assumptions: (1) Properties precede relations, because it takes more effort to describe and interpret other objects besides the target (e.g., Clark and Wilkes-Gibbs, 1986; Zipf, 1949); and (2) Spatial relations are preferred over any other relations, because they can be perceived in contrast to, for example, possessive relations. In accordance with these assumptions, the ordered list of preferred attributes in the block domain is ordered as follows: $\langle$ *type*, *color*, *shape*, *size*, *spatial* $\rangle$. In addition, Krahmer and Theune(2002, page 253) (see also Theune, 2000, page 127) propose a subsumption hierarchy of spatial relations in which less specific relations subsume more specific relations. For example *left of* and *right of* are both subsumed by the relation *next to*. In Figure 3.15, the KB for Example Domain III is presented that includes the most specific spatial relations, assuming that less specific values can be derived from more specific ones. For every object d the new set PR_d is presented as a union of the property set P_d and the set of relations $R = \{ \langle \textit{spatial}, \textit{left-of}(d') \rangle, \langle \textit{spatial}, \textit{right-of}(d'') \rangle \}$. In Example Domain III the objects d_2 to d_5 all have two locative relata that have no preference over each other. However, with an integration of linguistic salience weights as discussed in Section 3.5.1, relata can be ordered according to their salience weight. Initially this has no effect, but when discourse progresses the most salient relatum can be chosen as a spatial relation on the basis of salience weights.

$$
\begin{array}{ll}
PR_{d_1} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, left-of} \ (d_2) \rangle \ \} \\
PR_{d_2} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, left-of} \ (d_3) \rangle, \langle \textit{spatial, right-of}(d_1) \rangle \ \} \\
PR_{d_3} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, left-of} \ (d_4) \rangle, \langle \textit{spatial, right-of}(d_2) \rangle \ \} \\
PR_{d_4} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, left-of} \ (d_5) \rangle, \langle \textit{spatial, right-of}(d_3) \rangle \ \} \\
PR_{d_5} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, left-of} \ (d_6) \rangle, \langle \textit{spatial, right-of}(d_4) \rangle \ \} \\
PR_{d_6} & = \{ P_1, \ldots, P_n, \langle \textit{spatial, right-of}(d_5) \rangle \ \}
\end{array}
$$

Figure 3.15: KB with spatial relations for Example Domain III Figure 3.9.

In the approach of Krahmer and Theune (2002), spatial relations are added to a description in essentially the same way as properties. The algorithm iterates through the extended list of preferred attributes, adding properties and relations to L if they rule out any distractors. The Incremental Algorithm is slightly adjusted to accommodate the addition of spatial relations. If a spatial relation can be used to rule out one or more distractors, a recursive call to the Incremental Algorithm provides a distinguishing description of the relatum. The description of the relatum is integrated in the description of the target. In Figure 3.16 the variant of the Incremental Algorithm that includes spatial relations is presented. In line (1) whether the property added to L expresses a relation between the target and some relatum r', is checked. If this is the case, a recursive call to the algorithm is made, which results in a distinguishing referring expression of r' that is stored in the variable L', in line (2). In line (3) the description of r' is combined with the properties in L.[7]

```
          GenerateDistinguishingDescription($r, C$)

            $L := \emptyset$
              foreach $A \in \mathcal{A}_r$ do
                if $r \in [\![ \langle A, V \rangle ]\!]$ and $C \not\subseteq [\![ \langle A, V \rangle ]\!]$ then
                  $L := L \cup \{ \langle A, V \rangle \}$
                  $C := C \cap [\![ \langle A, V \rangle ]\!]$
(1)               if $V$ expresses a relation with $r'$ then
(2)                 $L'$ :=GenerateDistinguishingDescription($r', C$)
(3)                 $L := L \cup L'$
(4)               endif
                endif
                if $C = \{r\}$ then
                  if $\{ \langle type, V_r \rangle \} \notin$ L then
                    $L := L \cup \{ \langle type, V_r \rangle \}$
                  endif
                return $L$
                endif
              end foreach
            return failure
```

Figure 3.16: Pseudocode Incremental Algorithm with spatial relations.

There are several problems with this proposal. One is that the incremental design of the algorithm implies that if a relation does not suffice to rule out all distractors, the next relation in the list is tried. The lack of backtracking causes all relations that rule out at least one distractor to be realized. In human communication such extensive referring expressions are not plausible.

[7] Here a simplified pseudocode is presented, compare Krahmer and Theune (2002) for a more detailed version.

A more general problem with the inclusion of spatial relations is that **infinite recursions** might arise. As an illustration of the algorithmic problem of infinite recursion let's describe d_2 in Example Domain II in Figure 3.7. Of the properties, only ⟨ *color*, *white* ⟩ is added to L because it rules out the distractors d_1 and d_4. Because d_3 is not ruled out, a relation for example ⟨ *spatial*, *right-of*(d_1) ⟩ is selected. Successively the recursive step in line (2) in Figure 3.16 is initiated to generate a description for the relatum d_1. To describe d_1 the property ⟨ *color*, *black* ⟩, which rules out d_2 and d_3, is not satisfying because d_4 is still in the context set. To remove d_4 from C, a spatial relation is selected. The only available relation is ⟨ *spatial*, *left-of*(d_2) ⟩, which again results in a recursion to describe d_2. As a result the algorithm gets stuck in the generation of an infinite NP, in this case 'the white block to the right of the black one to the left of the white one to the right of ...etc.'. In describing a greedy GRE algorithm, Dale and Haddock(1991, page 166) suggest a heuristic to prevent this looping: "do not express a given piece of information more than once within the same NP". But arguably this is somewhat ad hoc. Another solution to the recursion problem is proposed by Varges (2004), who advocates performing a breadth-first search for the cheapest referring expression in a tree representation of all possible referring expressions. On the basis of inference rules a tree is built that contains all referring expressions including relations that can be used to refer to the objects in the domain. The tree is then filtered in order to rule out all unwanted solutions. By employing cost functions, recursive relations are avoided.

Another problem is the efficiency of the algorithm. The Incremental Algorithm is efficient because there is no possibility of backtracking. Unfortunately, as soon as relations are included, this advantage cannot be kept: the generation of relational descriptions is NP complete (e.g., Krahmer et al., 2001). However, two factors need to be noted. First, the use of salience can guide the search for a relatum. Following the empirical findings of Beun and Cremers (1998), Krahmer and Theune suggest choosing only salient relata, which in most cases offers a substantial reduction of the search space. Second, an upper bound can be defined to the number of properties and relations which can be included in the final description; as soon as such an upper bound is defined, tractability is regained (van Deemter, 2001). In summary, the integration of spatial relations in the Incremental Algorithm is not straightforward. Especially computational efficiency and human plausibility, which are the most appealing features of the algorithm, are hard to keep. Chapter 4 will show how the graph-based approach of Krahmer et al. (2003) solves these problems. But first the next section explores how the Incremental Algorithm can be extended with a notion of salience to produce descriptions that are more context-sensitive.

3.5 Salience

So far the Incremental Algorithm generates evoking expressions (see Section 2.5.1) which distinguish the target from every object in the domain. In contrast, in human communication, if a target object is salient, this generally leads to a reduction of the search space (assuming that not all objects are equally salient). The distractors from which the target object has to be distinguished need not all be objects in the domain, but only those that are at least as salient as the target object. Consequently, fewer properties are needed for ruling out the distractors. Krahmer and Theune (2002) enriched the Incremental Algorithm with a notion of linguistic salience. Yet, apart from the linguistic context, speakers also use the visual context in identifying objects, for instance in their use of relative properties and relata (see respectively Section 3.4.2 and 3.4.4). In this Section a notion of visual salience is proposed, which combines the inherent salience of objects and the focus of attention.

Inherent salience applies to objects that stand out perceptually with respect to the rest of the domain. Beun and Cremers assume that an object is inherently salient if it is the only object in the domain which has a particular property. They claim that inherently salient objects are referred to by **reduced descriptions**; i.e., descriptions which contain less properties than are strictly speaking required to generate a fully distinguishing description. In contrast, Horacek (1997), for instance, argues for the exact opposite: one should use the property that makes the object inherently salient, even if it does not rule out the other objects in the domain. For example: a single pink elephant should be referred to as 'the pink elephant' even if the other objects all appear to be flamingos. Arguably, world knowledge (that elephants are typically grey) plays an important role in this case, but not for the examples that Beun and Cremers discuss. This suggests that the respective positions of Beun and Cremers and Horacek do not really contradict each other, but apply to different cases. However, more research is required to test this hypothesis. Another reason for which an object might be more salient is that it is closely located to another salient object. In this respect (Beun and Cremers, 1998, page 127) introduce a notion of **focus of attention**: objects that are located close to some salient object are easier to describe, while at the same time objects in the neighborhood of a target object become more salient. The notion of a focus of attention is not only psychologically plausible, but is also beneficial from a computational point of view. By defining the focus of attention as a subset of the objects in the whole domain, the search space of the algorithm is reduced. It is safe to assume that objects that are currently in the focus of attention are more salient than objects that are not in focus.

Adapting the algorithm in this way to the context has a twofold effect: (1) The generated referring expressions become context-sensitive; and (2) The search space can be reduced. In this section it is proposed to fuse the notion of linguistic salience as proposed by Krahmer and Theune (2002) with the two-dimensional notion of the visual context as presented by van der Sluis and Krahmer (2001), see also (van der Sluis, 2001). In preparation of this three-dimensional notion of salience, the three notions of salience are discussed separately. In Section 3.5.1 the notion of linguistic salience is discussed as proposed by Krahmer and Theune (2002) and in Section 3.5.2 the visual notions of inherent salience and focus space salience are defined. In Section 3.5.3 a three-dimensional notion of salience is presented as a combination of linguistic and visual salience. This three-dimensional definition of salience is illustrated with a worked example in Section 3.5.4.

3.5.1 Linguistic Salience

The input of the Incremental Algorithm consists of a target object r and a context set C, where C is a subset of the domain of conversation. But which subset? In the description of the Incremental Algorithm, it is not explained how the context set is constructed. Dale and Reiter(1995, page 236) merely refer to the set of entities in focus spaces of the discourse focus stack in terms of the theory of discourse structure by Grosz and Sidner (1986) (i.e., the entities the hearer is assumed to attend to). In human conversation a **discourse history** is built in which the objects that already have been mentioned are stored. The idea is that once an object has been mentioned, it is linguistically salient and re-referring to this object can be done using a reduced, anaphoric description. For example in Figure 3.7, where d_2 has just been described as 'the white block to the right of the black one', it can be immediately referred to a second time with 'the block' or 'the white block'. At some point, however the objects that have been talked about some time ago, become less salient. Consequently the set of salient objects changes during a discourse: recently mentioned, highly salient objects are to be stored in the context set and less salient objects removed from the context set (this is also called **recency**).

Similar to the notion of a discourse history, Krahmer and Theune (1998, 2002) formalize the construction of the context set during a discourse and thereby enrich the Incremental Algorithm with a notion of linguistic context (c.f., Passonneau, 1996; Jordan, 2002). They propose a more specific definition of the set C with the use of salience weights. The linguistic salience of an object is modeled using a **salience weight function** (notated $\mathbf{Sw}(d,\ s)$, i.e., the salience weight of object d in state s), according to which a salience weight is assigned to each object in the domain of conversation. The salience weights have to be updated after each utterance. With the addition of salience weights, the context set can be specified as the target object r together with all objects in the domain having a salience weight higher than or equal to r. The context set C in state s_i can more formally be defined as:

$$\{d \in D \mid \mathbf{Sw}(r, s_i) \leq \mathbf{Sw}(d, s_i)\}$$

In the beginning of a discourse all objects are assumed to be equally salient, having a salience weight of 0. When discourse progresses, more objects have been mentioned and the number of distractors consequently decreases compared to the total number of objects in the conversational domain. This implies that when the target object is in some way salient, the search space is reduced. Hence, generally, fewer properties are required to rule out the distractors.

The salience weight function Krahmer and Theune (2002, page 242) propose is determined on the basis of the ranking of forward looking centers according to Centering Theory (Grosz et al., 1995), augmented with a notion of recency derived from Hajičová (1993). Linguistic salience weights range from 0 (minimum salience) to 1 (maximum salience). The linguistic salience function defined below calculates the salience weight of each object d in a state s_i. In the initial state s_0, the beginning of the discourse, no object has been described. Therefore, initially each object in the domain has a weight of zero. In this definition, $C_f(U_i)$ is the order of the forward looking centers of U_i (the sentence uttered at time i) according to Centering Theory. This order is such that the syntactic subject of U_i is the most salient (mapped to salience weight 1) followed by the indirect object (mapped to 0.9) and the other objects (mapped to 0.8). Thus, more formally, $\mathbf{Level}(d_i, \langle\ d_0, d_1 \ldots, d_n\ \rangle) = \mathbf{Max}(0, 1 - 0.i)$, where $\langle\ d_0, \ldots, d_n\ \rangle$ is the ordered set of forward looking centers of the relevant utterance. If an object is not mentioned in U_i its salience weight is reduced with 0.1, unless it is already 0.

$$\mathbf{Sw}(d, s_{i+1}) = \begin{cases} \mathbf{Level}(d, C_f(U_i)) & \text{if } d \in C_f(U_i) \\ \mathbf{Max}(0, \mathbf{Sw}(d, s_i) - 0.1) & \text{if } d \notin C_f(U_i) \text{ and } d \in C_f(U_j), j \leq i \end{cases}$$

To demonstrate the workings of the Incremental Algorithm enriched with the above salience function, the context set C in Figure 3.9 is used, together with the corresponding KB that contains the objects d_1 to d_6 as displayed in Figure 3.10. With the use of linguistic salience weights, the generated referring expressions become more sensitive to the discourse context. The sequence of three utterances presented below can be produced with the generated referring expressions in the following example.

U_1 ='the block in the middle is large and black'
U_2 ='the grey block has the same size'
U_3 ='it is located next to the black block'

In the beginning of a discourse, state s_1, no objects have been talked about and all objects in C are assigned a salience weight of 0. Suppose the target is d_4 for the first utterance, U_1. The Incremental Algorithm is called with the parameters

d_4 and the context set C, where C consists of d_4 together with all objects with a salience weight higher than or equal to the weight of d_4. Thus, in the beginning of a discourse all objects are in the context set. The resulting referring expression for d_4 is constructed along the same lines as before (Section 3.3.2). The algorithm succeeds when it encounters the set $L = \{\langle\ \textit{color}, \textit{black}\ \rangle, \langle\ \textit{size}, \textit{large}\ \rangle, \langle\ \textit{type}, \textit{block}\ \rangle\}$, which can linguistically be realized by the distinguishing description 'the large black block'. After the sentence U_1 has been produced with this description, the salience weights are updated. Following the linguistic salience function defined above, d_4 receives a maximum salience weight of 1, it is the first element of the list of forward looking centers. For all other objects the weights remain 0. The salience weights are updated as presented below.

$$\begin{array}{rlcrl}
\mathbf{Sw}(d_1, s_1) = & 0 & \qquad & \mathbf{Sw}(d_4, s_1) = & 1 \\
\mathbf{Sw}(d_2, s_1) = & 0 & & \mathbf{Sw}(d_5, s_1) = & 0 \\
\mathbf{Sw}(d_3, s_1) = & 0 & & \mathbf{Sw}(d_6, s_1) = & 0
\end{array}$$

The next utterance in the discourse, U_2, contains a reference to d_5. In this case, C again contains all objects $d_1 \ldots d_6$, since all have a salience weight higher than or equal to that of the target. The algorithm succeeds with the set L $\{\langle\ \textit{color}, \textit{grey}\ \rangle, \langle\ \textit{type}, \textit{block}\ \rangle\}$, which can be realized as for example 'the grey block' as produced in the second sentence U_2. Accordingly, the salience weights are updated to state s_2 as presented below: for d_4 the weight is decreased with 0.1 and d_5 receives the maximum of 1.

$$\begin{array}{rlcrr}
\mathbf{Sw}(d_1, s_2) = & 0 & \qquad & \mathbf{Sw}(d_4, s_2) = & 0.9 \\
\mathbf{Sw}(d_2, s_2) = & 0 & & \mathbf{Sw}(d_5, s_2) = & 1 \\
\mathbf{Sw}(d_3, s_2) = & 0 & & \mathbf{Sw}(d_6, s_2) = & 0
\end{array}$$

Suppose now that it is needed to express that d_5 is located next to d_4. First a call to the algorithm is made to refer to d_5. The context set C contains only d_5 because there is no object in the domain that has a salience weight higher than or equal to that of d_5. Correspondingly, the algorithm generates the set L: $\{\langle\ \textit{type}, \textit{block}\ \rangle\}$. Due to the pronominalization rule proposed by Krahmer and Theune (2002), in the sequential sentence U_3, the grey block can be referred to with the pronoun 'it'. The rule for pronominalization Krahmer and Theune propose provides for the generation of pronouns in cases where the target object r is the most salient entity in the domain and the linguistic context contains an antecedent for r (see also Theune, 2000, page 123). Furthermore, the algorithm is called again to generate once more a description for d_4. The salience weight of d_4 is 0.9, therefore $C = \{d_4, d_5\}$. To distinguish d_4 from its only distractor, the Incremental Algorithm succeeds with the set $\{\langle\ \textit{color}, \textit{black}\ \rangle, \langle\ \textit{type}, \textit{block}\ \rangle\}$. Instead of 'the large black block' as generated for U_1, the description for a second reference to d_4 can now be realized as 'the black block'. Although Figure 3.9 contains more black blocks, in the third sentence U_3, the linguistic context of the generated referring expression determines the target to be d_4. An update of the salience weights results in the same distribution as in state s_2.

$$\begin{array}{rrrr} \mathbf{Sw}(d_1, s_3) = & 0 & \mathbf{Sw}(d_4, s_3) = & 0.9 \\ \mathbf{Sw}(d_2, s_3) = & 0 & \mathbf{Sw}(d_5, s_3) = & 1 \\ \mathbf{Sw}(d_3, s_3) = & 0 & \mathbf{Sw}(d_6, s_3) = & 0 \end{array}$$

3.5.2 Visual Salience

Apart from linguistic salience, there are other reasons for which objects can be more prominent than the other objects in a domain of conversation. For instance, an object can be inherently salient because it has a peculiar property compared to the other objects in the domain (Beun and Cremers, 1998, page 127). But also, objects that are located close to highly salient objects may be viewed as somewhat salient, because they are located in the focus of attention (Beun and Cremers, 1998, page 127). Below, visual salience is modeled as a combination of focus space salience and inherent salience as it is done by van der Sluis and Krahmer (2001).

Inherent Salience
There are various ways to determine inherent salience; (see Cremers, 1996, page 24 for references and discussion). Here, a strong criterion is opted for: an object is inherently salient only if for some attribute it has a particular value V_i while the other objects in the domain all have a different value V_j for that particular attribute (where $i \neq j$).

Focus Space Salience
In this section first an insight is given into how the current focus of attention is defined using an example, before a formal definition is presented. The focus of attention is defined as a **focus space** which consists of the last mentioned object o and the set of objects directly related to o. As defined in Section 3.4.4, an object d is standing in a direct relation to the object o if d is the closest object to o for which that particular relation holds. The set of objects related to o can be illustrated with Figure 3.17. Suppose the last mentioned object is d_2. In this case the current focus space contains three objects as depicted in Figure 3.17: the object o (i.e., d_2) and the objects directly related to o, d_1 and d_3. Object d_4 is excluded from the focus space, because object d_3, standing in the same spatial relation to d_2, is located closer to d_2.

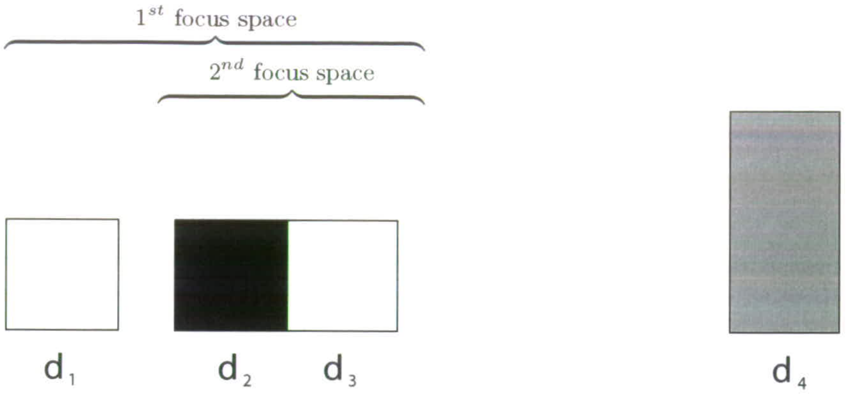


Figure 3.17: Focus spaces.

Suppose the current target object is d_3. Once the algorithm has generated a referring expression for d_3, the focus space needs to be updated. The updated focus space contains d_3 and the set of objects that are directly related to d_3, which are the objects d_2 and d_4. However, on a second note object d_4 seems rather far apart from the current focus. To be able to take into account the relative distance between the objects in the domain of discourse, the notion of **perceptual grouping** (Thorisson, 1994) is used. Thorisson defines a **Proximity Score**, in which the distance of each object in the domain to a particular object o (notated **Dist**(o,d)), is weighted against the maximal distance between o and some object y in the domain D. Thus, the Proximity Score is a function which is defined as follows:

$$\mathbf{Ps}(o, d) = \frac{\mathbf{Dist}(o, d)}{\underset{y \in D}{\mathbf{Max}} \ (\mathbf{Dist}(o, y))}$$

By setting a threshold to this fraction, far away objects can be excluded from the focus space of o. For example, consider Figure 3.17 again, and suppose, for the sake of illustration, that the threshold is set to 0.5. To measure the distances between the objects in Example Domain IV let's assume the blocks to have a base of 1 by 1 cm. Distances are measured from base center to base center. Assume the respective distances between the most recent target d_3 and the other objects in the domain are: **Dist**(d_3, d_1)= 2.5 cm, **Dist**(d_3, d_2)= 1 cm, and **Dist**(d_3, d_4)= 6 cm. Consequently, the maximal distance between object o and any object in the domain, is 6 cm. This results in the following fractions:

$$\begin{array}{lll}\mathbf{Ps}(d_3,d_1) = & 2.5/6 & = 0.42\\ \mathbf{Ps}(d_3,d_2) = & 1/6 & = 0.17\\ \mathbf{Ps}(d_3,d_3) = & 0/6 & = 0\\ \mathbf{Ps}(d_3,d_4) = & 6/6 & = 1\end{array}$$

Hence, d_4 can be excluded from the focus space of d_3 on the basis of perceptual grouping with a threshold of 0.5. To summarize, the new, updated focus space contains d_3 (the last mentioned object) and d_2, as presented formally below. The focus space of an object o is defined as the union of the set that contains the object o with the set of objects in the domain that are spatially related to o (notated $\mathbf{Rel}(o,d)$) and which are not too far away in terms of perceptual grouping with a threshold T as defined above.

$$\mathbf{focusspace}(o) =$$

$$\{o\} \cup \{\ d \in D \mid \mathbf{Rel}(o,\, d) \wedge \neg\exists d'[\mathbf{Rel}(o,\, d') \wedge (\mathbf{Dist}(o,\, d') \leq \mathbf{Dist}(o,\, d))] \wedge T \leq 0.5\}$$

Notice that the target object r need not be an element of the current focus space of o. When it is not, the term **focus shift** applies.

3.5.3 A Three-dimensional Notion of Salience

To define a salience weight in which the visual and the linguistic context information are fused, each object in the domain receives three salience weights: (1) Indicating whether or not the object is linguistically salient; (2) Indicating whether the object is inherently salient; and (3) Indicating the focus space salience of the object. The total salience weight of an object is determined by taking the weighted sum of the three separate salience weights. Arguably, some forms of salience are more important than others. For instance, linguistic salience may be assumed to be of prime importance, in the sense that an object o which has just been described is more salient than an object that is in the current focus space (i.e., close to r) but has not itself been mentioned so far. In a similar vein, an object that is in focus is somewhat more salient than an object that is inherently salient but falls outside the current focus space (as observed by Beun and Cremers, 1998).

The three-dimensional notion of salience for every object d in D in state s_i can be presented as a function $\mathbf{Sw}(d,s_i)$ which sums over the three kinds of salience. Linguistic salience (**L-sw**) is modeled as it is done by Krahmer and Theune (2002), (see Section 3.5.1), who determine linguistic salience on the basis of the ranking of forward looking centers according to Centering Theory (Grosz et al., 1995) augmented with a notion of recency. Linguistic salience weights range from 0 to 1. In the initial state, every object is assigned an **L-sw** weight 0. If an object is inherently salient, it has a constant **I-sw** weight of 1, for all other objects **I-sw** = 0. Finally, focus space salience (**F-sw**) is easily determined given the definition in Section 3.5.2. An object has an **F-sw** weight of 1 iff it is part of the current focus space. The **F-sw** weight 1 is assigned to every object d in the focus space of object o, where o is the most recently described object (or, slightly more general, the object with the highest **L-sw**). Below a formal definition of salience for each object $d \in D$ is presented. The salience weight of each object d in a state s_i is calculated as the weighted sum of the three kinds of salience associated with d in that state, where the three kinds of salience are ordered with respect to their importance. λ_1, λ_2 and λ_3 are the weights with $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and the lambda values themselves are an empirical matter (see below). In the initial state s_0 (the beginning of the discourse) no object has been described and it may be assumed that there is no focus space. Thus, initially, each object in the domain has an **L-sw** and an **F-sw** weight of 0, whereas the inherently salient objects receive a salience weight of 1.

$$\mathbf{Sw}(d, s_i) = \lambda_1 \mathbf{I\text{-}sw}(d, s_i) + \lambda_2 \mathbf{L\text{-}sw}(d, s_i) + \lambda_3 \mathbf{F\text{-}sw}(d, s_i)$$

where:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$$\mathbf{L\text{-}sw}(d, s_0) = 0$$

$$\mathbf{L\text{-}sw}(d, s_{i+1}) = \begin{cases} \mathbf{Level}(d, C_f(U_i)) & \text{if } d \in C_f(U_i) \\ \mathbf{Max}(0, \mathbf{L\text{-}sw}(d, s_i) - .1) & \text{otherwise} \end{cases}$$

$$\mathbf{F\text{-}sw}(d, s_0) = 0$$

$$\mathbf{F\text{-}sw}(d, s_{i+1}) = \begin{cases} 1 & \text{if } d \in \mathbf{Focusspace}(o) \wedge o = \underset{d'}{\mathbf{Max}}\ \mathbf{L\text{-}sw}(d', s_i)) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{I\text{-}sw}(d, s_i) = \begin{cases} 1 & \text{if object } d \text{ is inherently salient} \\ 0 & \text{otherwise} \end{cases}$$

The above defined generic notion of salience can be applied to any GRE algorithm. To integrate this notion in the Incremental Algorithm, Krahmer and Theune (2002) are followed in their restriction of the context set. The context set C only contains the objects that are equal or more salient than the target object. By using salience weights the Incremental Algorithm can restrict the context set in three ways: (1) The algorithm closely monitors the linguistic context to compute the linguistic salience according to Krahmer and Theune (2002); (2)

The algorithm explicitly tracks the focus of attention to compute the focus space salience; and (3) It acknowledges inherently salient objects. As a result, when an object is salient, reduced information can be used. For example, when the target object is part of the current focus space (and is not linguistically salient), the distractors are typically the other objects that are in the current focus space, together with the objects that are linguistically salient. Moreover, when a relatum is needed, a suitable salient object is selected from the salient objects in the context set (as also suggested by Krahmer and Theune (2002) (see Section 3.4.4).

3.5.4 Worked Examples

In this section the three-dimensional notion of salience is illustrated with an example in which a sequence of three referring expressions is generated. For this example it is assumed, based on the assumptions made above, that linguistic salience is of primary importance ($\lambda_1 = 0.7$), the influence of focus space salience is less ($\lambda_2 = 0.2$) and inherent salience is least influential ($\lambda_3 = 0.1$). For this example the context set C, depicted as Example Domain IV in Figure 3.18, is used. The corresponding KB is presented in Figure 3.19, as the union of the set of properties, P, and the set of Relations, R. The sequence of utterances presented below can be produced with the generated referring expressions in this example. The additional focus space salience weights and the inherent salience weights are of importance in choosing relata.

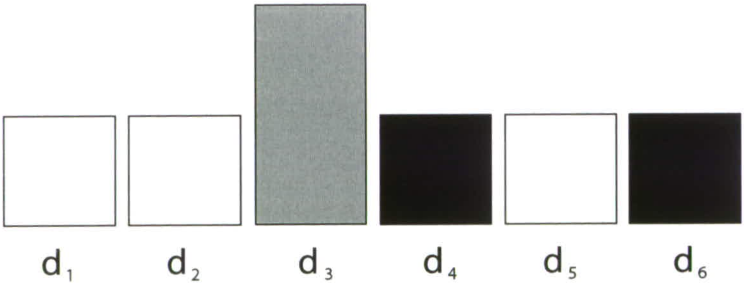


Figure 3.18: Example Domain IV.

U_1 ='look at the white block to the left of the black block'
U_2 ='the black block to the left of it has the same size'
U_3 ='now consider the white block to the left of the grey one'

$$\begin{aligned}
P_{d_1} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{white}\ \rangle, \langle\ \textit{size},\ \textit{1cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{square}\ \rangle\}\\
P_{d_2} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{white}\ \rangle, \langle\ \textit{size},\ \textit{1cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{square}\ \rangle\}\\
P_{d_3} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{grey}\ \rangle, \langle\ \textit{size},\ \textit{2cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{rectangular}\ \rangle\}\\
P_{d_4} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{black}\ \rangle, \langle\ \textit{size},\ \textit{1cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{square}\ \rangle\}\\
P_{d_5} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{white}\ \rangle, \langle\ \textit{size},\ \textit{1cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{square}\ \rangle\}\\
P_{d_6} &= \{\langle\ \textit{type},\ \textit{block}\ \rangle,\ \langle\ \textit{color},\ \textit{black}\ \rangle, \langle\ \textit{size},\ \textit{1cm}\ \rangle,\ \langle\ \textit{shape},\ \textit{square}\ \rangle\}
\end{aligned}$$

$$\begin{aligned}
R_{d_1} &= \{\langle\ \textit{spatial},\ \textit{left-of}\ \ (d_2)\ \rangle\ \}\\
R_{d_2} &= \{\langle\ \textit{spatial},\ \textit{left-of}\ \ (d_3)\ \rangle,\ \langle\ \textit{spatial},\ \textit{right-of}\,(d_1)\ \rangle\ \}\\
R_{d_3} &= \{\langle\ \textit{spatial},\ \textit{left-of}\ \ (d_4)\ \rangle,\ \langle\ \textit{spatial},\ \textit{right-of}\,(d_2)\ \rangle\ \}\\
R_{d_4} &= \{\langle\ \textit{spatial},\ \textit{left-of}\ \ (d_5)\ \rangle,\ \langle\ \textit{spatial},\ \textit{right-of}\,(d_3)\ \rangle\ \}\\
R_{d_5} &= \{\langle\ \textit{spatial},\ \textit{left-of}\ \ (d_6)\ \rangle,\ \langle\ \textit{spatial},\ \textit{right-of}\,(d_4)\ \rangle\ \}\\
R_{d_6} &= \{\langle\ \textit{spatial},\ \textit{right-of}\,(d_5)\ \rangle\ \}
\end{aligned}$$

Figure 3.19: KB of Example Domain IV in Figure 3.18. $PR_d = P_d \cup R_d$ for every object d in the domain.

In the initial situation there is no focus space, no linguistically salient object and one inherently salient object, d_3. According to the salience function, the salience weight of d_3 is 0.1 and the salience weights of the other objects are all 0. In the first utterance U_1, the target is d_5. Since no other object has a salience weight lower than that of d_5, d_5 has to be distinguished from all other objects in Example Domain IV. Since there are objects in C similar to d_5, a relatum is needed to identify d_5. In this case there are two possible distinguishing referring expressions to choose from (1) 'the white block to the right of the black block'; or (2) 'the white block to the left of the black block'. Both d_4 and d_6 have the same salience weight and there is no reason to prefer one over the other.[8] The Incremental Algorithm generates the one it first encounters, let's say (2). Accordingly, after the sentence U_1 has been produced, an update of the salience weights results in an increase of the weights of the objects d_5, d_4 and d_6 as presented below. Object d_5 is the last mentioned target object therefore it receives the highest linguistic salience weight. Additionally, d_5 is the center of the focus space and receives a focus space salience weight. The objects d_4 and d_6 are located in the focus space of d_5 and thus receive a focus space salience weight. Object d_6 also receives a linguistic salience weight, because it is mentioned as a relatum.

[8] Note that in human communication it might be more plausible to describe d_5 as 'the white block in between the two black blocks'.

$$\begin{aligned}
\mathbf{Sw}(d_1, s_1) &= \ \ 0 \\
\mathbf{Sw}(d_2, s_1) &= \ \ 0 \\
\mathbf{Sw}(d_3, s_1) &= 0.1 \ = (\mathbf{I\text{-}sw}(d_3, s_1) = 0.1) \\
\mathbf{Sw}(d_4, s_1) &= 0.2 \ = (\mathbf{F\text{-}sw}(d_4, s_1) = 0.2) \\
\mathbf{Sw}(d_5, s_1) &= 0.9 \ = (\mathbf{L\text{-}sw}(d_5, s_1) = 0.7) + (\mathbf{F\text{-}sw}(d_5, s_1) = 0.2) \\
\mathbf{Sw}(d_6, s_1) &= 0.8 \ = (\mathbf{L\text{-}sw}(d_6, s_1) = 0.6) + (\mathbf{F\text{-}sw}(d_6, s_1) = 0.2)
\end{aligned}$$

In the next sentence, U_2, the target is d_4. The context set C now contains all objects with a salience weight equal to or higher than d_4. In this case the target has a salience weight 0.2, which means that C contains not all objects in the domain but only d_4 d_5 and d_6. To uniquely describe d_4, a relatum is needed. Now the algorithm chooses the relatum d_5 on the basis of salience. Accordingly, the Incremental Algorithm generates the set *L*: { ⟨ *type*, *block* ⟩, ⟨ *color*, *black* ⟩, ⟨ *spatial*, *left-of*(d_5) ⟩, ⟨ *type*, *block* ⟩, ⟨ *block*, *white* ⟩ }. Like in the example in Section 3.5.1, the pronoun 'it' for d_5 can be generated on the basis of the rule proposed by Krahmer and Theune (2002). After U_2 is produced, d_4 gets the highest linguistic salience weight in the update of the salience weights. Object d_5 as a direct object decreases a bit compared to state s_1. The focus space is changed accordingly and consists of the objects d_3, d_4 and d_5. Object d_6 no longer has a focus space salience weight and also its linguistic salience decreases.

$$\begin{aligned}
\mathbf{Sw}(d_1, s_2) &= \ \ 0 \\
\mathbf{Sw}(d_2, s_2) &= \ \ 0 \\
\mathbf{Sw}(d_3, s_2) &= 0.3 \ = (\mathbf{F\text{-}sw}(d_3, s_2) = 0.2) + (\mathbf{I\text{-}sw}(d_3, s_2) = 0.1) \\
\mathbf{Sw}(d_4, s_2) &= 0.9 \ = (\mathbf{L\text{-}sw}(d_4, s_2) = 0.7) + (\mathbf{F\text{-}sw}(d_4, s_2) = 0.2) \\
\mathbf{Sw}(d_5, s_2) &= 0.8 \ = (\mathbf{L\text{-}sw}(d_5, s_2) = 0.6) + (\mathbf{F\text{-}sw}(d_5, s_2) = 0.2) \\
\mathbf{Sw}(d_6, s_2) &= 0.5 \ = (\mathbf{L\text{-}sw}(d_6, s_2) = 0.5)
\end{aligned}$$

In the third state of the discourse, in U_3, the target is d_2. Since object d_2 falls out of the current focus space, referring to d_2 means a focus shift. The salience weight of d_2 is 0 and all objects with a salience weight equal to or higher than d_2, are taken into account as distractors and thereby represented in the new context set. Since there are more blocks like d_2, the algorithm has to pick a relatum to distinguish d_2. In contrast to state s_1, the relatum now can be chosen on the basis of salience. Because d_3 is inherently salient and it is located in the current focus space it is preferred over d_1. Correspondingly the referring expression 'the white block to the left of the grey block' can be realized when the algorithm generates the set *L*: { ⟨ *type*, *block* ⟩, ⟨ *color*, *white* ⟩, ⟨ *spatial*, *left-of*(d_3) ⟩, ⟨ *type*, *block* ⟩, ⟨ *block*, *grey* ⟩ }. After U_3 is produced, the update of the salience weights renders all objects in the domain as somewhat salient. The new focus space consist of the objects d_1, d_2 and d_3, the salience weights of which increase. Object d_2 has the highest linguistic salience weight, while d_3 has the second highest. Because of the inherent salience of d_3, both d_2 and d_3 receive the same total salience weight. The linguistic salience weights of the objects d_4, d_5 and d_6 further decrease.

$$\begin{aligned}
&\mathbf{Sw}(d_1, s_3) = 0.2 = (\mathbf{F\text{-}sw}(d_1, s_3) = 0.2)\\
&\mathbf{Sw}(d_2, s_3) = 0.9 = (\mathbf{L\text{-}sw}(d_2, s_3) = 0.7) + (\mathbf{F\text{-}sw}(d_2, s_3) = 0.2)\\
&\mathbf{Sw}(d_3, s_3) = 0.9 = (\mathbf{L\text{-}sw}(d_3, s_3) = 0.6) + (\mathbf{F\text{-}sw}(d_3, s_3) = 0.2) + (\mathbf{I\text{-}sw}(d_3, s_3) = 0.1)\\
&\mathbf{Sw}(d_4, s_3) = 0.6 = (\mathbf{L\text{-}sw}(d_4, s_3) = 0.6)\\
&\mathbf{Sw}(d_5, s_3) = 0.5 = (\mathbf{L\text{-}sw}(d_5, s_3) = 0.5)\\
&\mathbf{Sw}(d_6, s_3) = 0.4 = (\mathbf{L\text{-}sw}(d_6, s_3) = 0.4)
\end{aligned}$$

The use of salience weights does not cause the Incremental Algorithm to be less efficient. The algorithm with salience weights is computationally at least as efficient as the original version, since the number of distractors in the context set is generally small and fewer properties have to be compared and selected. The update of the salience weights only concerns the objects in the current utterance and their focus space, together with the objects with a salience weight higher than zero: the objects that have been talked about in the last couple of sentences. The other objects in the domain that are not inherently salient all have a salience weight of 0. The three-dimensional salience function accounts for a context-sensitive algorithm in a multimodal way. By adding discourse sensitivity to the Incremental Algorithm, the function resolves one of the context issues mentioned in Section 3.3.3. With the determination of the exact composition of C at each point in the discourse, both accuracy and computational efficiency can be increased. The additional salience weights result in more accurate referring expressions for two reasons: (1) The target is distinguished only from the objects that are equally or more salient than the target itself; and (2) In the case that a relatum is needed, the most salient object is chosen.

3.6 Discussion

In this chapter, two basic GRE algorithms have been discussed: the Full Brevity Algorithm that generates minimal descriptions, using an extensive search method; and The Incremental Algorithm that generates descriptions in an incremental fashion. Because the latter efficiently produces descriptions which are more plausible in human communication, it is examined in more detail. As seen in Section 3.4, the Incremental Algorithm is far from complete and several extensions are proposed that each solve some problem or limitation. To improve the completeness of the Incremental Algorithm, a slight alteration accounts for plural NPs, the Boolean Algorithm provides for disjunctive combinations of properties, and spatial relations have been included. Furthermore, the generation of relative attributes is improved by the derivation of proportional information from absolute values of relative attributes. In Section 3.5, a three-dimensional notion of salience is proposed to integrate both the visual and the linguistic context in GRE.

In contrast to the benefits, the extensions also lead to new problems and loose ends: (1) The context-dependent relative properties might result in ambiguous descriptions that need avoidance; (2) It is unclear if negation is preferred over disjunction or the other way around, and (3) There is no preferred order for the multiple values of one single attribute. These loose ends are left for what they are. Nevertheless, as already mentioned in Section 3.3.3, the discussed extensions do not solve the two situations in which the Incremental Algorithm might not be able to generate a distinguishing expression even if one exists: (1) The Incremental Algorithm fails because of a lack of backtracking when properties overlap (van Deemter, 2002, section 3); or (2) In large domains or domains with homogeneous objects, the identification of a target object requires a very complex linguistic expression. This lack of backtracking together with some effects of the extensions discussed, might suggest that the Incremental Algorithm does not apply the right strategy for GRE. In Section 3.6.1 another approach for GRE is proposed which can produce the same variation of referring expressions. In Section 3.6.2 multimodality is considered to generate distinguishing descriptions in infinite domains or domains with highly similar objects.

3.6.1 Strategy and Coverage

Some of the extensions discussed in Section 3.4 give rise to algorithmic problems. For instance, the incremental strategy does not seem very suitable for the generation of spatial relations nor for the inclusion of disjunctive combinations of properties. For both extensions, an extra recursive step to the algorithm results in an increase of runtime and consequently loss of efficiency. Besides, it is difficult to combine the different extensions into a unified variant of the Incremental Algorithm; especially a variant that keeps the appealing properties of the original algorithm: efficiency and plausibility. Recently Krahmer et al. (2003) have introduced a graph-based approach, with which the basic algorithms can be simulated and improved, see also Krahmer et al. (2001). This graph-based generation algorithm models a domain as a labeled directed graph, in which objects are represented as vertices (or nodes) and the properties and relations of these objects are represented as edges (or arcs). Cost functions are used to assign weights to edges. The problem of finding a referring expression for an object is treated as finding the cheapest subgraph which uniquely characterizes the intended referent. The graph-based algorithm is presented as a meta-algorithm, in the sense that by defining the cost function in different ways it can mimic the basic GRE algorithms described above. Moreover, spatial relations can be included in the generated descriptions in a natural way. Additionally, the algorithm is able to integrate the solutions on completeness and context-sensitivity put forward so far by (van Deemter and Krahmer, to appear). In Chapter 4 the graph-based strategy is described in more detail.

3.6.2 Unimodal versus Multimodal

The completeness of the Incremental Algorithm is very much dependent on the domain of conversation. It easily fails when there exist homogeneous objects in the domain of conversation, or worse, when the domain of conversation is infinite. This can be illustrated with Figure 3.20, when singling out one particular object in a domain where all objects have most properties in common. In human communication, a distinguishing description by means of specifying the exact location of the object, ('the fourth block from the left in the third row'), or in terms of coordinates ('the block on position (4,3)') is respectively very inefficient or awkward. Moreover, such descriptions contradict the principle of Minimal Cooperative Effort (Clark and Wilkes-Gibbs, 1986), which states that both the speaker's effort in producing the description and the hearer's effort in interpreting it should be minimal. In cases such as that depicted in Figure 3.20, where a purely linguistic description may simply be too complex, including a deictic pointing gesture may be the most efficient way to single out the intended referent. In human communication, referring expressions which include pointing gestures are quite common (Beun and Cremers, 1998). Since the aim is to generate descriptions similar to those in human communication, it seems expedient to include pointing gestures. The most efficient way to single out a particular object in Figure 3.20 is to include a deictic pointing gesture in the description ('this block' together with a pointing gesture).

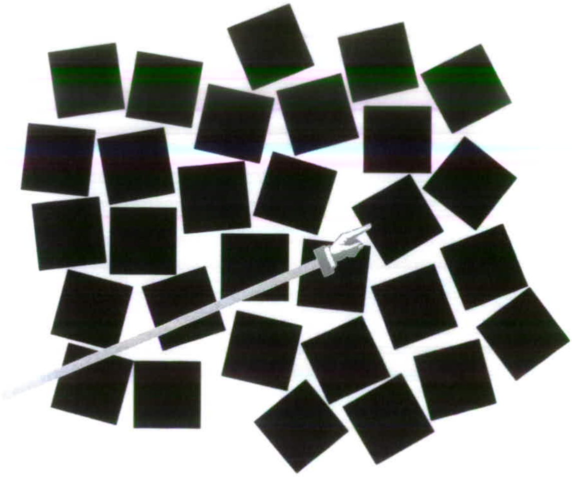


Figure 3.20: Disadvantage of the Incremental Algorithm.

In Chapter 4 a multimodal GRE algorithm is proposed, which originated from developments in GRE as discussed in this chapter. The algorithm is an extension of the graph-based algorithm (Krahmer et al., 2003), which generates natural language referring expressions combined with pointing gestures. The multimodal referring expressions are generated in a context-sensitive way, that employs the notion of salience as defined in Section 3.5. The proposed algorithm is evaluated with two studies that investigate multimodal referring expressions in human communication. These studies are reported in Chapter 5. Subsequently, in Chapter 6, the algorithm is adapted to the findings that result from these studies. Finally, a critical discussion of the algorithm is given in Chapter 7.

Chapter 4

Generating Multimodal Referring Expressions

4.1 Introduction

This chapter presents an algorithm that generates **multimodal referring expressions**: natural language referring expressions combined with pointing gestures. As argued in Section 3.6.2 there are at least two reasons to include a pointing gesture in a referential expression. First, in various situations a purely linguistic description may simply be too complex, e.g., because the domain contains many homogeneous objects. In those cases, including a deictic pointing gesture may be the most efficient way to single out the intended referent. Second, in human communication, referring expressions which include pointing gestures are quite common (Beun and Cremers, 1998).

This chapter is organized as follows. In Section 4.2 a model for pointing is presented and the graph-based algorithm in which it is implemented is introduced. Section 4.3 discusses the graph-based approach for the generation of multimodal referring expressions in more detail. In Section 4.4 a multimodal algorithm is presented, which is illustrated with worked examples. The multimodal algorithm is refined in Section 4.5, with the integration of the three-dimensional notion of salience discussed in Section 3.5. The discussion in Section 4.6 ends this chapter.

4.2 Overview

As discussed earlier in Section 2.5.2, existing algorithms for multimodal GRE are limited in that: (1) Usually only precise pointing gestures are generated; and (2) A clear criterion on when to generate pointing gestures is often missing.

In this chapter a new model for pointing is presented, which is not restricted to **precise pointing gestures**, i.e., gestures that exclusively indicate a single target. Instead, various kinds of pointing gestures are modeled. To generate multimodal referring expressions a variant of the graph-based algorithm by Krahmer et al. (2003) is proposed. The algorithm fuses both the visual and the linguistic parts of the referring expression in a compositional way. The chapter builds on ideas presented in van der Sluis (2001) and Krahmer and van der Sluis (2003).

4.2.1 The Flashlight Model

In most algorithms discussed in Section 2.5.2, pointing is only used if the object is close or when a purely linguistic description is too complex, where both closeness and complexity are measured with respect to a predefined threshold. The approach presented here is less restricted; it is not assumed that pointing is always precise and unambiguous. Instead, it allows for various gradations of precision in pointing, ranging from unambiguous to vague pointing gestures. A **precise pointing gesture** has a high precision for both speaker and hearer. Its scope is restricted to the target object, and this directly rules out the distractors. But, arguably, precise pointing is expensive in cases where the distance from the speaker to the target object is relatively large. The speaker has to overcome the distance to direct the pointing gesture to the target object in such a way that the hearer is able to unambiguously interpret the referring expression. On the other hand an **imprecise pointing gesture** generally includes some distractors in its scope because of a larger distance between the speaker and the target. Thus, such a pointing gesture has a lower precision for the hearer, although it is probably very precise from the speaker's point of view. From the speakers point of view an imprecise pointing gesture is intuitively less expensive (i.e., it takes less effort than a precise pointing gesture if the target is located at a distance). This intuition is in line with the alleged existence of neurological differences between precise and imprecise pointing. The former is argued to be monitored by a slow and conscious feedback control system, while the latter is governed by a faster and less conscious control system located in the center and lower-back parts of the brain (e.g., Smyth and Wing, 1984; Bizzi and Mussa-Ivaldi, 1990).

The model for pointing proposed here may be likened to a flashlight[1] as illustrated in Figure 1.2. When one holds a flashlight just above a surface, it covers only a small area (the target object). Moving the flashlight away enlarges the cone of light, shining on the target object but probably also on one or more distractors. Here, for the sake of simplicity, it is assumed that an object falls inside the scope of a pointing gesture if the 'cone' shines on part of it. A more fine-grained approach might distinguish between objects in the center (where the light shines brightly) and objects in the periphery (where the light is more blurred). A direct

[1] This analogy is suggested by Mariët Theune

consequence of this **Flashlight Model** for pointing is that it predicts that the amount of linguistic properties required to generate a distinguishing multimodal referring expression co-varies with the kind of pointing gesture used. In general, imprecise pointing requires more additional linguistic properties to single out the intended referent than precise pointing. In the model, the decision to point is based on a trade-off between the costs of pointing and the costs of a linguistic description. The latter are determined by summing over the costs of the individual linguistic properties used in the description. Arguably, the cost of precise pointing is determined by two factors: the size of the target object (large objects are easier to point to than small objects) and the distance between the target object and the pointing device (objects that are near are easier to point to than objects that are further away). In Section 4.3.5, Fitts' law (a fundamental empirical law about the human motor system attributable to Fitts (1954)) is used to model the costs of pointing. In addition, it is argued that Fitts' law allows the model to capture the intuition that imprecise pointing is cheaper than precise pointing.

4.2.2 A Graph-based GRE Algorithm

The algorithm described in this chapter is a multimodal variant of the graph-based GRE algorithm described by Krahmer et al. (2003). The algorithm uses a **domain graph** to represent the domain of conversation as a labeled directed graph. The objects in a domain graph are defined as the vertices (or nodes) in the graph. The properties and relations of these objects are represented as edges (or arcs). An edge that represents a property of a particular object, can be depicted as an arrow that starts and ends in the vertex which represents the object. An edge that expresses a relation of a particular object can be depicted as an arrow that originates in the vertex which represents the object and ends in the vertex which represents the relatum of the object. The properties and relations are represented as the labels of the edges. To generate a referring expression for a target object, the graph-based algorithm searches for a subgraph of the domain graph that represents the target. Which solution is returned depends on the cost function used. A **cost function** can be used to assign weights to the edges that represent the properties and relations, thereby determining their order of preference. Different definitions of the cost function mimic different search strategies. For instance, a minimal description can be generated by a search for the **smallest subgraph**: a graph with a minimal number of edges that distinguishes the target.

The graph-based approach has several advantages for GRE. For instance, relations are included naturally. Relations are represented as edges in the domain graph and can be selected in the same way as properties. This does not require any alteration of the algorithm. In general, a graph-based approach for GRE has the advantage that there are many search algorithms already in existence that deal with graph structures which enhance their use (Liebers, 2001; Messmer and

Bunke, 1995, 1998; Eppstein, 1999). With the various existing search strategies in combination with proper cost functions, the Full Brevity Algorithm (Section 3.3.1) and the Incremental Algorithm (Section 3.3.2) can easily be modeled. The Full Brevity Algorithm is mimicked by a search for the smallest distinguishing subgraph. For an imitation of the Incremental Algorithm, the graph-based algorithm uses a cost function that corresponds with the notion of preferred attributes and, accordingly, selects the cheapest edges first in the search for a distinguishing subgraph. Another advantage is that the incorporation of cost functions makes it possible to combine traditional rule-based approaches to generation with more recent, statistical approaches (e.g., Langkilde and Knight, 1998; Shaw and Hatzivassiloglou, 1999; Malouf, 2000; Ratnaparkhi, 2002). For instance, the costs of the various edges can be defined on the basis of frequency, i.e., their occurrence in a particular corpus.

As will be shown in this section, the graph-based approach lends itself well for GRE. In the next section a new extension is presented to the graph-based algorithm. For the generation of multimodal referring expressions, the domain graph is enriched with edges representing various kinds of pointing gestures. Since the algorithm looks for the cheapest subgraph, pointing edges are only selected when linguistic edges are relatively expensive or when pointing is relatively cheap.

4.3 Generating Multimodal Referring Expressions Using Graphs

4.3.1 Domain Graphs

Consider Example Domain I depicted in Figure 4.1, consisting of a set of objects with various properties and relations.[2] In this particular domain $D = \{d_1, \ldots, d_8\}$ is the set of objects, $Prop = \{$ *black*, *white*, *block*, *small*, *large* $\}$ is the set of properties of these objects and $Rel = \{$ *left-of*, *right-of* $\}$ the set of relations. A domain can be represented as a **labeled directed graph**. In general, let $Labels = Prop \cup Rel$ be the set of labels with $Prop$ and Rel disjoint, then $G = \langle\, V_G, E_G \,\rangle$ is a labeled directed graph, where $V_G \subseteq D$ is the set of vertices and $E_G \subseteq V_G \times Labels \times V_G$ is the set of labeled directed edges.[3] Two other notions that are used, are graph union and graph extension. The **union** of graphs $H = \langle\, V_H, E_H \,\rangle$ and $G = \langle\, V_G, E_G \,\rangle$ is the graph $H \cup G = \langle\, V_H \cup V_G, E_H \cup E_G \,\rangle$. If $G = \langle\, V, E \,\rangle$ is a graph and $e = (v, l, w)$ is an edge between vertices v and w and with label $l \in Labels$, then the **extension** of G with e (notated $G + e$) is the graph $\langle\, V \cup \{v, w\}, E \cup e \,\rangle$. In line with these definitions, Example Domain I can

[2] For the sake of simplicity the examples used to illustrate this algorithm are restricted to a 2D domain with only a limited number of objects. This is not an inherent limitation of the algorithm.

[3] As before subscripts are omitted where this can be done without creating confusion.

be represented as the graph presented in Figure 4.2. Only spatial relations (see Section 3.4.4) are modeled under the assumption that a distinguishing description does not use a distant object as a relatum when a closer one can be selected. Notice that properties are represented as loops, while relations are modeled as edges between different vertices.

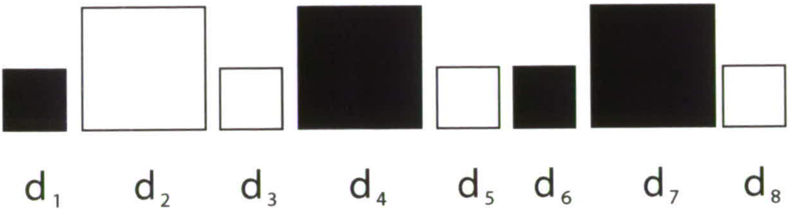


Figure 4.1: Example Domain I.

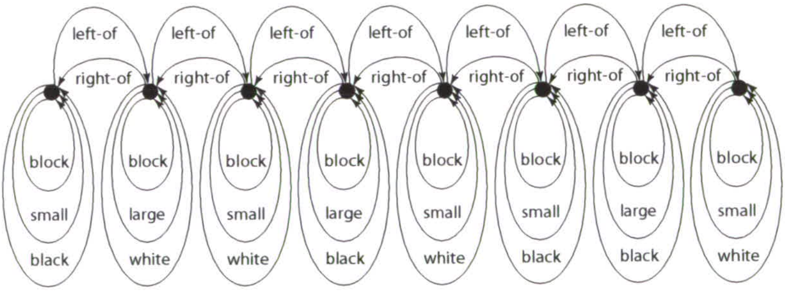


Figure 4.2: Example Domain I as a Graph.

4.3.2 Referring Graphs

Suppose in Example Domain I a distinguishing description referring to d_4 has to be generated. Then it has to be determined which properties and/or relations are required to single out d_4 from its distractors. This is done by creating **referring graphs**, which at least include a vertex representing the target object. Informally, a vertex v (the target object) in a referring graph H refers to a given object in the domain graph G iff the graph H can be 'placed over' the domain graph G in such a way that v can be 'placed over' the vertex of the given object in G and each edge from H with label l can be 'placed over' a corresponding edge in G with the same label. Furthermore, a vertex-graph pair is distinguishing iff it refers to exactly one vertex in the domain graph. The informal notion of one graph being 'placed over' another corresponds with a well-known mathematical construction on graphs, namely **subgraph isomorphism**. $H = \langle V_H, E_H \rangle$ can be **placed over** $G = \langle V_G, E_G \rangle$ iff there exists a subgraph G' of G such that H is isomorphic to G'. H is isomorphic to G' iff there exists a bijection $\pi : V_H \rightarrow V_{G'}$, such that for all vertices $v, w \in V_H$ and all $l \in Labels$:

$$(v, l, w) \in E_H \Leftrightarrow (\pi.v, l, \pi.w) \in E_{G'}$$

Given a graph H and a vertex v in H, and a graph G and a vertex w in G, it can be defined that the pair (v, H) refers to the pair (w, G) iff (1) H is a **connected graph**, i.e., each vertex has at least one edge that links it to another vertex; and (2) H is mapped to a subgraph of G by an isomorphism π and $\pi.v = w$. A vertex-graph (v, H) uniquely refers to (w, G) (i.e., (v, H) is distinguishing) iff (v, H) refers to (w, G) and there is no vertex w' in G different from w such that (v, H) refers to (w', G).

Consider Figure 4.3 containing a number of potential referring graphs for d_4, where the vertex denoting d_4 is circled. The first one, H_1 has all the properties of d_4 and hence can refer to d_4. It is not distinguishing, however: it fails to rule out d_7 (the other large black block). Graph H_2 is distinguishing. Here, the referring graph can only be placed over the intended referent d_4 in the domain graph. A straightforward linguistic realization can be 'the large black block to the left of the small white one and to the right of the small white one'.[4] Generally there is more than one distinguishing graph referring to an object. In fact, H_2 is not a minimal distinguishing graph referring to d_4. This is H_3, which might be realized as 'the large black block to the right of the white one'. Like the example in Section 3.5.4, this is a distinguishing description but not a particularly natural one; it is complex and arguably difficult for the hearer to interpret. In such cases, having the possibility of simply pointing to the intended referent would be very useful. Nevertheless, with the graph-based strategy the problem with the generation of relations as observed with the Incremental Algorithm in Section 3.4.4 is solved.

[4] A somewhat more involved realization module might realize this graph as 'the large black block between the two small white blocks'.

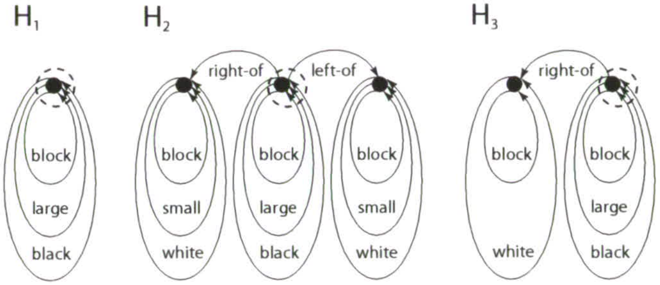


Figure 4.3: Three potential referring graphs for d_4 in Example Domain I.

4.3.3 Gesture Graphs

Suppose a pointing gesture is directed at d_4. Clearly this can be done from various distances and under various angles. The various hands in Figure 4.4 illustrate three levels of deictic pointing gestures, all under the same angle but each with different distances to the target object: **precise pointing** (*P*), **imprecise pointing** (*IP*) and **very imprecise pointing** (*VIP*). Here the presentation is limited to these three levels of precision and a fixed angle, although nothing hinges on this. Naturally, the respective positions of the speaker and the target object co-determine the angle under which the pointing gesture occurs; this in turn fixes the scope of the pointing gesture and thus which objects are ruled out by it (namely, those objects fully outside of the scope). If these respective positions are known, then computing the scope of a pointing gesture is straightforward; but the actual mathematics falls outside the scope of this thesis (c.f., Kranstedt et al., 2005; Kranstedt et al., to appear). For the sake of illustration, only three positions are represented from which a pointing gesture can be directed towards the target.

Just as properties and relations of objects can be expressed in a graph, so can various pointing gestures to these objects. All objects in the scope of a potential pointing gesture (with a certain degree of precision) are associated with an edge labeled with an indexed pointing gesture. Selecting this edge implies that all objects that fall outside the scope of the gesture are ruled out. This information is represented using a **gesture graph**. Let $Gest_v = \{P_v, IP_v, VIP_v\}$ be the set of deictic pointing gestures to a target object v. Then, given a domain graph $G = \langle\ V_G,\ E_G\ \rangle$, a gesture graph $F_v = \langle\ V_G,\ E_F\ \rangle$ is a labeled directed graph, where V_G is the set of vertices from the domain graph and $E_F = V_G \times Gest_v \times V_G$ the set of pointing edges. The subscript v in the gesture graph F_v indicates the target of the pointing gesture. Figure 4.5 displays a graph modeling the various pointing gestures in Figure 4.4. Notice that there is one gesture edge which is only associated with d_4, the one representing precise pointing to the target object (modeled by edge P_{d_4}). No other pointing gesture eliminates all distractors.

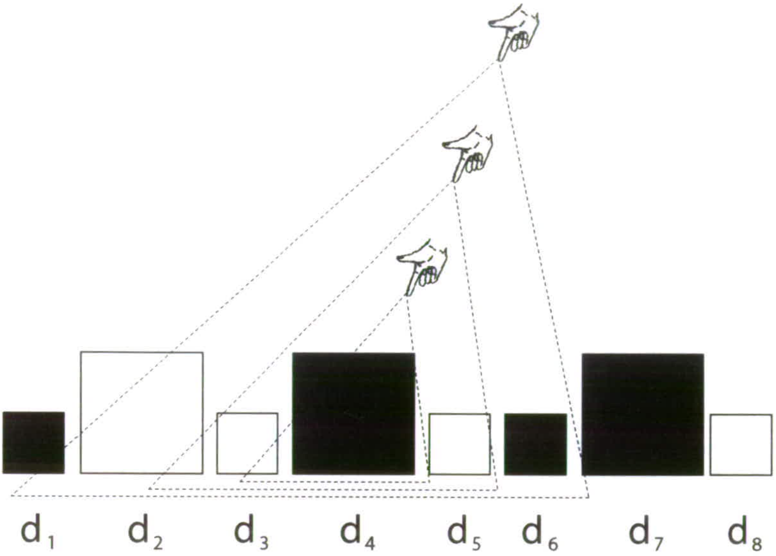


Figure 4.4: Pointing into Example Domain I.

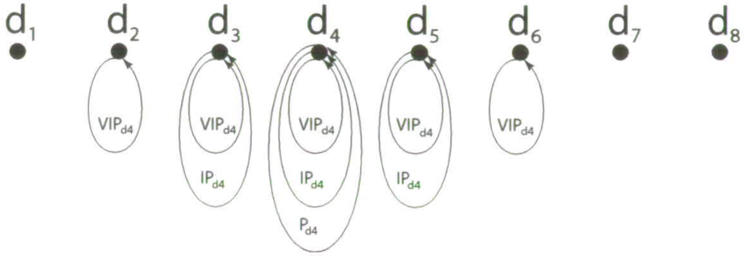


Figure 4.5: Deictic gesture graph.

4.3.4 Multimodal Graphs

Now the generation of multimodal referring graphs is based on the union of the domain graph G (which is relatively fixed) with the deictic gesture graph F (which varies with the target). To generate a multimodal referring expression for a target object v, the graph-based algorithm first has to construct the gesture graph F_v, in order to produce the multimodal graph $M = F_v \cup G$ in a particular domain D. Correspondingly, let the $Labels = Prop \cup Rel \cup Gest_v$ with $Prop$, Rel and $Gest_v$ disjoint. So $M = \langle V_M, E_M \rangle$ is a labeled directed graph where $V_M \subseteq D$ is the set of vertices and $E_M \subseteq V_M \times Labels \times V_M$ is the set of labeled directed edges. Thus, M represents the search space of the multimodal GRE algorithm. As noted before, the search for a subgraph that uniquely describes the target v depends on the cost function used. A cost function assigns weights to the labeled edges in the graph. In the case of a multimodal graph both the costs of linguistic edges and the costs of gesture edges have to be determined. In the next section the cost functions for both kinds of edges are discussed.

4.3.5 Cost Functions

In the graph perspective there are many ways to generate a distinguishing referring expression for an object. Cost functions are used to give preference to some solutions over others. Costs are associated with subgraphs H of the domain graph G. The cost function is required to be **monotonic**. This implies that extending a graph H with an edge e can never result in a graph which is cheaper than H. Formally:

$$\forall H \subseteq G \; \forall e \in E_G : \mathbf{Cost}(H) \leq \mathbf{Cost}(H + e)$$

It is assumed that if H is a subgraph of G, the cost of H (notated $\mathbf{Cost}(H)$ can be determined by summing over the costs associated with the edges of H. Thus:

$$\mathbf{Cost}(H) = \Sigma_{v \in V_H} \mathbf{Cost}(v) + \Sigma_{e \in E_H} \mathbf{Cost}(e)$$

The Costs of Properties and Relations

The cost of a subgraph is dependent on the costs of the edges in the graph. In principle within a graph the costs of all labeled edges can be determined separately. Accordingly, there are numerous ways to define these cost functions each corresponding with a particular search strategy. For instance a cost function might determine each edge to cost 1 point. In this case, when searching for the cheapest subgraph the algorithm generates the smallest distinguishing subgraph, which leads to the generation of minimal descriptions. Another approach is to define a cost function that models the notion of preferred attributes by Dale and Reiter (1995) (see Krahmer et al., 2003). In object descriptions people generally tend to include *type* properties. If that does not suffice, first absolute properties like *color* may be used, followed by relative ones such as *size*. A more fine-grained cost function might even differentiate between costs within one property. For instance, *yellow* can be cheaper than *ochre*, if *yellow* is considered more common than *ochre* (c.f., the basic level values as proposed by Dale and Reiter (1995) and Krahmer and Theune (2002)). In terms of costs, the *type* property can be for free, whereas other properties are more expensive. Absolute properties are cheaper than relative ones. There is little empirical work on the cost of relations, but it seems safe to assume that for the block domain relations are more expensive than properties. Relations are comparable to relative properties (they can not be verified on the basis of the intended referent alone). In addition, using a relation implies that a second object, the relatum, needs to be described as well and describing two objects generally requires more effort than describing a single object (see also, Section 3.4.4).

The Cost of Pointing

Arguably, at least two factors co-determine the cost of pointing: (1) The *size* of the target object (the larger the object, the easier, and hence cheaper the pointing gesture); and (2) The *distance* which the pointing device (in this case the hand) has to travel in the direction of the target object (a short distance is cheaper than a long distance). Interestingly, the pioneering work of Fitts (1954) captures these two factors in the **Index of Difficulty** (*ID*), which states that the difficulty to reach a target is a function of the size, or the width W, of the target and the distance to the target, or amplitude A:

$$ID = \log_2\left(\tfrac{2A}{W}\right)$$

Fitts' Law quoted from Fitts (1954).

Thus with each doubling of *Distance* and with each halving of *Size* the Index of Difficulty increases with 1 bit. The addition of the factor 2 in the numerator is unmotivated; Fitts added it to make sure that in his experimental conditions the *ID* is always positive. Fitts describes three experiments (a tapping, a disk transfer and a pin transfer task) and in all three a high correlation is found between the time subjects required to perform the task and the Index of Difficulty. In recent years various alternatives for the original *ID* have been proposed. The alternative proposed by MacKenzie (1991) removes the unmotivated 2 from the numerator and starts counting from 1 assuring that the *ID* is always positive.

$$ID = \log_2(\tfrac{A}{W} + 1)$$

Fitts' Law as modified by MacKenzie (1991).

MacKenzie shows that this version of the *ID* even fits the experimental data slightly better. Below the cost of pointing is derived from the modified *ID*. As argued, it seems a reasonable assumption that imprecise pointing is cheaper than precise pointing; it rules out fewer distractors, but also requires less motoric precision and effort from the speaker. The Index of Difficulty allows this intuition to be captured in the following way. *Distance* is not interpreted as the distance from the neutral, current position of the hand to the target object, but rather as the distance from the current position of the hand to the target position of the hand. For the imprecise variants of pointing this distance is smaller and hence the Index of Difficulty is lower. Thus, the smaller the distance from the current position of the hand to the target position for pointing, the lower the cost. In sum: if $g \in Gest_v$ is a pointing gesture, A is the distance from the hand's current position to its target position, and W is the size of target object, then the cost associated with that pointing gesture is defined as follows:

$$\mathbf{Cost}(g) = \log_2(\tfrac{A}{W} + 1)$$

4.4 A Graph-based Multimodal Algorithm

4.4.1 Sketch of the Algorithm

The multimodal variant of the graph-based algorithm described in this section generates the cheapest distinguishing graph for a target object, if one exists. Whether this cheapest graph includes pointing edges, and if so, to what level of precision, is determined by a trade-off between the respective cost of pointing and the costs of the linguistic edges. In Figure 4.6 the pseudocode of the algorithm's main function *GenerateReferringExpression* and the subgraph construction function *FindGraph* are presented. In line (1) *GenerateReferringExpression* is called with the parameters v (the target) and G (the domain graph). In line (2) a deictic gesture graph F_v is constructed for the target. Successively, in line (3) the gesture graph is merged with the domain graph which results in a multimodal graph M. The variable in which the best referring graph found so far is stored, *BestGraph*, is initialized as undefined ($\perp$) in line (4). H, the graph under construction, is initialized as the graph only consisting of the vertex v in line (5). In line (6) the value of *BestGraph* is assigned the result of the function *FindGraph*. The function *FindGraph* is called with the parameters the target (v), the best graph so far (*Bestgraph* $= \perp$), the graph under construction (H) and the multimodal graph (M).

The function *FindGraph*, in line (8), contains two conditions on which it returns the best graph, and a recursive step. In the recursion the graph under construction, H, is extended with edges with which the target v can be described. The first condition in line (9) is a check whether *BestGraph* is not $\perp$ (i.e., a solution has been found) and whether *BestGraph* is cheaper than H (i.e., the solution found earlier is cheaper than the graph under construction). If the latter is the case then H is discarded (since due to monotonicity it can never end up cheaper than the best solution found so far). The second condition, line (11), is a check whether the graph H refers uniquely to vertex v, in which case H is returned in line (12). These two conditions are checked for every relevant subgraph H of M

```
(1)   GenerateReferringExpression($v, G$)
(2)     construct(v, $F_v$, G)
(3)     $M := F_v \cup G$
(4)     $BestGraph := \bot$
(5)     $H := \langle \{v\}, \emptyset \rangle$
(6)     $BestGraph :=$ FindGraph($v, BestGraph, H, M$)
(7)     return $BestGraph$

(8)   FindGraph($v, BestGraph, H, M$)
(9)     if $BestGraph \neq \bot$ and Cost($BestGraph$) $\leq$ Cost($H$) then
          return $BestGraph$
        end if
(10)    $C := \{n \mid n \in V_M \wedge$ MatchGraphs($v, H, n, M$)$\}$
(11)    if $C = \{v\}$ then
(12)      return $H$
        end if
(13)    for each adjacent edge $e$ do
(14)      $I :=$ FindGraph ($v, BestGraph, H + e, M$)
(15)        if $BestGraph = \bot$ or Cost($I$) $\leq$ Cost($BestGraph$) then
(16)          $BestGraph := I$
            end if
        end foreach
(17)    return $BestGraph$
```

Figure 4.6: Pseudocode of the algorithm's main function *GenerateReferringExpression* and the subgraph construction function *FindGraph*.

constructed in the loop started in line (13). In this loop the algorithm recursively tries to extend H by adding **adjacent edges** e, that is edges which start in v or possibly in any of the other vertices added later on to H, the graph under construction. For each graph H the algorithm checks if the vertex-graph pair (v, H) is distinguishing or whether it also refers to other vertices than v in M. In line (10) the context set C is defined as the set that contains all vertices n, in the graph, V_M that can be referred to by H. The function *MatchGraph*(v, H, n, M) checks for subgraph isomorphisms with the current graph H in M. As soon as v is the only member of the context set C, line (11), a graph referring uniquely to v is found, line (12). In line (14) I is assigned the resulting graph of the recursive call to *FindGraph*. If the graph I is cheaper than the *BestGraph*, I is assigned to *BestGraph* in line (16). *FindGraph* repeats these steps until all relevant subgraphs have been tried. In line (17), the algorithm returns the cheapest distinguishing graph which refers to the target object if there is one, otherwise the undefined null graph ($\bot$) is returned. Note that the latter possibility never arises due to the presence of unambiguous pointing gestures, expensive though they may be. Which referring graph is the first to be found depends on the order in which the edges are tried. Clearly this is a place where heuristics are helpful, i.e., it is generally beneficial to try cheap edges before expensive ones. If this heuristic is applied, as soon as a distinguishing graph is found all graphs that are more expensive can be discarded. It only has to be checked if there are cheaper distinguishing graphs.

4.4.2 Worked Examples

This section demonstrates the multimodal graph algorithm with a series of three examples. These show how the algorithm generates referring expressions with or without the various pointing gestures depending on the distance from the hand to the position required for pointing to a target object. Before the workings of the algorithm are illustrated, a cost function has to be specified for the labeled edges in the domain graph. As suggested in Section 4.3.5 the properties in the current domain may be ordered as follows: $\langle$ *type*, *color*, *size*, *spatial relations* $\rangle$. In terms of costs, let us assume the following arbitrarily chosen costs: *type* edges are for free, *color* edges cost 1, *size* edges cost 2 and relational edges 2.50. For the three examples, suppose a description for object d_4 in Example Domain I in Figure 4.4 has to be generated. For the sake of illustration, let us assume that d_4 is a block with sides of 1 cm. Furthermore, for reasons of simplicity, 3 positions for the hand are adopted to direct a pointing gesture to the target object. Distances 0 cm, 20 cm and 40 cm are used to indicate respectively precise pointing (*P*), imprecise pointing (*IP*) and very imprecise pointing (*VIP*). The pointing gestures and their scopes are illustrated in Figure 4.4. For the examples presented in this section size and distance are arbitrarily chosen.

Example 1:
Suppose the distance from the current neutral position of the hand to the closest position required for precise pointing (*P*) is 130 cm. Acknowledging the defined positions to point from, the distance to bridge for imprecise pointing (*IP*) is 110 cm, whereas for very imprecise pointing (*VIP*) the hand has to move 90 cm. Some easy calculations show that the Index of Difficulty in these three cases is 7.03 bits, 6.79 bits and 6.51 bits respectively. Thus, precise pointing (P_{d_4}) costs 7.03, imprecise pointing (IP_{d_4}) 6.79 and very imprecise pointing (VIP_{d_4}) 6.51. Hence all necessary costs are defined and the function *GenerateReferringExpression* can be called with the parameters d_4 (the target) and G (the domain graph as presented in Figure 4.2). First of all the deictic gesture graph F_{d_4} is constructed as shown in Figure 4.5, and merged with G. This results in a multimodal graph M. The variable *BestGraph*, for the cheapest solution found so far, is initialized as the undefined graph $\perp$ (no solution is found yet), and the referring graph under construction H is initialized as the graph only consisting of the vertex d_4. The

function *FindGraph* is called with the parameters the target vertex (d_4), the best graph so far ($\perp$), the graph under construction (H) and the multimodal graph (M). The order in which the labeled edges are tried determines which referring graph is found first. As already suggested in Section 4.4, cheap edges are tried first. Accordingly, the first distinguishing graph found is depicted as H_1 in Figure 4.7, which costs $(0 + 1 + 2 + 2.50 + 0 + 1) = 6.50$. At this point, graphs which are more expensive can be discarded, since they are never cheaper than the best solution found so far, due to the monotonicity constraint. In the current situation pointing gestures are relatively expensive and there is no distinguishing graph which is cheaper than H_1. The distinguishing graph that contains a precise pointing gesture, H_3 in Figure 4.7, costs $(0 + 7.03) = 7.03$, graph H_4, containing the edge IP_{d_4} costs $(0 + 1 + 6{,}79) = 7.79$ and graph H_2 with the edge VIP_{d_4} costs $(0 + 1 + 2 + 6.51) = 9.51$. So it can be inferred that in this case, the first distinguishing graph found, H_1, happens to be also the cheapest one. H_1 can be realized as: 'the large black block to the right of a white one'. Thus, when pointing is relatively expensive, the algorithm generates fully linguistic expressions.

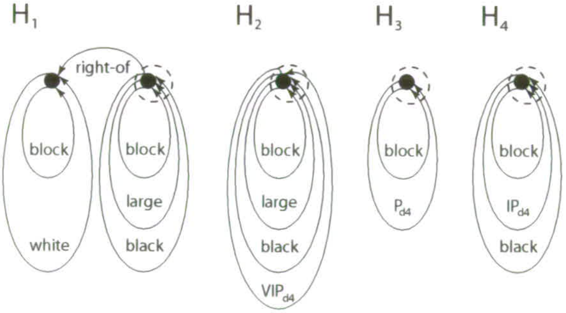


Figure 4.7: Four distinguishing multimodal referring graphs for d_4.

Example 2:
As an example of very imprecise pointing (*VIP*), suppose that the hand of the speaker is located somewhat closer to the defined position required for precise pointing P, let's say 43 cm. Accordingly, the distance to the position for an *IP* gesture is 23 cm and to reach the position for a *VIP* gesture the distance is 3 cm. Now the calculation of the Index of Difficulty in the three cases results in 5.46 bits, 4.59 bits and 2 bits respectively. Thus, precise pointing P_{d_4} costs 5.46, imprecise pointing, IP_{d_4} costs 4.59 and very imprecise pointing VIP_{d_4} costs 2. As before a call to the function *GenerateReferringExpression*(d_4, G) initiates the construction of a multimodal graph M, the graph under construction H and the undefined null graph *BestGraph*. The latter is subsequently defined with a call

to the function *FindGraph*(d_4, BestGraph, H, M). Adding cheap edges first, the first graph found is H_2 in Figure 4.7 for $(0 + 1 + 2 + 2) = 5$. Successively, the algorithm encounters the distinguishing graphs, H_1 for $(0 + 1 + 2 + 2.50 + 0 + 1) = 6.50$, graph H_3 with the edge P_{d_4} that costs $(0 + 5.46) = 5.46$ and H_4 that contains the edge IP_{d_4} which costs $(0 + 1 + 4.59)$ 5.59. All are more expensive than H_2 and are therefore discarded. Consequently, H_2 is generated and can be realized as: 'the large black block' combined with very imprecise pointing gesture (*VIP*).

Example 3:
As an example of precise pointing (P), suppose that the hand of the speaker is located even closer to the target, for instance only 3 cm away from the position required for a P gesture. Again the distances to the positions for producing the distinctive less precise pointing gestures have to be measured. The distance to target position for (*IP*) is 17 cm, whereas to reach the position for a *VIP* gesture the hand has to move 37 cm away from the target. The Index of Difficulty in the three cases is 2 bits, 4.17 bits and 5.25 bits respectively. Thus, precise pointing P_{d_4} costs 2 imprecise pointing IP_{d_4} 4.17 and very imprecise pointing VIP_{d_4} 5.25. Now a call to the multimodal algorithm initiates the same procedure as described above, only resulting in an even cheaper graph. The first graph found containing the cheapest edges is H_1 in Figure 4.7 for $(0 + 1 + 2 + 2.50 + 0 + 1) = 6.50$. This time, the distinguishing graph that contains the edge VIP_{d_4}, H_2 is not tried, since it is more expensive than 6.50, namely $(0 + 1 + 2 + 5.25) = 8.25$. The next best graph is H_4 containing the edge IP_{d_4} costs $(0 + 1 + 4.59) = 5.59$, which discards H_1. The cheapest distinguishing graph, however, costs $(0 + 2) = 2$ and is depicted as H_3 in Figure 4.7. H_3 is possibly realized as 'this block' together with a precise pointing gesture (P).

4.5 A Context-sensitive Multimodal Algorithm

At this point the graph-based multimodal GRE algorithm only produces descriptions for objects that have not been mentioned before. As seen in Section 3.5, the use of salience weights accounts for context-sensitive descriptions, while at the same time reducing the search space. The three-dimensional notion of salience presented in Section 3.5 fuses the visual and the linguistic context and therefore guarantees context-sensitive descriptions both linguistically and visually. In this section the three-dimensional notion of salience is integrated in the graph-based algorithm. The salience function as defined in Section 3.5.3 assigns salience weights to every vertex in the domain graph. On the basis of the salience weights of the vertices in the domain, a context set, C, can be constructed as a subgraph of the domain graph, which only contains the vertices that are at least as salient as the target. While discourse progresses such a restriction on the domain results in a decreasing number of distractors and fewer properties are needed to describe the target. So far, the context set C is defined as the set that contains all vertices, n, for which the function *MatchGraph*(v, H, n, M) holds (line (10) in Figure 4.6). The integration of salience results in a new version of the definition of the context set. Now, C is further restricted to the vertices n that are more than or equally salient as the target vertex v at a certain state s_i:

$$(10')\ C := \{n \mid n \in V_G \wedge \textbf{MatchGraphs}(v, H, n, M) \wedge \textbf{sw}(v, s_i) \leq \textbf{sw}(n, s_i)\}$$

Thus, the three-dimensional notion of salience as presented in Section 3.5 can be integrated in the graph-based algorithm. The model for multimodal GRE described in this chapter predicts that a distinguishing description for an object that is salient is less likely to contain a pointing gesture, unless pointing is very cheap. If an object is salient, this generally implies that its distractors are relatively few; typically, only a few objects are somehow salient. This in turn implies that fewer, or less expensive edges are required to rule out the distractors. Contrastively, in cases of a focus shift a pointing gesture is more likely to be included.

With respect to focus space salience, there are some interesting connections between focus space and multimodality to be considered. For instance, pointing gestures typically serve to demarcate the focus of attention. As shown in Figure 4.4, the scope of both an imprecise and a very imprecise pointing gesture decrease the number of objects in the context set. Therefore, in case one of these pointing gestures is included to refer to the most recent target object, the focus space can be adjusted to the objects in the scope of that pointing gesture. In case of a precise pointing gesture or no pointing gesture, the focus space just contains the directly related objects as defined in Section 3.5.2. Below a modified definition of focus space is presented, which adapts to the modalities used in the referring expressions. Basically, the focus space is extended with the objects that are in the scope of the pointing gesture used to indicate the last mentioned target object o. The scope of the pointing gesture g contains all vertices in the multimodal graph M that have an edge that is labeled g, where g is an element of the gestures of o.

$$\textbf{M-focusspace}(o) = \{o\} \cup \textbf{Scope}(g) \cup \textbf{Focusspace}(o)$$

where:
Focusspace(o) is defined as in Section 3.5.2 and
$\textbf{Scope}(g) = \{n \mid n \in V_M \wedge (n, g, n) \in E_M\}$ for $g \in Gest_o$

With this more principled definition of the focus space function, an update of the salience weights can result in various distributions of focus space salience, depending on the pointing gesture used in the previous referring expression, if any.

4.6 Discussion

This chapter has described a new model for the generation of multimodal referring expressions. The approach is based on only a few, independently motivated assumptions. A Flashlight Model for pointing was proposed, allowing for different gradations of pointing precision, ranging from precise and unambiguous to imprecise and ambiguous. The GRE algorithm used to generate the multimodal referring expressions according to this model is a graph-based algorithm which tries to find the cheapest referring expression for a particular target object (Krahmer et al., 2003). In the search for the cheapest solution, it is assumed that linguistic properties have certain costs (c.f., the preferred attributes from Dale and Reiter, 1995), whereas the costs of the various pointing gestures are derived from an empirically motivated adaptation of Fitts' law. The model has a number of nice consequences:

- There is no a priori criterion needed to decide when to include a pointing gesture in a distinguishing description. Rather the decision to point is based on a trade-off between the costs of pointing and the costs of linguistic property;
- The amount of linguistic properties required to generate a distinguishing multimodal referring expression is predicted to co-vary with the kind of pointing gesture;
- An isolated object does not require precise pointing; there is always a graph containing a less precise and hence cheaper pointing edge which has the same objects in its scope as the more precise pointing gesture;
- The algorithm never outputs a graph with multiple pointing edges to the same target, since there is always a cheaper graph which omits the less precise one;
- A precise pointing edge and a relational edge never occur together in a distinguishing graph, because a graph that contains a precise pointing gesture is distinguishing;
- The way the algorithm is defined, precludes any situations in which a pointing gesture is selected for the relatum.

To implement the Flashlight Model for pointing, the graph-based algorithm presents a very suitable framework. In contrast, an incremental strategy to the generation of multimodal descriptions does not seem to be straightforward. An incremental approach to generate multimodal descriptions is presented in the multimodal variant of the Incremental Algorithm as presented by van der Sluis and Krahmer (2001). Here pointing gestures are generated dependent on the number of linguistic properties needed to single out the target. If a purely linguistic expression is too complex, (i.e., the number of linguistic properties exceeds a certain threshold) a pointing gesture is included and the generated linguistic referring expression is simply discarded. Thus, only precise pointing gestures are generated together with a relatively simple referring expression which contains no more than a head noun ('this block' combined with a precise pointing gesture). Another way of extending the Incremental Algorithm with the generation of pointing gestures is to enrich the list of preferred attributes with the gestures *VIP*, *IP* and *P* (in that order of preference, modeling the increase in cost). In this approach, first a number of linguistic edges is selected (independent of the kind of pointing gesture) followed by one or more pointing edges. But that does not work, since the lack of backtracking entails that all selected properties are realized. And this implies that: (1) Multiple gestures might be generated (if *VIP* together with some linguistic properties is not distinguishing); and (2) If *P* is generated it comes with more properties than necessary, because *P* makes all earlier selected properties redundant. This seems to suggest that the Flashlight Model outlined in Section 4.2.1 is inherently non-incremental.

But also, the graph-based algorithm is more generally beneficial for GRE. For instance, relations are naturally incorporated, in contrast to the difficulties with the integration of relations in the Incremental Algorithm (Section 3.4.4). In a labeled directed graph both the properties and relations of the objects are represented as edges. Since no edge is added to the subgraph more than once, infinite recursion does not occur. In fact, not only relations, but all the extensions to the Incremental Algorithm as examined in Section 3.4 can be combined within one graph-based GRE algorithm as demonstrated by van Deemter and Krahmer (to appear). An unfortunate disadvantage is the algorithm's complexity. In general, finding subgraph isomorphism is an NP-complete problem (Garey and Johnson, 1979). Another factor that plays a role in the algorithm's complexity is the diversity of the labeled edges in the graph. On average, solutions are found quicker when the labeling of edges displays more variation. For some subclasses of graphs the subgraph isomorphism problem can be solved more efficiently. For instance for **planar graphs**, i.e., graphs that can be drawn in a two-dimensional environment without crossing edges, graph isomorphism is solvable in linear time (Eppstein, 1999). Moreover, by removing relational edges (i.e., edges between different nodes) until the result graph no longer contains crossing edges, all non-planar graphs can be transformed into planar ones (Liebers, 2001).

In using the graph-based algorithm in multimodal GRE, there are at least two ways to decrease the search space and thereby reduce runtime. First, with the generation of pointing gestures it is always possible to single out one object from the others due to the presence of unambiguous pointing edges. A precise pointing gesture *P* can always be generated even if it is very expensive. As an advantageous side effect of this a polynomial upper bound is obtained for the theoretical complexity of the algorithm. At least the cost of one distinguishing graph for the target object is known; the graph consisting of only a vertex for the target object and a precise pointing edge. This means that not all subgraphs of the merged multimodal graph M have to be inspected, but only those subgraphs which do not cost more than the precise pointing graph. Thus, only graphs with less than K edges (for some K depending on the cost of precise pointing) have to be inspected, which requires in the worst case $\mathcal{O}(n^K)$, with n the number of edges in the graph M. This worst case complexity is, of course, computationally rather unattractive for larger values of K. A second method to decrease runtime is the use of salience. As shown in Section 4.5, the multimodal notion of salience presented in Section 3.5, can be incorporated in the algorithm. The definition of focus space is slightly altered to the scope of the pointing gesture used.

The multimodal GRE algorithm presented in this chapter is based on a combination of the empirical observations of Beun and Cremers (1998) and Fitts (1954) and is thus designed to mimic human communication. But does the output of this algorithm resemble the expressions produced by human speakers? In an attempt to answer this question, a detailed empirical evaluation of the model (in the form of controlled production tests) is presented in the next chapter. The experiments particularly address the consequences of the model listed at the beginning of this section.

Chapter 5

Empirical Evaluation

5.1 Introduction

This chapter reports on two evaluation studies for the generation of multimodal referring expressions. These studies are a first step towards a full evaluation of the multimodal algorithm presented in Chapter 4. In Section 5.2 the question of how such an algorithm is best evaluated is discussed. In Section 5.3 and Section 5.4 two evaluative studies are discussed. For both studies a general overview is given, the method used is described, the results are presented and finally the findings are discussed. Section 5.5 discusses the consequences of these findings for the multimodal algorithm.

5.2 Evaluation Using Production Experiments

In Chapter 4 an algorithm is presented for the generation of multimodal referring expressions. How should one evaluate such an algorithm? Evaluating content determination algorithms for natural language generation systems is known to be difficult. Corpora, for instance, which are often used for the evaluation of other natural language processing applications, are not straightforwardly applicable to the evaluation of content determination algorithms, since typically the underlying semantic representations are not accessible. The descriptions extracted from corpora provide no information about the objects described, nor about their context. Adding additional modalities, like pointing gestures, only leads to further complications. In this chapter, production experiments are proposed for the evaluation of multimodal NLG algorithms. In such experiments, subjects are offered stimuli which they have to verbalize. In this way, spontaneous data is gathered, (i.e., subjects were not told what to say), while controlling the input representations at the same time (i.e., the target and its properties are known). It can then be investigated to what extent the verbalized output of the algorithm coincides with the utterances of the subjects in the dimension under investigation. A potential disadvantage of this method is that different aspects of an algorithm may require different experiments, and in addition, that performing these experiments tends to be a time-consuming process. For data-driven development and testing of multimodal interpretation and generation modules, it is important to collect data about how humans produce multimodal referring expressions combining speech and gesture (e.g., Piwek and Beun, 2001; Kranstedt et al., 2003; Kranstedt et al., to appear). As a case in point, two experiments are described in order to evaluate the algorithm presented in Chapter 4.

The first experiment, Study I presented in Section 5.3, is a simple experiment in a strict setting that addresses one of the crucial ingredients of the algorithm: the claim that the linguistic part of a multimodal referring expression co-varies with the kind of pointing gesture. It seems likely that imprecise pointing requires more linguistic material to single out the target object; but exactly what kind of material is used is not known. Moreover, it might be that the kind of target object plays a role in this. In the first experiment, these factors are controlled. The second experiment, Study II presented in Section 5.4, is a more elaborate experiment in an interactive setting, which accounts for the assumption that the use of pointing gestures for object identification depends on the size of the object. Because an imprecise pointing gesture does not seem efficient to single out a small object, locative relata are expected to be produced in the linguistic part of the referring expression instead. Imprecise pointing gestures are predicted to be used typically to demarcate large objects. Of course, precise pointing gestures can be used to indicate both large and small objects. Additionally, the kind of pointing gestures are closely looked at. Are the pointing gestures static in nature, or do speakers move their hand during a pointing gesture, for example drawing the shape of the target? Study I and Study II have been described in the papers by van der Sluis and Krahmer(2004a; 2004b), respectively.

5.3 Study 1: Precise vs. Imprecise Pointing

5.3.1 Overview

To elicit multimodal referring expressions a production experiment is conducted. Subjects had to perform an object identification task, in which they were first shown an isolated object which they subsequently had to single out among a set of comparable objects. Two sorts of target objects (geometrical figures and photos of famous mathematicians) were used to determine whether the kind of target influenced the results. Half of the subjects performed the tasks at a close distance (i.e., they were able to touch the target object directly), the other half of the subjects performed the same tasks from a certain distance, from which they could only indicate the location of the target. The experiment has a two by two design, with *target* as a within-subject variable and *distance* as between-subject variable. Table 5.1 summarizes the experimental design.

		DISTANCE	
		NEAR	FAR
TARGET	OBJECT	I	II
	PERSON	III	IV

Table 5.1: Overview of the experimental design with *distance* as between-subject and *target* as within-subjects variables.

In this section, a statistical analysis is presented of the resulting 600 multimodal referring expressions. In Section 5.3.2 the experiment conducted is described, a general overview is given, and subjects, experimental setting, materials and data processing are discussed. In Section 5.3.3 the results of the experiment are presented: the interaction of language and speech and their relation to the kind of target. Section 5.3.4 ends with a discussion.

5.3.2 Method

Subjects

Twenty native speakers of Dutch participated as subjects, all students and colleagues from Tilburg University. None was familiar with the multimodal generation algorithm being tested. For each condition, the group of subjects consisted of five men and five women.

Experimental setting

Subjects were led to believe they were testing a new computer system which could be operated by the combined usage of speech and gesture. They were told the system was in its testing phase; their input was required for calibration purposes. To evoke pointing gestures, the subjects were given a pen-like **digital stick**, a pen mouse of approximately 10 centimeters as depicted in Figure 5.1, which could be used as a pointing device. They were told that the digital stick emitted a signal which the computer could detect and interpret. In addition, subjects were equipped with a headset including a microphone through which they could speak to the computer. Their task was to identify the target objects via speech and gesture. Each target object was first displayed in isolation on a 17 inch screen, after which the target object was presented among a set of distractors from which the subject had to single it out. To avoid influencing the subjects in their realizations, no feedback was given by the experimenter or the computer. Half of the subjects performed the experiment in the **near condition**;

they were placed directly in front of the screen and could touch the target object with the stick (precise pointing). The other half of the subjects, those in the **far condition**, were placed at approximately 2.5 meters from the screen. By definition their pointing acts were always imprecise.

Figure 5.1: Digital stick or pen mouse.

Stimuli

Two kinds of target objects were used in the experiment: (1) 15 two-dimensional geometrical objects; and (2) 15 black and white photographs of persons (all famous mathematicians). The geometrical figures vary in shape (square, circle, triangle) and color (red, blue, green). The persons display a greater variety: some are male, some female, they may wear hats, glasses, moustaches and/or beards (only the men), and they may have long, short, grey or no hair. The 30 target objects were presented to subjects in a random order. For the identification task, the target object was presented on a computer screen together with a number of other objects from the same domain. To facilitate pointing, the objects were presented on the screen in two isolated groups of 2 or 3 objects, one containing the target the, **target group**, while the other group solely consisted of distractors, the **distractor group**. The position of the target group on the screen was systematically varied, as was the position of the target object within the target group. Figures 5.2 and 5.3 illustrate the stimuli for objects and persons respectively.

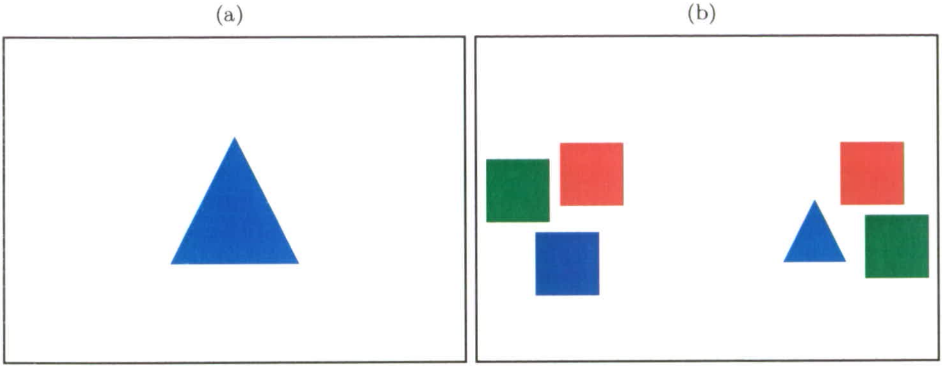


Figure 5.2: A stimulus example from the domain of geometrical objects. First, the target object is displayed in isolation (a). Subsequently it is presented together with a number of similar objects (b).

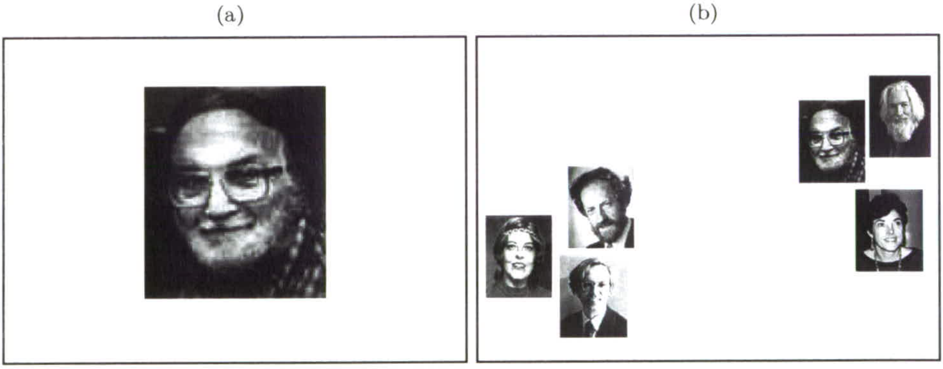


Figure 5.3: A stimulus example from the domain of photographed persons. First, the target object (a picture of a mathematician) is displayed in isolation (a). Subsequently it is presented together with a number of similar objects (b).

Data processing
The subjects were filmed during the experiment. The resulting data consist of (20 subjects × 30 stimuli) = 600 multimodal referring expressions. All utterances were transcribed. The kind of pointing gesture was classified, and the kinds of linguistic properties were determined and counted. All subjects produced a

correct, i.e., distinguishing, description for each target object. The descriptions were analyzed with respect to the following features:

- *Number of words* Per target description, the number of words used are counted.
- *Number of disfluencies* Per target description, the number of repairs, repetitions, pauses and filled pauses are counted.
- *Occurrences of type* In Dutch, *type* properties are mostly head nouns that describe the target. In a block domain these head nouns typically express the shape of an object (e.g., 'triangle', 'square', 'ball'). This feature counts the number of *type* properties used to describe the target.
- *Occurrences of properties* Per target description, the number of verbalized target properties (with the exception of *type*) are counted (e.g., 'round', 'green').
- *Number of locative expressions* Per target description, the number of locative expressions are counted (e.g., 'on the left side').

For each of the features an analysis of variance (ANOVA) with repeated measures is performed to test for significance, with distance as between-subject variable and target as within-subject variable.

5.3.3 Results

As intended, all subjects always used a pointing gesture. When near the target, this pointing gesture was always a precise one, where the target object was directly touched with the pointing device. When far from the target, subjects by definition employed imprecise pointing gestures, which basically denote in which of the two groups of objects on the screen the target object was located. This indicates that the operation of (im)precise pointing worked as planned, and the hypothesis can be tested that the kind of pointing gesture co-varies with the linguistic referring expression. No gender differences were found, so combined results for male and female subjects are presented.

As a first approximation, the number of words are considered together with the number of disfluencies in the multimodal referring expressions as a function of the distance and the target. The results are presented in Table 5.2. For both the number of words and the disfluencies there is a significant effect of distance ($F(1, 18) = 45.45, p < .01$) and ($F(1, 18) = 9.24, p < .01$), respectively, which indicates that in the far condition subjects use more words and less fluent speech than in the near condition. For the number of words there is also a significant effect of target ($F(1, 18) = 53.99, p < .01$); this implies that subjects require more words to refer to the persons than to the objects. In addition, there is an interaction between distance and target for both factors (words = $F(1,18) = 49.09, p < .01$ and disfluencies = $F(1,18) = 3.48, p < .08$). This can be explained by observing that the effect of distance is stronger for persons than for objects in the far condition but not in the near condition.

			DISTANCE	
			NEAR	FAR
	OBJECT	**words**	0.78 (1.21)	2.93 (0.87)
		disfl	.00(.00)	.16(.35)
TARGET				
	PERSON	**words**	0.84 (1.31)	5.45 (1.32)
		disfl	.01(.02)	.34(.25)

Table 5.2: Average number of *words* and *disfluencies* per description as a function of *distance* and *target*. Standard deviations are given between brackets.

So, it appears that subjects indeed adapt their linguistic material to the kind of pointing gesture they use. Although some differences among the subjects were observed, especially in the near condition, each of the subjects displayed consistent behavior throughout the experiment. In the near condition, a precise pointing gesture suffices to single out the target object. Half of the subjects in this condition only used a precise pointing gesture, three subjects typically accompanied the gesture with a demonstrative determiner, 'deze' (this one), the remaining two subjects tended to include some more words in their multimodal referring expressions. In the far condition, all subjects used imprecise pointing gestures, and hence were required to use additional linguistic material to produce unambiguous referring expressions.

Table 5.3 presents a more detailed analysis of the linguistic material, making a distinction between *type* information (whether the target is a square, a circle, person, etc., i.e., the information given in the head noun), the number of prenominal properties (*prop*) e.g., *color*, *hair style*, *hair color*, etc. and the number of locative expressions (*loc*) e.g., left, below, etc. Looking at the presence of the property *type*, a significant effect of distance is found ($F(1,18) = 144.6, p < .001$); no effect of target and no interaction either (in both cases $F(1,18) < 1$). That is, when subjects use a precise pointing gesture in this experiment they do not use type information, but when they use an imprecise pointing gesture, they do include type information (sometimes even twice, explaining the 1:01 for persons). For adjectival properties, both a significant effect of distance is found ($F(1,18) = 70.01, p < .01$), and a significant effect of target ($F(1,18) = 10.31, p < .01$). No interaction is found. In terms of the figures in Table 5.3: when subjects use a precise pointing gesture, they do not use adjectival properties, and when they use an imprecise pointing gesture they do. In addition, when subjects describe an object they are somewhat more likely to use a prenominal adjective than when describing a person. For locations, finally, a significant effect of distance is found ($F(1, 18) = 2.02, p < .05$), and a significant effect of target ($F(1, 18) = 20.47, p < .01$). There is also an interaction between target and distance ($F(1, 18) = 16.62, p < .01$). Inspection of the table reveals that these effects can be explained by the fact that location information is rare when a precise pointing gesture is used, but relatively common when describing a person in combination with an imprecise pointing gesture.

			DISTANCE	
			NEAR	FAR
TARGET	OBJECT	**type**	0.15 (0.32)	1.00 (0.00)
		prop	0.19 (0.34)	0.94 (0.13)
		loc	0.09 (0.27)	0.30 (0.43)
	PERSON	**type**	0.11 (0.17)	1.01 (0.04)
		prop	0.03 (0.11)	0.76 (0.26)
		loc	0.12 (0.33)	0.81 (0.45)

Table 5.3: Average numbers of attributes *type*, *prop*erty and *loc*ation given per description as a function of *distance* and *target*. Standard deviations are given between brackets.

5.3.4 Discussion

A straightforward production experiment was described; subjects generate distinguishing descriptions for selected target objects, and the resulting descriptions are analyzed and compared with the predictions made by the algorithm. The experimental results indicate that speakers indeed vary the linguistic part of a multimodal referring expression in relation to the distance from the target object; the amount of linguistic material co-varies with the kind of pointing gesture. In the near condition, eight out of ten speakers always produced multimodal referring expressions containing a demonstrative determiner, 'deze' (this), or no spoken material at all. The remaining two consistently added a head noun, e.g., 'deze driehoek' (this triangle). When, on the other hand, an imprecise pointing gesture is used, because of the distance to the target, the referring expressions contain much more spoken material. The kind of target object also had an influence on this. In general, fewer words are required to single out a geometrical figure than to identify a person, in the current experiment. Closer inspection of the data reveals that both objects and persons are described in terms of their type (e.g., 'triangle' and 'man' respectively). In addition, geometrical objects are more often accompanied by prenominal adjectives (e.g., 'blue'), while person descriptions tend to include locative expressions (e.g., 'in the top left corner'). This is probably due to the fact that describing persons is inherently more difficult than describing colored geometrical objects, since the number of potentially relevant attributes is much larger for persons than for geometrical objects.

5.4 Study 2: Pointing and Conversation

5.4.1 Overview

A disadvantage of the first study is that subjects were forced to point, and that the size of target object was kept constant. Therefore a second study is conducted in which subjects performed a topographical task in a more natural and interactive setting. 20 subjects (different from those in the first study) participated and were asked to locate countries on a world map. Again the subjects performed their tasks at two distances: close (10 subjects) and at a distance of 2.5 meters (10 subjects). The target objects in this study were selected in such a way that there was a distinction between the objects that are **easy to locate** (large or isolated countries) and the objects that are **difficult to locate** (small countries). In Chapter 4 it was argued that two factors that influence object identification are target size (some targets are easier to point to than others), and target distance (an object that is closer is easier to point to than an object that is further away). In the algorithm both these factors are combined and weighted in Fitts' law, an empirical measure of the difficulty people have in reaching a target (Fitts, 1954). This raises a number of questions, for example: (1) What is the influence of target size and distance on the decision to point? One would expect that people use more gestures when referring to easily reachable targets (large and/or close ones); (2) How are the pointing gesture and the linguistic information related? and (3) In what way and how much do relata occur in the referring expressions? Especially in describing difficult targets, easily recognizable relata may be helpful in identifying the target.

Section 5.4.2 describes the experiment conducted, gives a general overview, and discusses subjects, experimental setting, materials and data processing. In Section 5.4.3 the results of the experiment are presented: the interaction of language and speech, an analysis of the linguistic material and the gestures. Section 5.4.4 ends this section with a discussion.

5.4.2 Method

General Design

A production experiment was conducted to elicit multimodal referring expressions. Subjects had to perform an object identification task, in which they had to identify countries on a political world map like the one presented in Figure 5.4. The size of the world map is 100 by 140 cm. Half of the target countries were **easy** to locate (i.e., large or isolated), the other half was **difficult** to locate (i.e., small). Half of the subjects performed the tasks at a close distance (i.e., they could touch the target country directly), the other half of the subjects performed the same tasks from a distance (i.e., they could only indicate the location of the target).

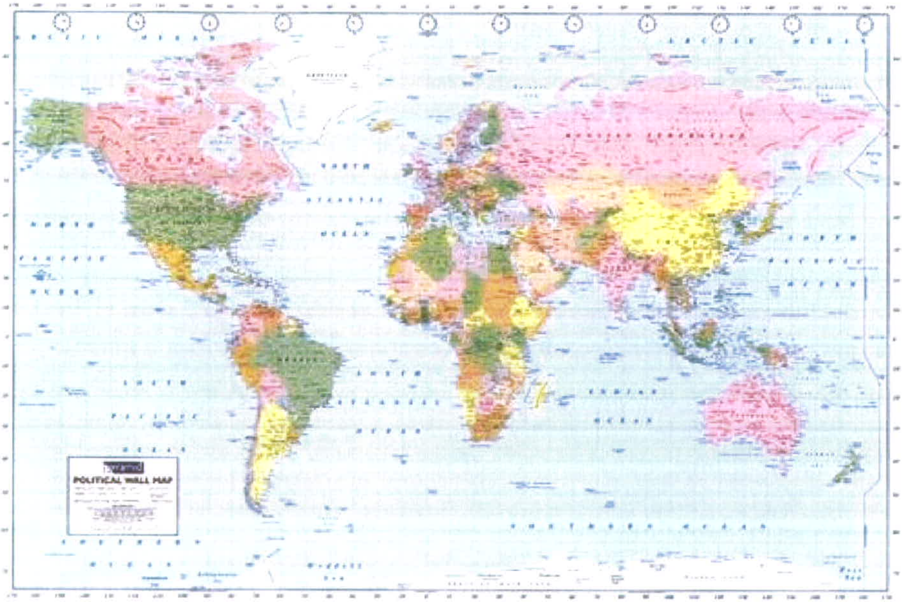


Figure 5.4: Political world map.

Table 5.4 summarizes the experimental design, with *target* as a within-subject variable and *distance* as between-subject variable.

		DISTANCE	
		NEAR	FAR
TARGET	EASY	I	II
	DIFFICULT	III	IV

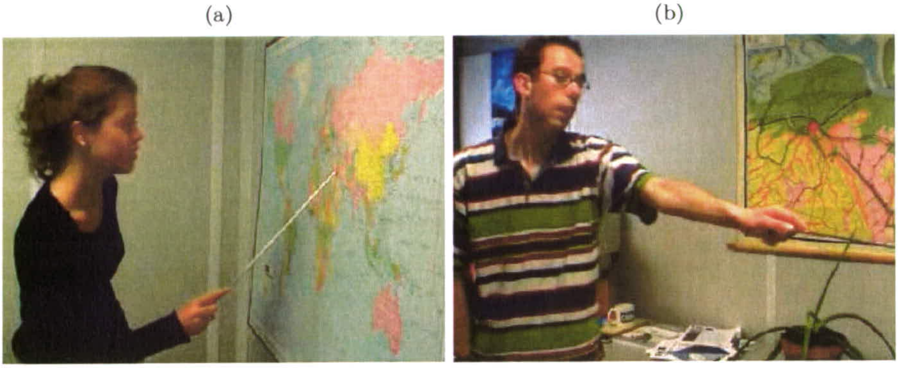
Table 5.4: Overview of the experimental design with *distance* as between-subject and *target* as within-subject variables.

Subjects

Twenty native speakers of Dutch participated as subjects in this study. All were students and colleagues from Tilburg University that did not participate in the first study. For each condition, the group of subjects consisted of five men and five women.

Experimental Setting

Subjects were told that their topographical knowledge was going to be tested just like in primary school. Half of the subjects performed the experiment in the **near condition** (precise pointing). The other half of the subjects, those in the **far condition**, were placed on approximately 2.5 meters from the map. Subjects in the far condition could use an imprecise pointing gesture to point in the direction where the target was located. By definition their pointing gestures were always imprecise. In Figure 5.5 an example is shown of each condition. Subjects were given a stick of 40 cm they could use for pointing if they so desired. Although the subjects used different strategies to identify targets, all subjects were consistent in their behavior during the task. Subjects were asked to be more specific when they made unclear references.


(a)

(b)

Figure 5.5: Example of subjects in the near condition(a) and in the far condition (b).

Stimuli

30 countries were selected which are easy to find and which can be divided into two kinds of target objects: 15 relatively small countries and 15 relatively large or isolated countries. Isolated countries, like islands or groups of islands, stand out because of their shape or color and they are considered to be as easy identifiable as the larger countries. The relatively small countries, like for example Italy, are called difficult targets, because they cannot be distinguished with an imprecise

pointing gesture and their description requires some effort. The large or isolated countries, for example Russia or Australia, are called easy targets because they can be identified with an imprecise pointing gesture and some straightforward linguistic referring expression. Except for the variability in size, the countries also differ in shape and color. The 30 target objects were presented to the subjects in a random order.

Data Processing

Subjects were filmed during the experiment. The resulting data consist of (20 subjects $\times$ 30 stimuli) $=$ 600 multimodal referring expressions. All utterances were transcribed. The pointing gestures were classified, and the kinds of linguistic attributes were determined and counted. All subjects produced a distinguishing description for each target. All tests for statistical significance were done using an analysis of variance (ANOVA) with repeated measures, with distance as between-subject variable and target as within-subject variable.

5.4.3 Results

Interaction of Language and Speech

All subjects always used a pointing gesture even though they were not explicitly instructed to do so. In the near condition, this pointing gesture is always a precise one, where the target is directly touched. In the far condition subjects by definition employed imprecise pointing gestures, which basically denote in what area on the map the target is located. This indicates that the variation in distance inevitably worked as intended.

First, the number of words and the number of disfluencies as defined in Section 5.3.2 in the multimodal referring expressions are considered; Table 5.5 presents the results. For the number of words there is an effect of distance ($F(1,18) = 241.04, p < .01$), and an effect of target ($F(1,18) = 33.12, p < .01$). These effects indicate that in the far condition subjects use more words than in the near condition and subjects require more words to refer to difficult objects than to easy ones. In addition, there is an interaction between distance and target ($F(1,18) = 23.93, p < .01$). This can be explained by observing that the effect of target is stronger in the far condition than in the near condition. The number of disfluencies show an effect of distance ($F(1,18) = 100.44, p < .01$) and an effect of target ($F(1,18) = 6.44, p < .05$), which indicates that both in the far condition and when referring to difficult objects subjects speak less fluently. Furthermore there is an interaction between distance and target ($F(1,18) = 7.17, p < .05$) which signals a stronger effect of target in the far condition compared to the effect of target in the near condition. In the near condition subjects do not use many words to refer to easy or difficult objects, consequently disfluencies are scarce.

			DISTANCE	
			NEAR	FAR
TARGET	EASY	**words**	2.28(1.09)	15.59(3.10)
		disfl	.19(.10)	1.57(.85)
	DIFFICULT	**words**	3.23(1.63)	27.25(6.28)
		disfl	.17(.11)	2.40(.65)

Table 5.5: Average number of *words* and *disfluencies* per description as a function of *distance* and *target*. Standard deviations are given between brackets.

Analysis of Linguistic Material

Thus, subjects appear to co-vary the linguistic material with the kind of pointing gesture they use. Although there were some differences observed among the subjects, especially in the far condition, each of the subjects displayed consistent behavior throughout the experiment. In the far condition, all subjects used imprecise pointing gestures, and hence were required to use more additional linguistic material to produce an unambiguous reference. For example a typical description of an easy object like Brazil is 'dat grote groene vlak daar' (that large green area over there) together with an imprecise pointing gesture. As an example of a difficult object, consider a description of Portugal: 'Portugal ehm is het eh groene land dat ten zuid westen of dat eh in zuid europa ligt naast het roze Spanje' (Portugal uhm is the uh green country which lies on the south west or which uh lies in southern Europe next to the pinkish Spain) together with an imprecise pointing gesture to indicate Portugal. In the near condition, a precise pointing gesture suffices to single out the target. Additionally, the name of the target is sometimes mentioned together with a 'here' or a 'there'.

Table 5.6 presents a more detailed analysis of the linguistic material with respect to the following features:

- *Occurrences of Name* Per target description, the number of times the name of the target is mentioned is counted (like 'Portugal' in the example above).
- *Occurrences of Type* Per target description, the number of *type* properties, in Dutch mostly head nouns, used to describe the target are counted (whether the target is called a 'country', 'area', 'isle', 'spot', 'part' etc., i.e., the information typically given in the head noun).
- *Occurrences of Properties* Per target description, the number of prenominal properties (with the exception of *type*) that are included to describe the target are counted (e.g., *color*, *size*, *shape*, etc.).
- *Number of Locative Expressions* Per target description, the number of locative expressions are counted. Locative expressions can be split into at least

two types: (1) 'in het zuiden' (in the south), as a general reference to a part of the world; and (2) 'naast het roze Spanje' (next to the pinkish Spain) as relatum. In the latter case 'naast het roze Spanje' as a whole is treated as a locative expression.

- *Number of relata* Per target description, the number of relata are counted. In the example of Portugal, the number of relata is two: 'Europa' (Europe) and 'Spanje' (Spain). The descriptions that identify relata, for example 'het roze Spanje' (the pinkish Spain) are dealt with separately.

			DISTANCE	
			NEAR	FAR
TARGET	EASY	**name**	.32(.26)	.84(.24)
		type	.03(.05)	.92(.29)
		property	.04(.06)	1.51(.20)
		location	.12(.13)	.18(.68)
		relata	.05(.08)	1.11(.46)
		-name	.03(.06)	.81(.60)
		-type	.00(.00)	.40(.32)
		-property	.00(.00)	.28(.46)
		-location	.03(.06)	1.35(.65)
	DIFFICULT	**name**	.33(.28)	1.07(.18)
		type	.04(.07)	.76(.16)
		property	.07(.08)	1.30(.21)
		location	.13(.18)	2.87(.98)
		relata	.11(.15)	2.21(.56)
		-name	.03(.06)	2.01(.95)
		-type	.03(.06)	.78(.45)
		-property	.00(.00)	.81(.34)
		-location	.09(.13)	2.72(.85)

Table 5.6: Average numbers of the attributes *name*, *type*, *property*, *location* for targets and *relata* given per description as a function of *distance* and *target*. Standard deviations are given between brackets.

In Table 5.6 *name*, *type*, *property*, *location*, are also presented for all relata used in all descriptions. First consider the between-subject effects, the near versus the far condition. The results show that for almost all features there is a significant effect of distance (*name*, $F(1,18) = 41.21, p < .01$; *type*, $F(1,18) = 132.21, p < .01$; *property*, $F(1,18) = 554.75, p < .01$; *location*, $F(1,18) = 76.57, p < .01$; *relata*, $F(1,18) = 119.787, p < .01$). Thus, in the far condition, speakers use more names, more *type*, *property* and *location* information and more *relata* to identify a target object. Looking at the within-subject effects, difficult versus easy objects, the results show that subjects tend to use more type and property information when referring to large objects (*type*, $F(1,18) = 5.96, p < .05$ and *property*, $F(1,18) = 5.94, p < .05$). In descriptions for difficult objects subjects use more *names*, *locations* and *relata* (*name*, $F(1,18) = 5.03, p < .05$; *location*, $F(1,18) = 27.72, p < .01$; *relata*, $F(1,18) = 51.157, p < .01$). In a comparison of the references for easy objects to those for difficult objects, it can be noted that the differences are almost non-existent in the near condition, while they are substantial in the far condition (*name*, $F(1,18) = 4.91, p < .05$; *type*, $F(1,18) = 6.99, p < .05$; *property*, $F(1,18) = 9.53, p < .05$; *location*, $F(1,18) = 27.24, p < .01$; *relata*, $F(1,18) = 41.149, p < .01$). Interestingly, in the far condition, easy objects are more often referred to using head nouns and properties, while descriptions of difficult objects tend to contain more locative expressions and relata.

In the separate analysis of relata, there are significant effects of distance for all features, (*name*, $F(1,18) = 33.964, p < .01$; *type*, $F(1,18) = 31.398, p < .01$; *property*, $F(1,18) = 23.139, p < .01$; *location*, $F(1,18) = 75.887, p < .01$) which can be explained by the fact that relata almost exclusively occur in the far condition. Moreover, all features used to describe relata display effects of target in the sense that in descriptions of easy objects, all these features are used less compared to their occurrences in references to difficult objects (*name*, $F(1,18) = 50.562, p < .01$; *type*, $(F(1,18) = 8.491, p < .01$; *property*, $F(1,18) = 18.656, p < .01$; *location* $F(1,18) = 66.822, p < .01$). When comparing the near and the far condition, the effects of target for the features used to describe relata are large (*name*, $F(1,18) = 49.450, p < .01$; *type*, $F(1,18) = 5.961, p < .05$; *property* $F(1,18) = 18.656, p < .01$; *location* $F(1,18) = 57.136, p < .01$). Hence, in the far condition subjects tend to use more attributes to describe relata of difficult objects, in comparison to the number of attributes used in describing relata of easy objects.

Analysis of Gestures

In Table 5.7 an analysis of the occurrences of gestures made during the references is presented, where the following features are considered:

- *Total number of pointing gestures* Per target description, pointing gestures directed towards the target and directed towards the relata are counted.

- *Pointing gestures directed towards the target* Per target description, pointing gestures directed towards the target are counted.

- *Pointing gestures directed towards relata* Per target description, pointing gestures directed towards relata are counted.

- *Number of static pointing gestures* Per target description, all static pointing gestures are counted, gestures directed towards the target as well as gestures directed towards relata. **Static pointing gestures** display no movement during the stroke of the gesture.

- *Number of dynamic pointing gestures* Per target description, all dynamic pointing gestures are counted, gestures directed towards the target as well as gestures directed towards relata. **Dynamic pointing gestures** are defined as gestures that include some kind of movement in the stroke of the gesture, vertical, horizontal or circling. As suggested by (Kendon, 2004, page 201-205), pointing gestures can be deictic and characterizing at the same time. Additional movement during the stroke of the pointing gesture might indicate a property of the object.
- *Number of vertical dynamic pointing gestures* Per target description, the number pointing gestures that display vertical movement during the stroke of the gesture are counted.
- *Number of horizontal dynamic pointing gestures* Per target description, the number pointing gestures that display horizontal movement during the stroke of the gesture are counted.
- *Number of circular dynamic pointing gestures* Per target description, the number pointing gestures that display circular movement during the stroke of the gesture are counted.

Table 5.7 shows that all subjects pointed at every target at least once, no matter the distance or size. Although it is hard to distinguish difficult objects with imprecise pointing gestures, surprisingly, subjects in the far condition tend to point even more often (almost twice) to difficult objects. When the number of *total pointing gestures* is considered in more detail, it appears that subjects in the far condition direct considerably more pointing gestures *to relata* in describing difficult objects than in describing easy objects. Apart from the distribution of pointing gestures, also the kinds of precise and imprecise pointing gestures are looked at. Most precise pointing gestures are of a static nature, whereas the imprecise pointing gestures display a greater variability: between static and dynamic gestures and also within the dynamic gestures.

More specifically, the total number of pointing gestures displays both an effect of distance ($F(1,18) = 24.52, p < .01$) and an effect of target ($F(1,18) = 13.45, p < .01$), which indicate that subjects in the far condition use more pointing gestures especially with references to difficult objects. Moreover, a significant interaction was found between target and distance ($F(1,18) = 11.62, p < .01$). In contrast, pointing gestures that indicate a relatum display effects both of target ($F(1,18) = 14.17, p < .01$) and distance ($F(1,18) = 19.44, p < .01$). In the near condition there are no such pointing gestures because relata usually do not occur. In the far condition, except for pointing at the target, subjects also use

			DISTANCE	
			NEAR	FAR
	EASY	**total**	1.00(.00)	1.32(.22)
		to target	1.00(.00)	1.13(.14)
		to relata	.00(.00)	.20(.17)
		static	.85(.32)	.66(.33)
		dynamic	.17(.31)	.70(.40)
		-vert	.01(.02)	.23(.14)
		-hor	.03(.04)	.09(.09)
		-circ	.13(.32)	.37(.34)
TARGET				
	DIFFICULT	**total**	1.02(.04)	1.86(.59)
		to target	1.02(.05)	1.03(.21)
		to relata	.00(.00)	.85(.64)
		static	.90(.25)	1.11(.41)
		dynamic	.15(.26)	.73(.26)
		-vert	.01(.02)	.30(.18)
		-hor	.03(.03)	.05(.05)
		-circ	.10(.25)	.37(.50)

Table 5.7: Average numbers of pointing gestures (e.g., given per description as a function of *distance* and *target*. The *total* number of pointing gestures is divided in the pointing gestures directed *to target* and *to relata* as well as in *static*, *dynamic* pointing gestures. The dynamic pointing gestures are subdivided in gestures containing *vert*ical, *hor*izontal and *circ*ular movement. Standard deviations are given between brackets.

pointing gestures to indicate relata, especially when the target is difficult to describe. The interaction between target and distance ($F(1,18) = 14.17, p < .01$) signals a difference in target effect. The type of pointing gestures used in the near condition is, in almost all cases, static. In the far condition the type of pointing gestures varies; dynamic pointing gestures are almost used as often as static ones. The static pointing gestures only display an effect of target ($F(1,18) = 19.34, p < .01$), which indicates that subjects tend to use more static gestures to identify difficult objects. There is no effect of distance, but there is an interaction ($F(1,18) = 12.06, p < .01$), which implies that the effect of target differs significantly between the far and the near condition. Dynamic gestures only display an effect of distance ($F(1,18) = 11.16, p < .01$): subjects use more dynamic gestures in the far condition. The effects of distance are only present for horizontal and vertical pointing gestures (respectively $F(1,18) = 17.594, p < .01$ and $F(1,18) = 25.321, p < .01$).

5.4.4 Discussion

A production experiment conducted in a natural, interactive setting was described where subjects produced distinguishing descriptions for selected target objects. The experimental results, contrary to expectation, indicate that speakers always include pointing gestures in their descriptions regardless of the difficulty of the target and the distance to the target. This could be a result of the fact that the subjects were equipped with a stick with which they could point, or simply because the nature of the task provokes pointing gestures. When the target is close, speakers tend to point only once in the direction of the target in a static fashion. When the target is located at a larger distance, the variability in the kind of pointing gestures increases. In half of the cases in which a pointing gesture is used, some movement is made during the stroke. Surprisingly, speakers use more pointing gestures to refer to difficult targets than to easy targets. A closer inspection of the data shows that the extra pointing gestures are directed towards relata and not the target. Furthermore speakers co-vary the linguistic part of a multimodal referring expression with the distance to the target and the kind of target. When the target is close, speakers reduce the linguistic material to almost zero, whereas subjects tend to produce overspecified descriptions if the target is located further away (in line with earlier work by, for instance, Pechmann, 1989). This can be due to the inherent uncertainty of imprecise pointing. Speakers may not be sure whether the imprecise pointing gesture is sufficiently clear and so, to guarantee that their reference is distinguishing, they include additional information. As expected, descriptions of difficult targets often contain less fluent speech (more uhs/ums), because more speaker effort is required (e.g., Goldman-Eisler, 1968; Clark and FoxTree, 2002). Typically the features of difficult targets are harder to recognize at a distance and there is a tendency to include descriptions of relata to indicate the location of the target. In describing difficult targets, speakers include the name of the target together with at least one property and almost three locative expressions. In contrast, descriptions of easy targets, generally include a head noun and one or two adjectival properties.

5.5 Output of the Multimodal Algorithm

The graph-based algorithm presented in Chapter 4 does not use an a priori criterion to decide when to use a pointing gesture. The output modality is determined by a trade-off between the costs of pointing and the costs of a linguistic description, which have to be defined on an empirical basis. The underlying assumption of the algorithm for the generation of multimodal referring expressions is twofold: (1) the amount of linguistic information necessary to identify a target co-varies with the type of pointing gesture included; and (2) the linguistic information and pointing gesture depend on the kind and the size of the target. In this chapter two production experiments were presented to evaluate the output of the graph-based algorithm. Study I was conducted in a very strict setting where the distance to the objects and the target objects themselves were varied. Study II was conducted in a more natural setting, where both the distance to the objects and the size of the target objects were varied.

Overall, it may be concluded that the co-variation of the linguistic material and the kind of pointing gesture corresponds well with the results of the studies. The two studies show some clear differences between speakers (in line with earlier work, for instance by Piwek and Beun (2001)). In the near condition, most speakers reduce the linguistic material almost to zero. Note that the algorithm from Chapter 4 agrees with the majority of speakers concerning the fact that the more precise the pointing gesture, the less linguistic material is generated to refer to an object. Accordingly, a precise pointing edge and a relational edge never occur together in a distinguishing graph.

In the far (but not in the near) condition, subjects tend to produce more overspecified descriptions (in line with earlier work, for instance by Dale and Reiter (1995)). This is also in line with the observation made by Oviatt (1999) that information can be presented simultaneously via multiple modalities to facilitate processing of the utterance. However, the algorithm makes a different predictions when it comes to overspecification. This is due to the fact that the search strategy used in the algorithm is aimed at providing minimal descriptions. It is worth stressing though, that different search strategies are compatible with the graph-based perspective. Krahmer et al. (2003) illustrate this by describing a different search strategy which mimics the Incremental Algorithm and thus gives rise to a certain amount of redundancy. However, it can be said that overspecification as generated by the Incremental Algorithm (see Section 3.3.2) is caused by chance, in that it depends on the objects in the distractor set. Therefore in Chapter 6 the algorithm is equipped with a more systematic method of generating overspecified referring expressions. The method addresses both unimodal and multimodal overspecification, where multimodal overspecification also takes into account dynamic pointing gestures and their relation to the shape of the target.

Chapter 6

Overspecification in GRE

The task involved in the generation of multimodal referring expressions is that of deciding what the best way is to refer to a target via combinations of modalities in the current context. As outlined in Chapter 3, most algorithms that generate referring expressions focus on **minimal referring expressions** (i.e., the shortest distinguishing descriptions possible for a given referent). However, as seen in Chapter 5, many referring expressions produced by human speakers are **overspecified** (i.e., distinguishing but not minimal see also, e.g., Pechmann, 1989; Beun and Cremers, 1998; Arts, 2004). In order to mimic human production of overspecified referring expressions in automatic generation, this chapter discusses overspecification by considering two questions: (1) Why and when do speakers overspecify? and (2) How do speakers overspecify? This chapter attempts to answer these questions by addressing overspecification as occurring in human communication on the basis of empirical findings. The phenomenon of overspecification is understudied in the field of NLG, but it has been addressed by a number of researchers in (cognitive) linguistics. Inspired by their observations and in particular the findings from Chapter 5, a variant of the multimodal algorithm is proposed that can generate overspecified descriptions based on strategies observed in human communication.

This chapter is organized as follows. In Section 6.1 an analysis is presented of the kinds of overspecification that are employed by human speakers, both unimodally and multimodally. In Section 6.2 the graph-based algorithm proposed in Chapter 4 is adjusted in such a way that it is able to generate overspecified referring expressions, based on the outcomes of the analysis. The performance of the new algorithm is addressed in Section 6.3, in which examples from human communication taken from the studies from Chapter 5 are compared to the algorithm's output. The chapter ends with a discussion. A preliminary version of the ideas in this chapter was described in van der Sluis and Krahmer (2005).

6.1 Overspecification in Human Communication

In order to improve and adapt the multimodal graph-based algorithm to human reference behavior, in this section two kinds of overspecification as it occurs in human communication are discussed: (1) **Unimodal overspecification**, which is due to the inclusion of more linguistic properties than necessary for singling out a target; and (2) **Multimodal overspecification**, caused for example by the production of pointing gestures together with locative expressions. Section 6.1.1 discusses unimodal overspecification with respect to the observations in linguistic studies. In Section 6.1.2 the multimodal referring expressions of the two studies presented in Chapter 5 are analyzed anew, considering multimodal overspecification. In Section 6.1.3 the findings of both sections are combined in a discussion that aims at an application to automatically generated multimodal referring expressions.

6.1.1 Unimodal Overspecification

In order to make the notion of unimodal overspecifation more explicit, Example Domain I in Figure 6.1, first shown in Chapter 3, is reproduced below. Some possible ways to refer to object d_3 are displayed in Figure 6.2.

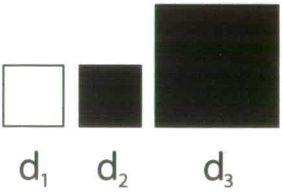


Figure 6.1: Example Domain I.

(1) 'the square'
(2) 'the black block'
(3) 'the large block'
(4) 'the large black block'
(5) 'the large black square block'
(6) 'the large black block on the right'

Figure 6.2: Possible realizations for d_3.

Of the descriptions in Figure 6.1, (1) and (2) are underspecified because they fail to distinguish the target, d_3. Underspecified descriptions lack the crucial properties with which a target can be singled out from the other objects in its context. Hence, description (1) fails because all objects in Example Domain I have the property *square*, and description (2) fails because d_2 is also black. The descriptions (3), (4), (5) and (6), are all distinguishing; they are only applicable to d_3. Description (3) is a minimal description; it refers to d_3 in the shortest way possible in terms of the number of properties. The descriptions (4), (5) and (6) are overspecified, because they use more properties than necessary to refer to d_3. While d_3 is the only large block in Example Domain I, description (4) contains the redundant property *black*; description (5) contains two redundant properties, namely *black* and *square*; and description (6) contains three redundant properties.

Even though minimal descriptions seem to require less effort, speakers often produce overspecified descriptions. Why? In the literature a number of partially overlapping suggestions can be found. For instance, Pechmann (1989) explains overspecification from the assumption that language production is incremental in nature, meaning that the perceived distinguishing properties are almost simultaneously verbalized (c.f., also Kempen and Hoenkamp, 1987; Clark and Clark, 1977). According to this view, speakers are highly affected by their perception of the domain of conversation. This causes for example easily perceptible properties to be mentioned earlier than other object properties (c.f., Mangold and Pobel, 1988). In this incremental process, the speaker may find out that the target is not distinguished by the first property included in the description. Consequently, the speaker adds more properties to the description. It may be that a first property that was included is made redundant by the inclusion of a later property, which leads to an overspecified referring expression. This view is consistent with current theories of reference proposed by Ariel (1991; 2001) and Gundel et al. (1993). These theories explain the degree of overspecification in terms of accessibility or focus of attention, which is influenced by features like the absolute properties in the domain, the discourse history and the focus space. Thus, the less accessible or salient an object in the discourse, the more overspecified the referring expression used to indicate the object. In line with this view is the finding of Beun and Cremers (1998) that speakers tend to use overspecification specifically to refer to objects that are outside the focus space.

Apart from object or domain related influences and aspects that concern language production itself, factors that relate to the discourse have also been argued to play a role in the production of referring expressions (c.f., Jordan, 2002; Maes et al., 2004). Such factors are for instance discourse goals or task importance and the different modes of communication and situational conditions (c.f., Goodman, 1986; 1987). With respect to these factors, Maes et al. (2004) (see also Arts, 2004) state that overspecified referring expressions are affected by the *principle of distant responsibility* (Clark and Wilkes-Gibbs, 1986), which says that a speaker or writer must be certain that the information provided in an utterance is understandable for the user. The experiments performed by Maes et al. (2004) on written instructive texts indicate that, when both participants in a discourse have access to a visual domain of conversation and no feedback is given, highly overspecified referring expressions are produced in order to reduce uncertainty. Notice that these findings cannot be explained from Pechmann's assumption that overspecification is due to the incremental nature of language production, because the experiments performed by Maes et al. involve written texts, where the participants had sufficient time to consider their instructions to the reader. More evidence for the relation between overspecification and discourse goals is provided by Jordan (2002), who finds that the overspecification found in the COCONUT corpus is typically produced in situations in which the speaker wants to stress a commitment, persuade the hearer, or focus on a change in the constraints of the task. The fact that the importance of the task triggers the speaker to put in extra effort (i.e., overspecify a description) to aid the hearer is also argued for by Arts (2004). Arts addresses the contrast between the principle of distant responsibility, which causes overspecification, and Grice's maxim of quantity, which essentially says that one should not include more information than necessary. Arts reports on a series of object identification tasks. In the experiments the participants are first confronted with a description of the target object, after which a group of objects, including the target, is presented on a computer screen. The task was to identify the target based on the description and indicate this identification by pressing a button on the keyboard. The correlation between the object description and the identification time is analyzed, where identification time is defined as the length of the time interval that starts at the instant the object domain is presented to the participant, and ends when the participant presses the key which signals that she has identified the target. Interestingly, the average identification time for minimally specified expressions does not differ from the average identification time for overspecified referring expressions. From this it can be inferred that overspecification is employed to facilitate identification. Moreover, perception experiments indicate that overspecification sometimes even leads to faster identification times (c.f., Deutsch, 1976; Sonnenschein, 1982, 1984; Mangold and Pobel, 1988; Pechmann, 1989; Cremers, 1996; Beun and Cremers, 1998; Campana et al., 2004).

What kinds of overspecification do speakers actually produce as a result of the factors mentioned above? Surprisingly, in the experiments reported by Maes et al. (2004), in which the participants had to write instructions on how to use an alarm clock for readers that needed to operate on such a device, functional properties are rarely used. In contrast, the kinds of properties used to describe the target are all of a perceptual nature. Moreover, experiments by Pechmann (1989), Beun and Cremers (1998) and Arts (2004) imply that those properties that are easily perceived are likely to be included in a referring expression in order to facilitate identification for both speaker and hearer. Based on experiments with a block domain, Mangold and Pobel (1988) propose a hierarchy of properties dependent on the effort it takes to perceive these properties. This hierarchy starts with the property *color*, which is perceived easiest, followed by *size* and *shape*, which appeared to be more difficult to observe. However, in a comparison with Arts (2004, page 114), who infers from perception experiments that *shape* is an important factor in constructing a mental image of the target (c.f., Ariel, 2001) which aids the identification process of the hearer, it might be best to conclude that the preferences of these properties is domain dependent. Two tendencies in overspecification emerged in the experiments by Mangold and Pobel (1988): (1) Properties that remove some distractors are more likely to be mentioned than properties that are not distinguishing at all, (i.e., to describe object d_3 in Figure 3.1 an overspecified description is used that contains two distinguishing properties; *square* is not included); and (2) If the distinguishing property is not easily perceived (i.e., in case of Mangold and Pobel's hierarchy, if the *shape* property is the only property by which the target differs from the other objects), highly overspecified referring expressions are produced. The perception and production experiments in a block domain conducted by Arts (2004, page 111) show that, in particular, the inclusion of locative expressions is highly beneficial; compared to object descriptions that included only absolute properties like *shape* and *color*, objects referred to by overspecified descriptions that included locative expressions were identified faster. Moreover, in a comparison between distinguishing descriptions that solely contained locative expressions and overspecified descriptions that included locative expressions together with absolute properties, Arts found that the absolute properties merely increased identification time (c.f., Kato and Nakano (1997)). Arts considers the addition of locative expressions as "linguistic pointing" that mirrors the physical pointing gestures. From this perspective locative expressions can be explained as an extra effort to facilitate identification.[1]

6.1.2 Multimodal Overspecification

In this section, an explorative study with respect to multimodal overspecification is performed on the data resulting from the two studies presented in Chapter 5. The analysis aims at a specification of the kind and degree of multimodal overspecification produced by human speakers in relation to three factors: (1) The distance to the target object; (2) The complexity of the target (e.g., blocks versus photographed persons) object; and (3) The size of the target object. For both studies, features are defined from which the degree and the kind of multimodal overspecification can be determined. For each of the features an analysis

[1] Note that in the reading experiments conducted by Arts, the time it took the subjects to read the description presumably increased, which can be explained by the fact that a locative expression causes the referring expression to lengthen with approximately three words.

of variance (ANOVA) with repeated measures is performed to test for significance.

Study I, described in Section 5.3, addresses an experiment in which pointing was forced. Because the participants obligatorily produced pointing gestures, the pointing gestures themselves are not considered in this analysis. However, the effect of the pointing gestures, namely directing the attention to a part of the domain, is taken into account. In the experiment the objects were presented on the screen in two isolated groups (see Figure 5.2 and Figure 5.3 in Section 5.3.2), one containing the target (the **target group**), while the other group solely consisted of distractors (the **distractor group**). In analyzing overspecification, only the objects that are in the target group are considered as distractors, because the pointing gestures that the participants were obliged to produce restrict the focus of attention to the section of the screen where the target group is located. Study I resulted in 600 referring expressions, i.e., distinguishing descriptions for the target objects. These descriptions are analyzed with respect to overspecification by extracting three features from each target description:

- *Overspecification by type properties* As already mentioned, in Dutch, *type* properties are mostly head nouns that describe the target. In a block domain these head nouns typically express the shape of an object. This feature counts the number of *type* properties that can be removed from the target reference while keeping a distinguishing description.

- *Overspecification by object properties* This feature counts the number of verbalized target properties (with the exception of *type*) that can be removed from the target reference while keeping a distinguishing description.

- *Overspecification by locative properties* This feature counts the number of locative expressions that can be removed from the target reference while keeping a distinguishing description.

In Study I two groups of subjects had to identify geometrical objects and photographed persons. One group of subjects performed the task standing close to the computer screen on which the stimuli were projected, the subjects in the other group were positioned somewhat further away from the screen. The 30 stimuli presented to each subject consisted of 15 geometrical objects and 15 photographed persons. The geometrical objects are referred to as **simple objects**, whereas the photographed persons that display a variety of different properties, are considered **complex objects**. As seen in Section 5.3, linguistic referring expressions were used mainly in the far condition, whereas in the near condition subjects tend to use only a precise pointing gesture to indicate the target. When looking at the linguistic descriptions in the far condition, a typical example of a referring expression to indicate a geometrical object is 'het rode vierkant' (the red square).

In cases where 'het vierkant' (the square) would have been the minimal description, in the analysis presented here 'rode' (red) is analyzed as redundant, leading to an overspecified description of the target. With respect to the photographed persons, subjects produced referring expressions like 'de man met de bril links' (the man with the glasses on the left) where 'de man' (the man) would have been the minimal description since the target group contained only one male person. In such cases *met de bril* (with glasses) and *links* (on the left) are respectively analyzed a redundant property and a redundant locative expression. In general, for all stimuli in Study I it could be unambiguously established in what way the descriptions were overspecified.

From the descriptive means and standard deviations presented in Table 6.1 it can be inferred that, in the far condition, subjects use more overspecified descriptions to indicate a target, object or person, when the target is located further away than when it is located at a short distance. More specifically, subjects use more redundant properties and more redundant locative expressions when referring to objects that are located far away. The between-subject analysis shows that descriptions contain significantly more redundant properties ($F(1, 18) = 27.45, p < .01$) and more redundant locative expressions ($F(1, 18) = 9.27, p < .01$) in the far condition than in the near condition. From the within-subject analysis it can be concluded that there is a difference in the way the descriptions are overspecified. Subjects produce significantly more redundant locative expressions when referring to persons ($F(1, 18) = 14.88, p < .01$), whereas they use more redundant properties to indicate geometrical objects ($F(1, 18) = 48.52, p < .01$). The interaction between the two factors signals a stronger effect of target in the far condition than in the near condition for both redundant properties ($F(1, 18) = 12.13, p < .01$) and redundant locative expressions ($F(1, 18) = 15.94, p < .01$).

			DISTANCE	
			NEAR	FAR
TARGET	PERSON	**type**	.08(.17)	.03(.08)
		property	.03(.11)	.28(.17)
		location	.07(.21)	.69(.38)
	OBJECT	**type**	.15(.33)	.00(.00)
		property	.19(.33)	.74(.11)
		location	.08(.25)	.30(.41)

Table 6.1: Descriptive means and standard deviations of the redundant *type*, *property* and *location* information contained in the multimodal referring expressions resulting from Study I, where two groups of subjects, one in the *near* condition and one in the *far* condition, identified geometrical *objects* and photographed *persons*. Standard deviations are given between brackets.

From this analysis it can be concluded that when the target is located far away, speakers more often produce overspecified referring expressions than in cases where the target is located close by, even when imprecise pointing gestures are included. This effect might be explained by the uncertainty of the speaker about the imprecise pointing gesture. The speaker wants to be sure that the hearer understands the reference correctly and therefore provides extra information. This uncertainty may also explain the fact that speakers prefer absolute properties to refer to easily identifiable geometrical objects, while locative information is favored when referring to objects that are difficult to distinguish from their distractors. If speakers are unsure about the identifiability of a complex object, like a photographed person, they may decide on identification using unambiguous locative information. In contrast, speakers choose to use absolute properties to distinguish simple geometrical objects that are not so easily confused.

Study II, described in Section 5.4, consisted of an experiment in which pointing was optional. The multimodal referring expressions that result from this experiment can be analyzed with respect to multimodal overspecification. In the following, two types of multimodal overspecification are defined with which the data from Study II is approached. As noted in Section 2.4.2 pointing gestures are used to indicate an object or an area, but they can be used to characterize an object as well (Kendon, 2004). Accordingly, the shape or orientation of an object can be indicated by the inclusion of movement during the stroke of the gesture. With respect to these possible interpretations of pointing gestures, it is interesting to see to what extent the pointing gestures overlap with the linguistic properties used in the accompanying referring expressions, for instance with locative expressions or with linguistic indications of the shape of the target. Accordingly, the data, which results from Study II, is examined on two kinds of multimodal overspecification: (1) Pointing gestures that co-occur with locative expressions; and (2) Dynamic pointing gestures that co-occur with descriptions that contain the property *shape*.

With respect to the co-occurrence of pointing gestures and locative expressions, a locative expression is considered a linguistic pointing gesture following Arts (2004). Arguably, multimodal overspecification arises in cases where a pointing gesture and a locative expression are used together in a referring expression to indicate the same target or the same relatum, where one of them would be distinguishing. However, because both the scope of a pointing gesture and the scope of a locative expression might be vague, it is uncertain to what extent they converge. As an example reconsider the Flashlight Model for pointing as presented in Figure 4.4. For the sake of simplicity, the scopes of the various pointing gestures in this figure are defined very precisely. For both precise and imprecise pointing gestures, it is clear which of the objects in the domain are contained in the scope of the gesture. In reality it is not that simple. The scope of a precise pointing gesture may uniquely indicate one object, but, at least from a hearer's point of view, the boundaries of the scope of an imprecise pointing gesture is vague. Although imprecise pointing gestures can direct the attention to the area where the target is located, it is unsure which objects exactly are contained in their scope, especially in very crowded or vast domains. Therefore it might be that the pointing gesture rules out distractors other than the locative expression. In such cases pointing gestures and locative expressions lead to partly overspecified referring expressions. However, discovery of the exact scope of pointing gestures and the scope of locative expressions demands a detailed analysis (c.f., Kranstedt et al., 2005; Kranstedt et al., to appear), which cannot be captured from the data of Study II. The analysis presented here therefore assumes the scope of a pointing gesture and the scope of a locative expression to coincide when used to identify the same object, in order to discover generic relations between pointing gestures and locative expressions.

As seen in Section 5.4.3, in about half of the cases in which an imprecise pointing gesture is produced, the gesture includes some kind of movement during the stroke of the pointing gesture (circular, horizontal or vertical). Closer inspection reveals that the type of movement in these gestures is closely related to the shape of the target. For instance, to indicate Russia, subjects tend to produce a horizontal movement while pointing, whereas subjects employed vertical movement to indicate Chile or Japan and circular movement to refer to China or the United States. In these cases properties that address the shape of the target for example, *uitgestrekt* (vast) or *langgerekt* (long stretched) or *langwerpig* (long shaped) can be used as a linguistic alternative. Still, a vertical movement in the stroke might indicate *long shaped* but does not necessarily have exactly the same meaning (c.f., Kopp et al., 2004 on image description features) The two might also be taken complementarily, as in 'long shaped on a vertical axis'. Similarly, a dynamic pointing gesture can be interpreted as an area indication that might not completely overlap with the denotation of a property like *vast*. There is a difference in the way the descriptions are overspecified. For the sake of simplicity, in the analysis below, dynamic pointing gestures that appear together with the property *shape* to indicate the same object are analyzed as a kind of multimodal overspecification.

In order to provide a better insight in the manifestation of these two types of multimodal overspecification, in the following analysis the total number of pointing gestures and locative expressions and the number of dynamic pointing gestures and the occurrences of the property *shape* are taken into account as well. The 600 referring expressions that resulted from Study II are analyzed with respect to overspecification by a measurement of the following features per target description:

- *Overspecification by pointing gestures that co-occur with locative expressions* This feature counts the number of locative expressions that join with a pointing gesture to indicate the same object in a target reference. These are considered multimodal overspecifications.

- *Number of locative expressions* Per target description, all locative expressions are counted, locative expressions addressing the target as well as locative expressions addressing relata.

- *Total number of pointing gestures* Per target description, pointing gestures directed towards the target and directed towards the relata are counted. This is the same as in Section 5.4.3.

- *Overspecification by dynamic pointing gestures that co-occur with the property shape* If a dynamic pointing gesture directed at an object co-occurs with the mentioning of the *shape* property of the same object in the linguistic description, the referring expression is considered multimodally overspecified.

- *Occurrences of shape* Per target description, all occurrences of *shape* are counted, no matter if applicable to the target or its relata.

- *Number of dynamic pointing gestures* Per target description, all dynamic pointing gestures are counted, gestures directed towards the target as well as gestures directed towards relata. This is the same as in Section 5.4.3.

In this study two groups of subjects had to identify countries on a world map; the subjects in one group were located close to the map and those in the other group were located further away from the map. Both groups had to identify 15 small and 15 large countries. Below the small countries are referred to as **difficult** to identify, whereas the large countries are assumed to be **easy** to identify. As seen in Section 5.4, linguistic referring expressions together with imprecise pointing gestures are used mainly in the far condition, whereas precise pointing gestures are used in the near condition. When looking at the multimodal referring expressions in the far condition, a country like Suriname for instance is indicated as 'Suriname is dat kleine gele stukje ten Noorden van Brazilie' (Suriname is the little yellow part on the north side of Brazil) together with an imprecise pointing gesture. In such a case the joint appearance of the locative expression *ten Noorden van Brazilie* (on the north side of Brazil) and the pointing gesture are considered a type of multimodal overspecification, where the pointing gesture and the locative expression denote the location of the target. An example of a referring expression to indicate Chile is: 'Chili is die rare hele smalle lange paarse strook eh links onderaan in Zuid Amerika' (Chile is the weird very thin long purple strip eh left below in South America). This description is then produced together with a pointing gesture with vertical movement during the stroke. In the following analysis, the *shape* properties *smalle* (thin) and *lange* (long) together with the vertical movement in the pointing gesture are determined as a kind of overspecification.[2]

			DISTANCE	
			NEAR	FAR
SIZE	EASY	**locative expressions & pointing**	.08(.12)	1.61(.59)
		locative expressions	.16(.17)	3.06(1.17)
		pointing	1.00(.00)	1.32(.22)
		shape & dynamic pointing	.00(.00)	.13(.09)
		dynamic pointing	.17(.31)	.70(.40)
		shape	.00(.00)	.18(.15)
	DIFFICULT	**locative expressions & pointing**	.13(.17)	2.35(.74)
		locative expressions	.21(.31)	5.60(1.79)
		pointing	1.02(.04)	1.86(.59)
		shape & dynamic pointing	.07(.02)	.20(.11)
		dynamic pointing	.15(.26)	.73(.26)
		shape	.09(.14)	.32(.17)

Table 6.2: Descriptive means and standard deviations of the occurrences of the *shape* property, and the number of *locative expressions*, the total number of *pointing* gestures, the number of *dynamic pointing* gestures and two types of multimodal overspecification: (1) The co-occurrences of *locative expressions & pointing*; and (2) The co-occurrences of *shape & dynamic pointing* as contained in the multimodal referring expressions resulting from Study II where two groups of subjects, one in the *near* and one in the *far* condition, identified *easy* and *difficult* objects. Standard deviations are given between brackets.

In Table 6.2 the descriptive means and standard deviations for the two kinds of multimodal overspecification are presented. For the sake of illustration the occurrences of the property *shape*, the total number of locative expressions, the number of dynamic pointing gestures and the total number of pointing gestures, the latter two of which are repeated from Table 5.7, are displayed as well. For all features, the between-subject effects are significant, meaning that they are used considerably more in the far condition than in the near condition. Multimodal overspecification caused by the joint appearance of locative expressions and pointing gestures occurs almost twice as often in references to difficult objects than in references to easy objects. In both cases a pointing gesture is always accompanied by at least one locative expression. The significant within-subjects effect for the co-occurrence of locative expressions and pointing gestures $(F(1,18) = 23.28, p < .01)$ can be explained by the fact that subjects tend to use this combination more often in reference to difficult objects than in reference

[2] Although the noun *strip* also denotes a shape aspect of the target, it is not considered in the analysis. Here *strip* is taken as *type*.

to easy objects. This is supported by the within-subjects effect of the number of locative expressions ($F(1,18) = 50.30, p < .01$). Moreover, there is an interaction between the target and distance measures for this kind of multimodal overspecification ($F(1,18) = 18.75, p < .01$), which signals a stronger effect of target in the far condition than in the near condition. This is also the case for the locative referring expressions themselves ($F(1,18) = 8.28, p < .01$). Multimodal overspecification caused by the co-occurrence of dynamic pointing gestures and *shape*, appears in about a third of the descriptions that contain the property *shape*. The analysis of the property *shape* results in a significant within-subjects effect as well ($F(1,18) = 14.47, p < .01$), which displays the fact that *shape* is more often used in reference to difficult objects than in reference to easy objects. The interaction of *shape* implies that the effect of target is stronger in the far than in the near condition ($F(1,18) = 8.28, p < .01$). Finally, the number of dynamic pointing gestures is more or less the same for easy and difficult targets. Nevertheless, there is an interaction in the number of dynamic pointing gestures ($F(1,18) = 11.25, p < .01$) signalling a stronger effect of target in the far condition compared to the near condition.

This analysis confirms the results of the analysis performed on the data for Study I, in that speakers use more redundant information in identifying objects that are located further away than in references to objects that are located close by. In terms of multimodal overspecification, speakers produce more often combinations of pointing gestures and locative expressions when the target is located far away than when the target is located at a clos distance. When the target is difficult to identify, this kind of redundancy occurs more often than in cases where the target is easy to refer to. As in the analysis of Study I, this can be regarded as a way in which the speaker tries to reduce uncertainty. The speaker adds extra information, i.e., a pointing gesture or a locative expression, in cases where the target is far away and difficult to describe. Less evidence is found for multimodal overspecification in terms of dynamic pointing gestures that occur together with the property *shape* in the linguistic referring expression. Note however, that not all countries have a very prominent shape. Still, in most cases in which speakers use *shape*, they produce a dynamic pointing gesture as well. Especially in reference to targets that are difficult to distinguish, shape is used more often.

6.1.3 Discussion

From the research on unimodal overspecification summarized in Section 6.1, it can be concluded that there are several factors that play a role in the reasons why speakers overspecify their referring expressions:

- Discourse goals: in cases where the hearer has to be instructed to perform a task on only a single occasion, instructions contain less overspecified descriptions than in cases where a hearer has to actually learn a task in order to perform it on more than one occasion (Maes et al., 2004).

- Task importance: overspecification is especially used in situations in which the speaker wants to stress a commitment, persuade the hearer, or focus on a change in the task constraints (Jordan, 2002).

- Modes of communication: different modes of communication involve different types of feedback. For instance when the discourse participants cannot see or hear each other, writers tend to produce highly overspecified referring expressions (Maes et al., 2004).

- Situational conditions: in a discourse in which participants have no access to the same object domain, for example when the participants discuss functions of two different alarm clocks, writers have to spend more effort in order to refer to an object (Maes et al., 2004).

In general these factors can be explained by an uncertainty on the side of the speaker about the hearer being able to interpret the referring expression. Thus, in the automatic generation of referring expressions these factors can be used to indicate the algorithm's estimation of the probability that the user might misunderstand a particular referring expression (c.f., the principle of distant responsibility Clark and Wilkes-Gibbs, 1986) and thereby to determine the degree of overspecification of the referring expression to be generated.

When the target is a salient object in the domain, because of its properties or because it is located in the focus of attention, the speaker can be more confident in identifying the target and generate a less overspecified or minimal description. In contrast, in distinguishing a target that is not salient the speaker might be relatively uncertain and produce a highly overspecified description. In the case that the distinguishing properties of a target are not easily perceived, the speaker includes more properties than necessary (Mangold and Pobel, 1988). These redundant properties increase the certainty of the speaker about the likelihood of a correct interpretation on the side of the hearer. From Section 6.1 three indications can be derived about the kind of properties that are selected:

- Properties that are easily perceived are generally faster produced and interpreted than object properties that are not so easy to discover (Pechmann, 1989).

- The kinds of properties selected to indicate the target, remarkably, are all of a perceptual nature as opposed to a functional one. Functional properties of objects are hardly used, not even in descriptions produced for a task that is focussed on operating a functional device (Maes et al., 2004).

- When measuring identification times on the part of the hearer, the inclusion of locative expressions is very advantageous (Arts, 2004). The production experiments performed by Arts suggest that speakers are aware of this fact and therefore include locative expressions as an extra effort to reduce identification time.

Multimodal overspecification, as it is analyzed in Section 6.2 with respect to the data resulting from Study I and Study II, occurs especially when the target is located further away from the speaker. Study I shows that speakers use superfluous locative expressions to identify objects that are difficult to describe, whereas more properties are used for easily identifiable targets. This is in line with the observation by Mangold and Pobel (1988) that objects which do not differ from their distractors by easily perceivable properties, are described in terms of highly overspecified descriptions compared to objects that have distinguishing properties that are more prominent. The speaker's distant responsibility as defined by Clark and Wilkes-Gibbs (1986) fits well within this view. The analysis of the data resulting from Study II presents positive evidence for multimodal overspecification in terms of co-occurrence of pointing gestures and locative expressions. When referring to objects that are located at a distance, speakers often use pointing gestures and locative expressions together. In reference to difficult objects, speakers use up to two pointing gestures that overlap with at least half of the locative expressions in the same reference. When the target is large, speakers still produce linguistic descriptions that contain one, and in some cases two, locative expressions accompanied by pointing gestures. This can partly be explained by the fact that the scope of both pointing gestures and locative expressions can be vague. As noted above, it is difficult to define which objects are exactly contained in the scope of an imprecise pointing gesture. The interpretation of locative expressions can be similarly fuzzy. For instance, in cases like 'in South America', which reduces the number of distractors, but does not eliminate all. Moreover, the speaker may be uncertain if the hearer knows where to look for South America in the first place. For this reason speakers may employ extra locative expressions and pointing gestures to increase their certainty about the hearer's understanding of the reference. Accordingly, pointing gestures that are combined with locative expressions do not lead to overspecified descriptions in all cases, they might also add to the precision of the reference. Although in Study II the task and the domain might have invoked the use of locative expressions, the within- and between-subject effects, as well as the interactions in the use of locative expressions in the analyzed data, stresses their importance, which is also claimed by Arts (2004) and Maes et al. (2004). Less support is found for the other type of multimodal overspecification, which is based on the co-occurrence of dynamic pointing gestures and the *shape* property. Still, speakers use *shape* significantly more often in reference to difficult targets. But of course, all countries in the domain were not equally identifiable by there shape; some had a prominent shape like Chile and others had a shape similar to neighboring objects, like for example Iraq. So, when it can be concluded that *shape* is used mainly to identify objects that have a shape that stands out (c.f., Beun and Cremers, 1998; Arts, 2004), in about two thirds of these cases, a dynamic pointing gesture is produced as well. Note also that, as discussed above, Arts (2004) provides evidence for the importance of *shape* which facilitates identification as an absolute property on its own, but also in combination with locative expressions. Accordingly the results of the analysis on multimodal overspecification endorse the following guidelines for automatic generation of multimodal referring expressions:

- Include more locative expressions if a distinguishing property of the target or the target itself is not easily perceived, use absolute properties otherwise;
- Include locative expressions in combination with imprecise pointing gestures;
- In the case that the target has a distinguishing or prominent shape and a pointing gesture is generated, include *shape* together with a dynamic pointing gesture that conforms with the shape of the target.

In the next section, the multimodal algorithm is adapted to the findings discussed in this section.

6.2 Automatic Generation of Overspecification

Most GRE algorithms, including the algorithm proposed in Chapter 4, do not generate overspecified referring expressions. In contrast to multimodal referring expressions produced in human generation, the linguistic descriptions that the graph-based algorithm (Section 4.3) generates, together with (im)precise pointing gestures, are always minimal. In this section a variant of the multimodal graph-based algorithm is proposed, which results in the possible generation of overspecified referring expressions based on strategies observed in human communication. To determine the degree and the kind of overspecification of the referring expression to be generated, the algorithm makes use of a certainty score. Every referring graph, i.e., a graph representing the target, receives a certainty score. Intuitively, the certainty score of a referring graph represents the speaker's estimate of the probability that the resulting expression will be understood by the hearer. This probability may depend on, for instance the perceptibility of the property. The context determines what an acceptable likelihood of misunderstanding is for a particular task. This is captured using a Certainty Threshold. When a distinguishing referring graph does not satisfy the Certainty Threshold, a more extensive graph is required to increase the certainty score. Via the combination of costs and certainty scores, the algorithm can generate the whole range of referring expressions from minimal ones to highly overspecified ones. The certainty score and the Certainty Threshold are explained in Section 6.2.1. In Section 6.2.2 the distribution of the certainty scores over the various edges is addressed. Section 6.2.3 presents a revised version of the graph-based algorithm, which is illustrated with a worked example in Section 6.2.4.

6.2.1 Certainty Score

To be able to decide if the degree of overspecification of a referring graph is satisfactory or not, the algorithm uses a **certainty score**. The certainty score is defined as a probability, *Pr* in the interval [0,1], which indicates the speaker's calculation of the likelihood of misunderstanding by the hearer. The value 0 expresses that the speaker thinks that the hearer will certainly not be able to interpret the referring expression, while the value 1 reflects assumed certainty that the hearer can correctly interpret the generated referring expression. To facilitate calculations $-\log_2 Pr$ is used, and thus, a low positive numerical value indicates a low certainty, while a higher numerical value indicates a higher certainty. In the generation process, every generated graph receives a certainty score which is the summation of the certainty scores that relate to the properties, relations and pointing gestures contained in the graph, i.e., every edge, *e*, in a graph has a certainty score. Formally, let $G = \langle V, E \rangle$ be a labeled directed graph then:

$$\textbf{CertaintyScore}(G) = \sum_{e \in E_G} \textbf{CertaintyScore}(e)$$

To determine if a graph is adequate to refer to the target, the certainty score of the graph is compared to the Certainty Threshold. The **Certainty Threshold** is a positive numerical value, which depends on aspects that relate to contextual factors such as task importance, the principle of distant responsibility, or the kind of objects in the domain, as discussed in Section 6.1. Thus, the more important the task the higher the Certainty Threshold. A certainty score below the Certainty Threshold, represents uncertainty whether the hearer can interpret the referring expression that can be realized from that graph in a given context. In that case the algorithm will look for a graph with more edges, or a graph with edges that have higher certainty scores, thereby increasing the degree of overspecification in order to reach the required confidence level.

6.2.2 Choice of Edges

The Certainty of Properties

In general the certainty score of a graph increases when a property or a relation is appended to the graph. As seen in Section 6.1.1, additional linguistic information strengthens the speaker's confidence in the hearer's understanding. Specifically, perceptible properties in contrast to functional properties have a positive influence on the certainty score of a referring graph. Also, locative expressions provide the speaker with more confidence. As seen in Section 6.1, properties that are easily perceived are generally produced faster and interpreted faster than properties that are not so easy to discover (Pechmann, 1989). This can be illustrated by the intuition that the certainty score of the property *type* probably depends on the variability of the objects in the domain, i.e., if all objects in the domain are blocks, *type* does not increase the certainty score. From this it seems best to conclude that the exact influence of the properties on the speaker's certainty of a referring expression is domain dependent. Additionally, the determination of the certainty scores of the properties might differentiate between scores within one property. For instance, *blue* can result in more confidence than *cyan*, if *blue* is considered more common than *cyan* (c.f., Dale and Reiter (1995) and Krahmer and Theune (2002) on basic level values).

Given the domain dependency of the influence of properties, the decision was made to determine that properties have a certainty score that is weighed against the number of objects in the domain and against the other objects in the domain that have this property (c.f., Dale (1989) on discriminatory power). For now it is assumed that the speaker's estimation of the likelihood, *Pr*, of misunderstanding a linguistic property, *p*, is approximated by:

$$\mathrm{P}r = 1 - (\tfrac{N-n}{N-1})$$

Where N is the number of objects in the domain and n is the number of objects that have linguistic property p.[3] This captures the intuition that the more often a particular property occurs in a domain, the less certain the speaker can be that including this property in a target description will help the hearer to identify the target. Notice that this probability is 0 (no chance of misunderstanding) for unique properties. The certainty score of an edge expressing a linguistic property p is thus defined below as 1 minus the discriminatory power of $p + 10^{-1}$, where the 10^{-1} is added to avoid the certainty score of properties that occur only once in the whole domain remaining undefined.

$$\textbf{CertaintyScore}(p) = -\log_2([1 - (\tfrac{N-n}{N-1})] + 10^{-1})$$

The Certainty of Pointing

Like additional linguistic edges, pointing edges increase confidence. Intuitively, the inclusion of a precise pointing gesture takes away all uncertainty about the identity of the target, which causes the certainty score to reach 'full' confidence. By contrast, imprecise pointing gestures have a lower certainty score, because the scope of such gestures may include not only the target but also other objects. Empirical evidence presented in Section 6.1.2 supports this intuition. Although in

[3] More precisely, n is the number of vertices from which an edge expressing p departs.

Study II pointing gestures were not obligatory, they were always used to identify the target. The fact that in this study multimodal overspecified referring expressions occur more often when the distance to the target is large (i.e., when more objects are located in the scope of the gesture), reflects less confidence on the part of the speaker that the hearer can interpret the gesture correctly. Hence the more objects located in the scope of the gesture, the lower the certainty score of the edge that represents the gesture. Given the variability in the precision of pointing gestures, the decision was made to determine that gestures have a certainty score that is weighed against the number of objects in the scope of the gesture as well as against the other objects in the domain. So it is assumed that the speaker's estimation of the likelihood, Pr, of misunderstanding a pointing gesture, g, is approximated by:

$$\mathrm{P}r = 1 - (\tfrac{N-n}{N-1})$$

Where N is the number of objects in the domain and n is the number of objects in the scope of the gesture. This captures the intuition that the more objects that are located in the scope of a gesture, the less certain the speaker can be that the gesture will help the hearer to identify the target. Notice that this probability is 0 (no chance of misunderstanding) for precise pointing gestures. The certainty score of a pointing gesture g is thus defined below as 1 minus the discriminatory power of the pointing gesture + 10^{-10}, where the 10^{-10} is added to obtain a very small positive number that indicates the certainty score of a precise pointing gesture. A smaller constant is used for pointing edges than for linguistic edges (10^{-10} vs. 10^{-1}) to account for the intuition that a precise pointing edge is more 'certain' than a unique linguistic property.

$$\mathbf{CertaintyScore}(g) = -\log_2([1 - (\tfrac{N-n}{N-1})] + 10^{-10})$$

The function to compute the certainty scores can be illustrated as follows: When a pointing gesture is very precise (i.e., the pointing finger is touching the target) then the scope of the gesture contains only the target, which supposedly cannot result in any confusion on the part of the hearer. The certainty score of a precise pointing gesture is defined as $-\log_2$(1 - (N - 1/N - 1) + 10^{-10}) = 33.22 (i.e., maximal confidence). In the case that there are more objects located in the scope of the gesture, the certainty score of the gesture becomes lower. Consider the scope of an imprecise pointing gesture containing three objects, where the domain contains six objects. The certainty score of the pointing gesture is then $-\log_2$(1 - (6 - 3/6 - 1)) = 1.32, which is considerably lower than the certainty score of a precise gesture.

6.2.3 Sketch of the Algorithm

In this section a variant of the multimodal algorithm described in Section 4.4 is presented, which is able to generate overspecified multimodal referring expressions. In Figure 6.3 the pseudocode of the algorithm's main function *GenerateReferringExpression* and the subgraph construction function *FindGraph* are displayed. In the multimodal graph-based algorithm, the function *GenerateReferringExpression* constructs a multimodal domain graph that represents the domain of conversation as a labeled directed graph. The objects in the domain graph are defined as the vertices (or nodes) in the graph. The properties, relations and pointing gestures that can be used to identify the objects in the domain are represented as edges (or arcs). Cost functions that assign weights to the edges are used to determine their order of preference in selection. Correspondingly, the decision to point is based on a trade-off between the costs of pointing edges and the costs of linguistic edges. To generate a multimodal referring expression, the function *FindGraph* searches for the cheapest subgraph (i.e., a **referring graph** that uniquely represents the target in the domain graph). In this section the function *FindGraph* is adapted so that it returns the cheapest graph that satisfies the Certainty Threshold.

The core of the algorithm is explained in Section 4.4; here only the revisions and their consequences are addressed. In Figure 6.3, the function *GenerateReferringExpression*, line (1), takes as input the target object (v), the domain graph (G) and the Certainty Threshold (T). The function constructs a multimodal domain graph (M), and invokes the function *FindGraph* in line (2). *FindGraph*, line (3), takes as input the target (v), the best graph so far (initially *Bestgraph* $= \bot$), the graph under construction (H), the multimodal graph (M) and the Certainty Threshold (T). *FindGraph* contains two conditions on the fulfilment of which it returns the best graph, and a recursive step. In the recursion, the graph under construction, H, is expanded with the edges with which the target, v, can be described. In the condition in line (4) three checks are performed: (1) It is checked whether *BestGraph* is not $\bot$ (i.e., a solution has been found); (2) It is checked whether *BestGraph* is cheaper than H (i.e., the solution found earlier is cheaper than the graph under construction); and (3) It is checked whether the certainty score of *Bestgraph* is higher than or equal than the Certainty Threshold T. For the latter check in line (5), the function *CertaintyScore* sums the Certainty scores of all the edges in *BestGraph* and compares it to the threshold T. By performing these three checks, the algorithm balances the *BestGraph* between, on the one hand the cost of the graph that should be as low as possible, and on the other hand the certainty score that should be equal to or higher than the Certainty Threshold. The condition is checked for every relevant subgraph H of M constructed in the loop started in line (7). In this loop the algorithm recursively tries to extend H by adding **adjacent edges** e, that is edges which start in v or possibly in any of the other vertices added later on to H, the graph under construction. The resulting graph of the recursive call to *FindGraph* is assigned to the variable I. Graph I is defined to be the subsequent *BestGraph* only if it fulfills three conditions: (1) I should not be $\perp$; (2) I should be cheaper than the current *BestGraph*; and (3) The certainty score of I should be higher than or equal to the Certainty Threshold, T. The function *FindGraph* repeats these steps until all relevant subgraphs have been tried. The cheapest distinguishing graph that satisfies the Certainty Threshold is returned to *GenerateReferringExpression* in line (2), where it is stored in the variable *BestGraph* and successively returned by the function *GenerateReferringExpression*.

```
(1)  GenerateReferringExpression($v, G, T$)

       construct(v, $F_v$, G)
       $M := F_v \cup G$
       $BestGraph := \perp$
       $H := \langle \{v\}, \emptyset \rangle$
(2)    $BestGraph :=$ FindGraph($v, BestGraph, H, M, T$)
       return $BestGraph$

(3)  FindGraph($v, BestGraph, H, M, T$)

(4)    if $BestGraph \neq \perp$ and
         Cost($BestGraph$) $\leq$ Cost(H) and
(5)      CertaintyScore($BestGraph$) $\geq T$ then
         return $BestGraph$
       end if
       $C := \{n \mid n \in V_M \wedge$ MatchGraphs$(v, H, n, M)\}$
       if $C = \{v\}$ then
         return $H$
       end if
(6)    for each adjacent edge $e$ do
         $I :=$ FindGraph($v, BestGraph, H + e, M$)
           if $BestGraph = \perp$ or
             Cost($I$) $\leq$ Cost($BestGraph$) and
(7)          CertaintyScore($I$) $\geq T$ then
             $BestGraph := I$
           end if
       end foreach
       return $BestGraph$
```

Figure 6.3: Pseudocode of the revised algorithm's main function *GenerateReferringExpression* and the subgraph construction function *FindGraph*.

6.2.4 Worked Example

In this section the algorithm presented in Section 6.2.3 is illustrated with a simple worked example in the block domain presented as Example Domain I in Figure 6.1. For the current example the target is d_3. Table 6.3 presents the costs and certainty scores of the edges that can be used to describe d_3. Currently, the algorithm selects edges to describe a target on the basis of cost functions inspired by the notion of preferred attributes as proposed by Dale and Reiter (1995) (see Section 3.3.2). The relevant properties are ordered according to the preference that human speakers and hearers have when discussing objects in a certain domain. Hence, for the objects in the Example Domain presented in Figure 6.1 the costs of edges that represent absolute properties are determined to be lower than the costs of the edges that represent relative properties, while edges representing spatial relations are even more expensive. The certainty scores are determined by applying the function presented in Section 6.2.2. In Example Domain I, which consists of only square blocks, *type* and *shape* have a very low influence on the certainty score (0.14), whereas the spatial relation *right-of(d_2)* and the edge *large* have a high influence on the certainty score (3.32). The costs of the pointing gestures are defined by the definition presented in Section 4.3.5. For the sake of simplicity, only two pointing gestures directed at d_3 are considered: a precise pointing gesture, P, which is expensive (it costs 8), but which is certain (33.22), and a very imprecise pointing gesture, VIP, which is cheap (it costs 2) but is less certain (0.74).

edges	**costs**	**certainty scores**		
block	0	$\log_2(1 - (3 - 3 / 3 - 1) + 10^{-1})$	=	0.14
black	1	$\log_2(1 - (3 - 2 / 3 - 1) + 10^{-1})$	=	0.74
square	1.5	$\log_2(1 - (3 - 3 / 3 - 1) + 10^{-1})$	=	0.14
large	2	$\log_2(1 - (3 - 1 / 3 - 1) + 10^{-1})$	=	3.32
right-of(d_2)	2.5	$\log_2(1 - (3 - 1 / 3 - 1) + 10^{-1})$	=	3.32
P	8	$\log_2(1 - (3 - 1 / 3 - 1) + 10^{-10})$	=	33.22
VIP	2	$\log_2(1 - (3 - 1 / 3 - 1) + 10^{-10})$	=	0.74

Table 6.3: Costs and certainty scores of the edges that can be used to describe d_3 in Figure 6.1.

In the following discussion, the effects of the certainty score on the description of object d_3 in Figure 6.1 are demonstrated for three cases for which the Certainty Threshold is varied. In the block domain of Figure 6.1, in which there is only a small number of objects that have distinguishing properties, the Certainty Threshold can be set between 0 and 15. In the first case the Certainty Threshold is extremely low, i.e., the objects in the domain are relatively accessible, for instance as a result of the fact that the object domain has been talked about already. The second case shows what happens if the Certainty Threshold has a moderate value. Finally, the third example sketches the outcome of the algorithm if a very high Certainty Threshold is applied, i.e., it is very important that the algorithm makes sure that the hearer can resolve the target, for example because of a high task importance.

In the first case, where accessibility is high, suppose that the Certainty Threshold is 0. To describe d_3 the algorithm invokes the function *FindGraph*, which returns the cheapest graph that satisfies the Certainty Threshold. This graph contains the edges *large* and *block*, of which the latter is included for free without any effect on the certainty score. The certainty score of this graph is $(0.14 + 3.32) = 3.46$ which means that the Certainty Threshold is met with a minimal graph. Accordingly, the algorithm generates the graph depicted as H_1 in Figure 6.4 at the of cost 2. H_1 can be realized as 'the large block'.

In the second case, the Certainty Threshold has a moderate value of 5. The cheapest graph that *FindGraph* can produce to describe d_3 contains the edges *block* and *large* for cost $(0 + 2) = 2$. This minimal graph does not satisfy the Certainty Threshold $(3.46 \leq 5)$. Consequently, the algorithm searches for a more expensive graph which meets the Certainty Threshold at the lowest cost. The graph that contains the edges *block*, *black* and *large* and the graph with the edges *block*, *large* and VIP both have a certainty score of $(0.14 + 0.74 + 3.32) = 4.20$, which is still too low. Also the graph containing the edges *block*, *black*, *large* and VIP, which has a certainty score of $(0.14 + 0.74 + 3.32 + 0.74) = 4.94$ for cost $(0 + 1 + 2 + 2) = 5$ does not meet the Certainty Threshold. The graph containing the edges *block*, *large*, *right-of*(d_2), with a certainty score of $(0.14 + 3.32 + 3.32) = 6.78$ does meet the Certainty Threshold at cost $(0 + 2 + 2.5) = 4.5$. This graph, depicted as H_2 in Figure 6.4, is returned by the algorithm and can be realized as 'the large block on the right', where the spatial relation *on the right*, with the domain itself as an implicit relatum subsumes the more specific relation *right-of*(d_2) (c.f., Section 3.4.4 for a discussion on subsumption).

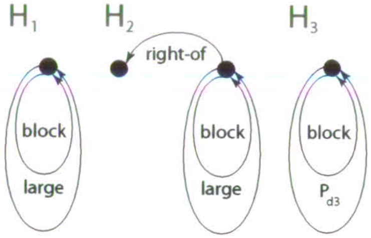


Figure 6.4: Referring graphs for object d_3 in Figure 6.1.

In the third case the Certainty Threshold is set very high, at 15. To generate an adequate description for d_3 which meets the Certainty Threshold, the algorithm selects the graph with a precise pointing gesture, P and the property *block*, that is associated with a high certainty $(0.14 + 33.22) = 33.36$ for cost $(0 + 8) = 8$. Correspondingly, the algorithm generates this graph, depicted as H_3 in Figure 6.4, that can be realized as 'this block' with a precise pointing gesture directed at d_3.

6.3 Human versus Automatic Generation

In this section, two more elaborate examples are presented, which show the workings of the revised algorithm proposed in Section 6.2.3 on the example domains used in the studies described in Chapter 5. In this way the output of the algorithm can be compared to the multimodal referring expressions produced by human speakers. The algorithm is illustrated with three worked examples in which the algorithm generates referring expressions similar to the ones observed in the production experiments presented in Sections 5.3 and 5.4. In Section 6.3.1, an example in the domain with geometrical objects, as presented in Study I, is employed to sketch the workings of the algorithm in an unimodal situation. In Section 6.3.2, two examples in the world map domain, as presented in Study II, are considered for the generation of multimodal referring expressions.

6.3.1 Unimodal Overspecification

Example 1: The Triangle
As an example of the generation of overspecified unimodal referring expressions, the geometrical object domain as presented in Figure 6.5 (a) is taken. In Study I, this object domain was one of the stimuli presented on a computer screen, on which subjects had to identify the triangle by including at least a pointing gesture. In the far condition the subjects performed obligatory imprecise pointing gestures that directed the attention to the part of the screen in which the target was located. In this example, only the linguistic descriptions are taken into account. In Figure 6.6, the descriptions of the triangle in Figure 6.5, which were produced by the ten subjects in the far condition in Study I, are presented. All descriptions contain a reference to the triangular shape of the target. Moreover in eight of the ten descriptions *blue* is redundantly included. Only in descriptions (3) and (8) is the property *color* not verbalized. Description (3) is a minimal description and description (8) contains a redundant indication of the location of the target on the screen.

In order to generate referring expressions, the object domain in Figure 6.5 (a) is translated into the graph presented in Figure 6.5 (b). To generate a referring expression similar to the referring expression presented in Figure 6.6, the Certainty Threshold is set to 4 (a moderate value). The costs and certainty scores of the edges that can be used to describe the triangle in Figure 6.5 are presented in Table 6.4, where the costs are assumed to be the same as in the example discussed in

Section 6.2.4 and the certainty scores are determined with the definition presented in Section 6.2.2. In contrast to the domain applied for the worked example in Section 6.2.4, in this domain the property *shape* is informative, whereas *size* does not have an additional value. Only the most general relation, *right-of(screen)*, is defined.

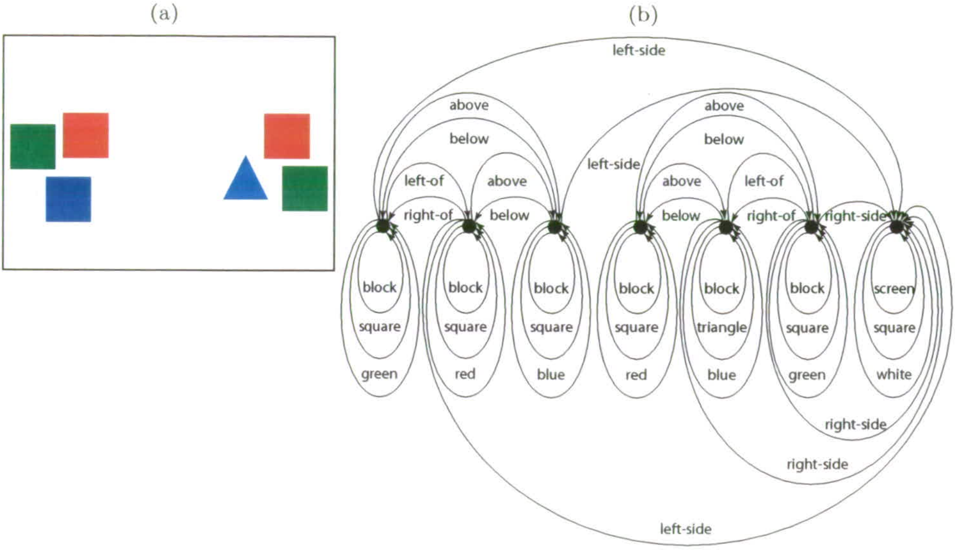


Figure 6.5: An example from the domain of geometrical objects taken from Study I (a) and (b) Scenegraph for the domain presented in (a).

(1)	een *blauwe* driehoek	a *blue* triangle
(2)	*blauwe* ehm driehoek	*blue* ehm triangle
(3)	de driehoek	the triangle
(4)	*blauwe* driehoek	*blue* triangle
(5)	*blauwe* driehoekje	*blue* triangle
(6)	*blauwe* driehoek	*blue* triangle
(7)	eh *links onder* t *blauwe* driehoek	eh *left below* the *blue* triangle
(8)	driehoek *rechts*	triangle *right*
(9)	driehoek *rechts blauw*	triangle *right blue*
(10)	*blauwe* driehoek	*blue* triangle

Figure 6.6: Descriptions of ten subjects in the far condition in Study I for the target triangle in Figure 6.5. The redundant properties and the redundant locations are displayed in italics.

	costs	certainty scores		
block	0	$\log_2(1 - (6 - 6 / 6 - 1) + 10^{-1})$	=	0.14
blue	1	$\log_2(1 - (6 - 2 / 6 - 1) + 10^{-1})$	=	1.74
triangle	1.5	$\log_2(1 - (6 - 1 / 6 - 1) + 10^{-1})$	=	3.32
size	2	$\log_2(1 - (6 - 6 / 6 - 1) + 10^{-1})$	=	0.14
right-of(screen)	2.5	$\log_2(1 - (6 - 3 / 6 - 1) + 10^{-1})$	=	1

Table 6.4: Costs and certainty scores of the edges that can be used to describe the triangle in Figure 6.5 (a).

The costs and certainty scores that can be used to describe the triangle in Figure 6.5 are presented in Table 6.4. To generate a referring expression for the triangle, the unimodal version of the algorithm invokes the function *Findgraph* that returns the cheapest distinguishing referring graph with a certainty score that satisfies the Certainty Threshold. The cheapest distinguishing graph is the graph that contains the edges *block* and *triangle* for cost (0 + 1.5) = 1.5. is depicted as H_1 in Figure 6.7. This minimal graph, has a certainty score of (0.14 + 3.32) = 3.46, however, which is lower than the Certainty Threshold ($3.46 \leq 4$). In order to meet the Certainty Threshold, the algorithm has to find a more expensive graph. By adding the property *blue* the threshold is reached at the lowest cost. Adding a spatial relation would also satisfy the Certainty Threshold (0.14 + 3.32 + 1) = 4.46, but is more expensive (0 + 1.5 + 2.5 = 4). Thus, the algorithm returns the graph that contains the edges *block*, *triangle* and *blue* for the cost (0 + 1.5 + 1) = 2.5 and certainty score (0.14 + 3.32 + 1.74) = 5.20. This graph is depicted as H_2 in Figure 6.7 and can be realized as 'the blue triangle', or 'the blue triangular block'.[4]

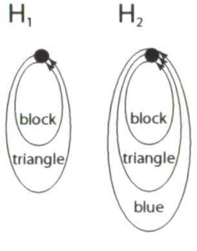


Figure 6.7: Referring graphs as generated for the triangle presented in Figure 6.5.

[4] The inclusion of 'block' is a consequence of the notion, due to Dale and Reiter (1995), that *type* is the most preferred attribute and should always be included. An alternative would be to associate a small cost with *type*, so that it will always be included when it is informative.

Table 6.5 illustrates the performance of the algorithm, presenting the descriptions from Figure 6.6 in comparison with the output of the algorithm, graph H_2 from Figure 6.7. The table displays two criteria on which the performance is measured: (1) *exact match* i.e., all edges in the returned graph are represented in the description and no other information is included in the description; and (2) *lean match* i.e., all edges in the returned graph are represented in the description and maybe other information is represented as well. Value 1 means there is a match, value 0 means no match. From Table 6.5 it can be read that in six of the ten cases the algorithm exactly matches the descriptions produced in the experiment i.e., they solely contain the properties *type* and *color*. In four cases the match is not exact: description (3) and (8) omit the property *blue*, whereas descriptions (7), (8) and (9) include extra locative information. In eight of the ten cases the algorithm reaches a lean match, where all information represented by the returned graph is contained in the descriptions. In only two of the cases a lean match is not found: description (3) and (8) do not include the property *blue*.

		exact match	**lean match**
(1)	a *blue* triangle	1	1
(2)	*blue* ehm triangle	1	1
(3)	the triangle	0	0
(4)	*blue* triangle	1	1
(5)	*blue* triangle	1	1
(6)	*blue* triangle	1	1
(7)	eh *left below* the *blue* triangle	0	1
(8)	triangle *right*	0	0
(9)	triangle *right blue*	0	1
(10)	*blue* triangle	1	1

Table 6.5: Evaluation of the output of the algorithm by comparing the output, graph H_2 depicted in Figure 6.7 to the ten descriptions as presented in Table 6.6, where *exact match* is defined as 1 in the case the edges in the graph are represented exactly in the description and 0 otherwise, and *lean match* is defined as 1 in the case that the output of the algorithm is present in the description but other properties may be included as well, and 0 otherwise.

6.3.2 Multimodal Overspecification

Example 2: Brazil

To illustrate the generation of multimodal referring expressions, in this section the algorithm is applied in the map domain of Study II. In Figure 6.8 the descriptions of Brazil by ten subjects in the far condition in Study II are presented. Except for descriptions (9) and (10) they all contain locative expressions. Five descriptions include South America as a relatum. Description (6) includes 'dat eiland' (that isle), which might be taken as a synonym for South America. The descriptions (1) and (5) also include other relata. The mentioning of Argentina in description (1) might be due to the dialogue context, in which Argentina was already discussed. The spatial relations verbalized in description (5) seem very exemplary compared to the other descriptions. Except for (5) and (6), all descriptions contain the properties *color* and *size*. Description (6), although not using the word 'groot' (large), expresses the size of Brazil by signalling that the country takes almost half of South America, which can be interpreted as Brazil being comparatively large. Description (5) includes the *color* of the target, but not the *size*; instead a lot of spatial relations are used for identification. In all referring expressions pointing gestures are included, half of them displaying a circular movement during the stroke of the gesture.

(1)	Brazilie is ehm het (VIP circle) *grote groene land boven Argentinie*	(1)	Brazil is ehm the (VIP circle) *large green country above Argentina*
(2)	ah Brazilie dat is (VIP) dat *grote groene land in Zuid Amerika*	(2)	ah Brazil that is (VIP) that *large green country in South America*
(3)	Brazilie is (VIP circle) dat *hele grote groene deel in Zuid Amerika* dus *aan de* (VIP) *bovenkant*	(3)	Brazil is (VIP circle) that *very large green part in South America* so *at the* (VIP) *upper part*
(4)	(VIP circle) dat *grote groene vlak rechts onder Zuid Amerika*	(4)	(VIP) that *large green area right below South America*
(5)	Brazilie is (VIP circle) *Zuid Amerika* eh het is eh zeg maar het *groene land* wat *onder ehm onder grenst onder grenst aan ehm Bolivia Paraguay* en *bovenaan ehm ondermeer Suriname*	(5)	Brazil is (VIP circle) *South America* eh it is eh let's say the *green country* that *below ehm below borders below ehm borders with Bolivia Paraguay on top ehm among others Suriname*

(6)	Brazilie ligt (VIP circle) daar ehm neemt bijna de helft in van t ehm dat eiland en is *groen*	(6)	Brazil lies (VIP circle) there ehm takes almost half of the ehm that isle and it is *green*
(7)	Brazilie ligt *in Zuid Amerika* (VIP circle) dat is dat *hele grote land* daar dat *groene* het *grootste land* (VIP circle) *van Zuid Amerika*	(7)	Brazil lies *in South America* (VIP circle) it is that *very large country* there the *green* the *biggest country* (VIP circle) *of South America*
(8)	Brazilie is (VIP) het *groene land* dat eh is een *heel groot land* dat eh *op ja ze noemen het continent Zuid Amerika*	(8)	Brazil is (VIP) the *green country* that eh is a *very large country* that *eh on yes they call the continent South America*
(9)	Brazilie je ziet daar (VIP) t *grote groene vlak*	(9)	Brazil, you see there (VIP) the *large green area*
(10)	ja dat is daar (VIP) dat *groene grote*	(10)	yes that is there (VIP) that *green large*

Figure 6.8: Multimodal descriptions for Brazil by ten subjects in the far condition in Study II. The properties and locative expressions are in italics. *VIP* = Very Imprecise Pointing gesture and *circle* = circular movement in the stroke of the gesture.

For the sake of illustration in the example the representation of the domain is limited to the map of South America as displayed in Figure 6.9. This map is considered as a part of the world map presented in Figure 5.4, which is used in Study II, where Brazil has a green color. The domain graph depicted in Figure 6.10 contains ten countries in South America and their properties and relations. The graph has been slightly simplified by limiting of the number of countries located in South America and by representing the relations between two vertices by one bidirectional edge, instead of two single directed edges. These simplifications do not influence the main points of this section.

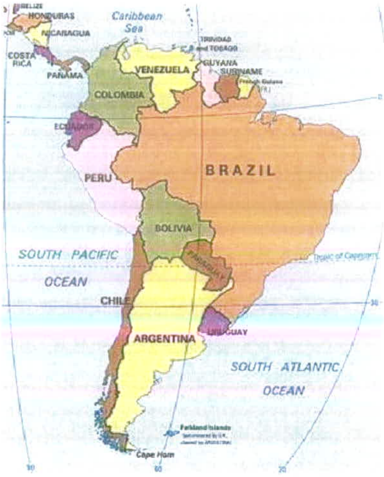


Figure 6.9: Political map of South America, which is considered as a part of the world map presented in Figure 5.4.

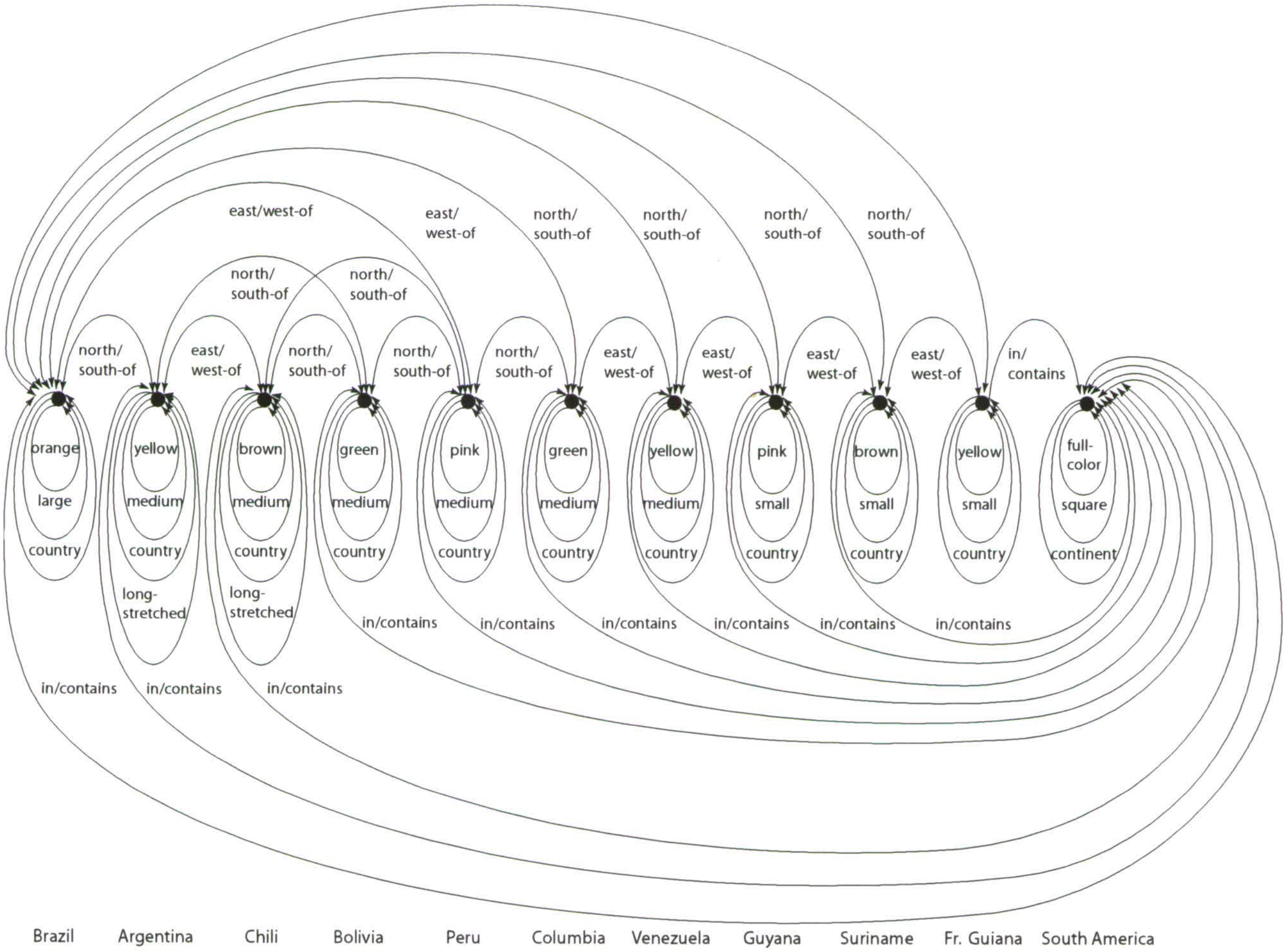


Figure 6.10: Domain graph for the domain presented in Figure 6.9.

Table 6.6 presents the costs and certainty scores of the edges that can be used to identify Brazil. The costs are defined as before. According to the World Factbook[5] of 264 countries. On the world map presented in Figure 5.4, 15 countries are considered relatively large and 36 are colored green. For this example, the certainty scores of two spatial relations are defined: a general one, *in-south-america*, and a more specific one *north-of-argentina*, where Argentina is considered the most salient country in South America besides Brazil, because it is relatively large and has a bright yellow color. Only two types of pointing gestures are included, precise pointing gestures (*P*) and very imprecise pointing gestures (*VIP*). Because South America happens to have a more or less vertical orientation, the pointing gestures are restricted to the north-south direction. An imprecise pointing gesture directed at Brazil is assumed to direct the attention to the north of South America, which excludes the distractors that are located on the southern part of the map. For each target, the algorithm constructs a gesture graph, which is unified with the domain graph to obtain a multimodal domain graph. As an illustration, a gesture graph for Brazil is depicted in Figure 6.11. A precise pointing gesture indicates Brazil uniquely. As shown in the gesture graph presented in Figure 6.11, the scope of an imprecise pointing gesture directed at Brazil includes 9 countries (i.e., the northern half of South America) and therefore has a certainty score of 2.73. As explained in Section 6.2.2 a precise pointing gesture has a much higher certainty score of 33.22.

	costs	**certainty scores**		
country	0	$\log_2(1$ - (264 - 264 / 264 - 1) + $10^{-1})$	=	0.14
green	1	$\log_2(1$ - (264 - 19 / 264 - 1) + $10^{-1})$	=	2.12
large	2	$\log_2(1$ - (264 - 15 / 264 - 1) + $10^{-1})$	=	2.74
north-of(argentina)	2.5	$\log_2(1$ - (264 - 4 / 264 - 1) + $10^{-1})$	=	3.18
in(south-america)	2.5	$\log_2(1$ - (264 - 13 / 264 - 1) + $10^{-1})$	=	2.73
P	8	$\log_2(1$ - (264 - 1 / 264 - 1) + $10^{-10})$	=	33.22
VIP	2	$\log_2(1$ - (264 - 9 / 264 - 1) + $10^{-10})$	=	5.06

Table 6.6: Costs and certainty scores of the edges that can be used to describe Brazil as presented in Figure 5.4.

[5] http://www.cia.gov/cia/publications/factbook/

To generate a multimodal referring expression for Brazil equivalent to the ones presented in Figure 6.8, the Certainty Threshold is set to 10. The multimodal algorithm constructs a multimodal domain graph that unites the domain graph depicted in Figure 6.10 with the gesture graph depicted in Figure 6.11, that contains a precise pointing gesture for Brazil and a very imprecise pointing gesture that has the countries in the northern half of South America in its scope. The function *FindGraph* is subsequently called to search for the cheapest distinguishing referring graph in the multimodal domain graph that satisfies the Certainty Threshold. In the case that the other green countries on the world map are not considered as large, the cheapest distinguishing referring in Figure 6.12 graph contains the edges *country*, *green* and *large*, as depicted as graph H_1. Graph H_1 has cost $(0 + 1 + 2) = 3$ and the certainty score $(0.14 + 2.12 + 2.74) = 5$. Since the certainty score is lower than the Certainty Threshold $(5 \leq 10)$, the algorithm searches for another more expensive graph to increase certainty in the cheapest possible way. There are three candidate graphs that meet the Certainty Threshold: (1) The graph that contains the edges *country*, *green*, *large*, VIP and *in*(*south-america*), which has a certainty score of $(0.14 + 2.12 + 2.74 + 5.06 + 2.73) = 12.79$; (2) The graph that contains the edges *country*, *green*, *large*, VIP and *north-of*(*argentina*) with a certainty score of $(0.14 + 2.12 + 2.74 + 5.06 + 3.18) = 13.24$; and (3) The graph that contains the edges *country*, *green*, *large*, *in-south-*(*america*) and *north-of*(*argentina*), with a certainty score of $(0.14 + 2.12 + 2.74 + 2.73 + 3.18) = 10.91$. With two spatial relations, candidate (3) costs $(0 + 1 + 2 + 2.5 + 2.5) = 8$. Candidates (1) and (2) are somewhat cheaper and both cost $(0 + 1 + 2 + 2 + 2.5) = 7.5$. The algorithm thus returns graph (1) or graph (2) dependent on which edges are found first. This corresponds well with the descriptions produced in Study II as presented in Figure 6.8. When the algorithm incorporates the notion of salience as discussed in 4.5, the most salient relatum is chosen. Graph (1) is depicted as H_2 in Figure 6.12 and can be realized as 'the large green country in South America' with a very imprecise pointing gesture directed at Brazil. Graph (2) is depicted as H_3 in Figure 6.12 and can be realized as 'the large green country north of Argentina' with a very imprecise pointing gesture directed at Brazil.

Table 6.7 illustrates the performance of the algorithm, by presenting the descriptions from Figure 6.8 in comparison with the output of the algorithm, graphs H_2 and H_3 from Figure 6.12. As before two criteria are used on which the performance is measured: (1) *exact match* i.e., exactly the edges in the returned graph are represented in the description and no other information is included; and (2) *lean match* i.e., at least all the edges in the returned graph are represented in the description. Value 1 means there is a match, value 0 means no match. Table 6.7 shows that four of the ten descriptions exactly match the output of the algorithm and six do not. From the descriptions that do not match exactly, description (3) includes an extra locative expression and an extra pointing gesture; description

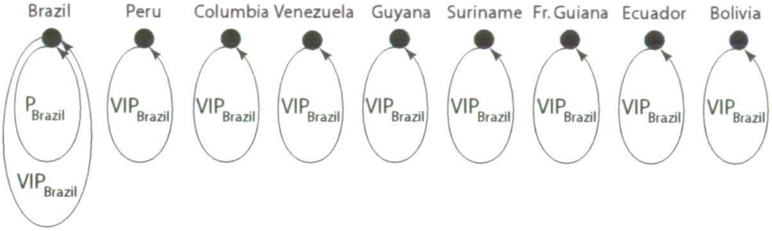


Figure 6.11: Gesture graph for Brazil as presented in Figure 6.9.

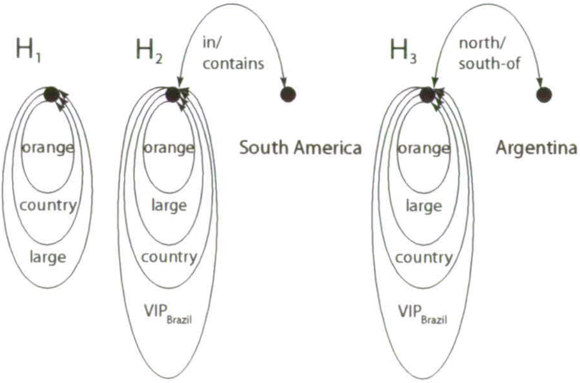


Figure 6.12: Referring graphs as generated by the algorithm for Brazil as depicted in Figure 6.9.

(5) includes a number of locative expressions, while the property *large* is omitted; description (7) fails to match exactly because of a repetition and descriptions, plus an extra pointing gesture; and descriptions (9) and (10) do not include one of the locative expressions *in South America* or *north of Argentina*. When looking at the lean match, six descriptions meet the algorithm's output and four do not. The descriptions that do not match are the descriptions: (5) which does not contain the property *large* and uses a lot of unexpected relata; (6) which does not include a locative expression and omits the property *large*; and (9) and (10) do not contain a locative expression.

		exact match	**lean match**
(1)	Brazil is ehm the (VIP circle) *large green country above Argentina*	1	1
(2)	ah Brazil that is (VIP) that *large green country in South America*	1	1
(3)	Brazil is (VIP circle) that *very large green part in South America* so *at the* (VIP) *upper part*	0	1
(4)	(VIP) that *large green area right below South America*	1	1
(5)	Brazil is (VIP circle) *South America* eh it is eh let's say the *green country* that *below ehm below borders below ehm borders with Bolivia Paraguay on top ehm among others Suriname*	0	0
(6)	Brazil lies (VIP circle) there ehm takes almost half of the ehm that isle and it is *green*	0	0
(7)	Brazil lies *in South America* (VIP circle) it is that *very large country* there the *green* the *biggest country* (VIP circle)*of South America*	0	1
(8)	Brazil is (VIP) the *green country* that eh is a *very large country* that *eh on yes they call the continent South America*	1	1
(9)	Brazil, you see there (VIP) the *large green area*	0	0
(10)	yes that is there (VIP) that *green large*	0	0

Table 6.7: Evaluation of the output of the algorithm by comparing the output, graph H_2 or H_3 as depicted in Figure 6.12 to the descriptions presented in Figure 6.8, where *exact match* is defined as 1 in the case that the edges in the graph are represented exactly in the description and 0 otherwise, *lean match* is defined as 1 in the case that the output of the algorithm is present in the description but other properties may be included as well, and 0 otherwise.

Example 3: Chile

As a second example in the map domain, consider Chile as a target. Figure 6.13 presents the descriptions of Chile by ten subjects in the far condition in Study II. Except for the description (1) and (5) all descriptions contain the properties *color*, *shape*, *size*, one or two spatial relations and one or more pointing gestures. Description (1) lacks a specific mentioning of South America as a spatial relation and description (5) does not include the property color. Apart from (4), all descriptions contain more than one spatial relation. Five descriptions, (1), (2), (6), (7) and (9) include Argentina as a spatial relation. Descriptions (1), (2), (3), (8), (9) and (10) include extra locative information to specify the location of Chile on the South American continent. All descriptions contain at least one very imprecise pointing gesture. Apart from description (4), in all descriptions at least one imprecise pointing gesture is performed which displays vertical movement during the stroke of the gesture. Descriptions (2), (3), (8) and (9), contain more than one pointing gesture.

(1) ehh.. dat is eh (VIP vert) *naast Argentinie* dat is dat *langerekte strook in het paars* eh *aan de linkerkant*

(2) Chili is een (VIP vert) *hele lange strook* die de hele kant afloopt *van* eh (VIP) *Zuid Amerika links onder* dus als je *onderaan t buurland* begint kom je kom je *tussen t* (VIP vert) *buurland* en eh peru de *smalle paarse strook naast Argentinie de grote gele Argentinie*

(3) Chili is die (VIP vert) *lange pier* daar he dus je ziet dat (VIP) continent daar Zuid Amerika geheten dan zie je die *lange paarse strook* daar *aan de eh linker kust* dat is Chili

(4) Chili is het *paarse land* dat eh *heel langwerpig eh hele langwerpige land* (VIP) *op het Zuid Amerikaanse continent*

(5) Chili ligt *in Zuid Amerika* en (VIP vert) dat is helemaal dat *hele langgerekte dunne land in het westen van Zuid Amerika*

(1) ehh.. that is eh (VIP vert) *next to Argentina* it is the *long stretched strip in purple* eh *on the left side*

(2) Chile is a (+ VIP vert) *very long strip* which runs along the whole border *of* eh (VIP) *South America bottom left* so if you start *at the bottom of the neighbor* you come you come *between the* (VIP vert)*neighbor and eh Peru* the *thin purple strip next to Argentina the large yellow Argentina*

(3) Chile is that (VIP vert) *long string* there eh so you see that (VIP) continent there called South America than you see that *long purple strip* there *at the eh left coast* that is Chile

(4) Chile is the *purple country* that eh *very long shaped eh very long shaped country* (VIP) *in the South American continent*

(5) Chile lies *on South America* and (VIP vert) that is wholly that *very long stretched thin country at the west of South America*

(6)	eh Chili dat ligt .. (VIP vert) daar is *langwerpig eh paars* en *rechts van een geel een geel land dat even langwerpig is bijna*	(6)	eh Chile that lies .. (VIP vert) there is *long shaped eh purple* and *to the right of a yellow a yellow country which is almost as long*
(7)	eh dat ligt *in* (VIP vert) *Zuid Amerika* dat *langgerekte paarse land* eh daar *naast Argentinie*	(7)	eh that lies *in* (VIP vert) *South America* that *long stretched purple country* eh there *next to Argentina*
(8)	Chili dat is *in Zuid Amerika* (VIP vert) het werelddeel *links onder helemaal linksonder* ligt een (VIP vert) *langgerech langgerekt paars lila land*	(8)	Chile that is *in South America* (VIP vert) the continent *left below wholly left below* lies a (VIP vert) *long str long stretched purple violet country*
(9)	Chili,... is volgens mij eh (VIP vert) deze *strook eh in Zuid Amerika aan de aan de westkant van* (VIP vert) *Zuid Amerika* die *lange verticale strook naast Argentinie* en het is *donkerrose paars*	(9)	Chile,... that is i think eh (VIP vert) this *strip* eh *in South America to* the *to the west side of* (VIP vert) *South America* that *long vertical strip next to Argentina* and it is *dark pink*
(10)	Chili is (VIP vert) die *rare hele smalle lange paarse strook* eh *links onderaan in Zuid Amerika*	(10)	Chile is (VIP vert) that *strange very thin long purple strip* eh *left below in South America*

Figure 6.13: Multimodal descriptions for Chile by ten subjects in the far condition in Study II. The properties and locative expressions are in italics. *VIP* = Very Imprecise Pointing gesture and *vert* = vertical movement in the stroke of the gesture.

For the generation of a multimodal referring expression for Chile the same costs of properties and relations pointing edges are used as defined before. Table 6.8 presents the costs and certainty scores of the edges that can be used to represent Chile on the map presented in Figure 5.4. To compute the certainty scores of the edges, the following figures are used: the world map consists of 264 countries, of which 15 are colored purple and 6 have long stretched shape. Furthermore, Chile is considered a medium-sized country, since there are 225 of the countries in the world which are smaller (see Footnote 5). When all countries that are equally large or smaller than Guyana are considered small, the number of middle-sized countries is ((264 - 15 large countries) - 178 small countries) = 71. For this example, the costs and certainty scores of two spatial relations are defined, a general one, *in-south-america*, and a more specific one *west-of-argentina*. Argentina is considered the most salient country in the southern half of South America, which is confirmed by the descriptions presented in Figure 6.13. The algorithm constructs a gesture graph as depicted in Figure 6.14, which contains a precise pointing gesture for Chile and an imprecise pointing gesture that has the countries in the southern half of South America in its scope. A precise pointing gesture P has cost 8 and a certainty score of 33.22. A very imprecise pointing gesture VIP costs 2 and has the certainty score 5.64, where 5 countries are assumed to be located in its scope. The gesture graph from Figure 6.14 is unified with the domain graph depicted in Figure 6.10.

	costs	**certainty scores**		
country	0	$\log_2(1 - (264 - 264 / 264 - 1) + 10^{-1})$	=	0.14
purple	1	$\log_2(1 - (264 - 15 / 264 - 1) + 10^{-1})$	=	2.74
long-stretched	1.5	$\log_2(1 - (264 - 6 / 264 - 1) + 10^{-1})$	=	3.06
medium-sized	2	$\log_2(1 - (264 - 71 / 264 - 1) + 10^{-1})$	=	1.43
west-of(argentina)	2.5	$\log_2(1 - (264 - 1 / 264 - 1) + 10^{-1})$	=	3.32
in(south-america)	2.5	$\log_2(1 - (264 - 13 / 264 - 1) + 10^{-1})$	=	2.73
P	8	$\log_2(1 - (264 - 1 / 264 - 1) + 10^{-10})$	=	33.22
VIP	2	$\log_2(1 - (264 - 5 / 264 - 1) + 10^{-10})$	=	5.64

Table 6.8: Costs and certainty scores of the edges that can be used to describe Chile as presented in Figure 5.4.

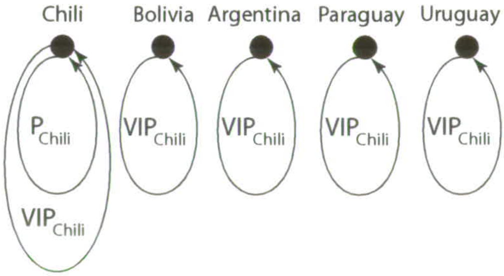


Figure 6.14: Gesture graph for Chile as presented in Figure 6.9.

To generate a multimodal referring expression similar to the ones presented in Figure 6.13 for Chile, the Certainty Threshold is set to 14, which is slightly higher than the Threshold needed for Brazil. The difference can be explained by the fact that, compared to Brazil, Chile is relatively small and therefore more difficult to describe as confirmed by the results from Study II presented in Section 5.4. In the case that the other long stretched countries on the world map are colored brown, the cheapest graph the algorithm can find is the one that includes the edges *type*, *color* and *long-stretched*. This graph, depicted as H_1 in figure 6.15, has the cost $(0 + 1 + 1.5) = 2.5$ and the certainty score $(0.14 + 2.74 + 3.06) = 5.94$, which is obviously lower than the Certainty Threshold $(5.94 \leq 14)$.

The graph containing the edges *country*, *purple*, *long-stretched*, *medium-sized* and VIP has a certainty score of $(0.14 + 2.74 + 3.06 + 1.43 + 5.64) = 13.01$, which means that the certainty score of the edge *medium-sized* is too low to increase the certainty score of the graph to a satisfying level. There are two candidates that meet the threshold, namely: (1) The graph containing the edges *country*, *purple*, *long-stretched*, VIP and *in-south-america* has a certainty score of $(0.14 + 2.74 + 3.06 + 5.64 + 2.73) = 14.31$; and (2) The graph that contains the edges *country*, *purple*, *long-stretched*, VIP and *west-of-argentina*, with a certainty score $(0.14 + 2.74 + 3.06 + 5.64 + 3.32) = 14.90$. Both options cost $(0 + 1 + 1.5 + 2 + 2.5) = 7$. Depending on which edges are found first, the algorithm returns graph (1) or (2), which suits the descriptions form Study II as presented in Figure 6.14. In the version of the algorithm in which the salience function is used, the most salient relatum can be chosen. Graph (1) is depicted as graph H_2 in Figure 6.15. This graph can be realized as 'the purple long stretched country in South America' with a very imprecise pointing gesture directed at Chile. Graph (2) is depicted as graph H_2 in Figure 6.15. This graph can be realized as 'the purple long stretched country west of Argentina' with a very imprecise pointing gesture directed at Chile.

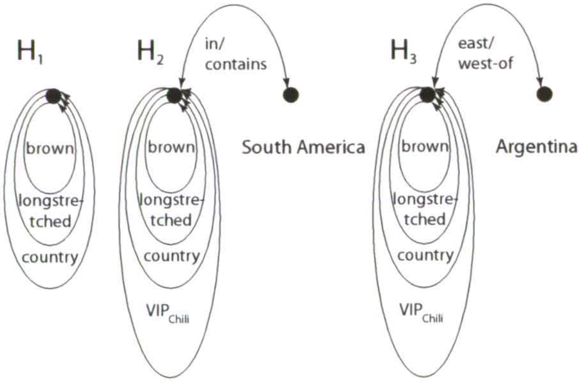


Figure 6.15: Referring graphs as generated by the algorithm for Chili as depicted in Figure 6.9.

When the multimodal referring expression is performed by an embodied conversational agent, the gesture included in the output can be enriched with a vertical movement during the stroke of the gesture. Following Theune et al. (2005), it is assumed that the generation of dynamic pointing gestures is not addressed in the phase of the Microplanner in which the referring expressions are generated, but in the Microplanner's lexicalization phase (see Section 2.3.2). In the lexicalization phase, it can be checked whether the generated graph includes the property *shape* and a pointing gesture, upon which the pointing gesture can be produced with a movement during the stroke of the gesture that corresponds with the shape of the target. This would be consistent with the ideas proposed by Schegloff (1984) on representational gestures as triggered by lexical items (c.f., Butterworth and Hadar, 1989).

Table 6.9 presents the performance of the output of the algorithm. The descriptions from Figure 6.13 are compared with graphs H_2 and H_3 from Figure 6.12. Two criteria illustrate the algorithms performance: (1) *exact match* i.e., exactly the edges in the returned graph are represented in the description and no other information is included; and (2) *lean match* i.e., at least all the edges in the returned graph are represented in the description. Value 1 means there is a match, value 0 means no match. Only two of the ten descriptions correspond exactly to the algorithm's output: (4) and (6). The other eight differ in that most of them contain a lot more information than the algorithm generates. Description (1) contains a more specific locative expression, i.e., *on the left side*. Description (2) contains more than one pointing gesture and a lot more linguistic information, including some repetitions. Description (3) contains two pointing gestures and a more specific locative expression, i.e., *on the left coast*. Description (5) contains an extra property *thin*. Description (7) contains both the locative expressions, *next to Argentina* and *in South America*. Description (8) contains two pointing gestures and a more specific locative expression *wholly left below*. Description (9) contains two pointing gestures, a property *vertical* and three locative expressions. Description (10) contains the property *strange*, and more specific locative expression *left below*. To sum up a lot more information is included in the descriptions than in the output of the algorithm. There are at least two reasons for this outcome: (1) Property edges like *vertical*, *strange* and *thin* as well as more specific locative expressions like *on the left coast* and *left below* are not defined in the domain graph used by the algorithm presented in Figure 6.10; and (2) The way in which the algorithm is defined, prevents the inclusion of multiple pointing edges. When looking at lean matches, the algorithm performs considerably better; nine descriptions match the algorithms output. Only description (5) is not in correspondence, because it does not contain the property *purple*.

		exact match	**lean match**
(1)	ehh.. that is eh (VIP vert) *next to Argentina* it is the *long stretched strip in purple* eh *on the left side*	0	1
(2)	Chile is a (+ VIP vert) *very long strip* which runs along the whole border *of* eh (VIP) *South America bottom left* so if you start *at the bottom of the neighbor* you come you come *between the* (VIP vert)*neighbor and eh Peru* the *thin purple strip next to Argentina the large yellow Argentina*	0	1
(3)	Chile is that (VIP vert) *long string* there he so you see that (VIP) continent there called South America than you see that *long purple strip* there *at the eh left coast* that is Chile	0	1
(4)	Chile is the *purple country* that eh *very long shaped eh very long shaped country* (VIP) *in the South American continent*	1	1
(5)	Chile lies *on South America* and (VIP vert) that is wholly that *very long stretched thin country at the west of South America*	0	0
(6)	eh Chile that lies .. (VIP vert) there is *long shaped eh purple* and *to the right of a yellow a yellow country which is almost as long*	1	1
(7)	eh that lies *in* (VIP vert) *South America* that *long stretched purple country* eh there *next to Argentina*	0	1
(8)	Chile that is *in south America* (VIP vert) the continent *left below wholly left below* lies a (VIP vert) *long str long stretched purple violet country*	0	1

		exact match	lean match
(9)	Chile,... that is i think eh (VIP vert) this *strip* eh *in South America to* the *to the west side of* (VIP vert) *South America* that *long vertical strip next to Argentina* and it is *dark pink*	0	1
(10)	Chile is (VIP vert) that *strange very thin long purple strip* eh *left below in South America*	0	1

Table 6.9: Evaluation of the output of the algorithm by comparing the output, graph H_2 or H_3 as depicted in Figure 6.15 to the descriptions presented in Figure 6.13, where *exact match* is defined as 1 in the case that the edges in the graph are represented exactly in the description and 0 otherwise, *lean match* is defined as 1 in the case that the output of the algorithm is present in the description but other properties may be included as well, and 0 otherwise.

6.4 Discussion

In this chapter a graph-based multimodal algorithm is presented that generates overspecified multimodal referring expressions, based on observations in human communication. The observations from (cognitive) linguistic research prove to be valuable in modeling the generation of referring expressions. A key feature is the notion of distant responsibility attributable to Clark and Wilkes-Gibbs (1986), which states that a speaker must be certain that the information provided in an utterance is understandable for the addressee. The analysis of multimodal overspecification presented in Section 6.1.2 confirms findings in cognitive linguistics, when looking at the linguistic part of the referring expressions in Study I. Redundant perceptual properties are included in descriptions to identify objects that are easily distinguished, while locative expressions are used for objects that are more complex to describe. From Study II, it can be inferred that overspecified descriptions do not occur in combination with precise pointing gestures. Overspecification appears when subjects are located at some distance from the target, and imprecise pointing gestures are used. The use of pointing gestures is related to overspecified accompanying linguistic descriptions in two ways: (1) Co-occurrence of pointing gestures and locative expressions; and (2) Co-occurrence of dynamic pointing gestures and the description of *shape*.

From the analysis it appears that imprecise pointing gestures are generally combined with locative expressions. Moreover this co-occurrence is applied more often in identifying difficult targets, than in reference to targets that are easy to identify, which suggests their positive effect on the certainty of the speaker. About half of the pointing gestures analyzed in Study II involve some kind of movement during the stroke of the gesture. Although the generation of dynamic pointing gestures is assumed to be taken care of in the Microplanner's lexicalization phase, which is not addressed in this thesis, it can be noted that a dynamic pointing gesture is performed in more than two thirds of the cases in which the property *shape* is mentioned in the linguistic part of the referring expression. However, from the data resulting from Study II, it can be inferred that dynamic pointing gestures are applied far more often than *shape*. This provides evidence for the model presented by Kita and Özyürek (2003), which suggests that gestures, although affected by the linguistic part of an utterance, may express properties not communicated by the linguistic expression alone. As such, the information conveyed in the movement during the stroke of a pointing gesture may be (partially) redundant or entirely complementary to the linguistic referring expression. It is therefore recommended to look for other criteria on the basis of which dynamic pointing gestures can be generated.

Based on the observations in human communication, the algorithm employs a notion of certainty to generate multimodal referring expressions that range from minimal to highly overspecified ones. The certainty score which is computed for each distinguishing graph generated by the algorithm indicates the algorithm's estimation of the probability that the hearer might misunderstand the referring expression that can be realized from the graph. If the certainty score exceeds the Certainty Threshold the graph is returned. The certainty score of a graph is computed by summing the scores of the edges of the graph. The ease of perceptibility of properties and relations is decoded in a domain dependent computation of the certainty scores of linguistic edges. Obviously this is only one way to determine the scores. Another option can be to relate the certainty scores of the properties to the scope of the pointing gestures and the scope of the spatial relations included in a referring graph. In this way, the domain can be narrowed down to a context set that contains the objects located in the scope of a pointing gesture, similar to the revised definition of focus space presented in Section 4.5. In its current form the algorithm has a number of nice consequences that mimic human behavior:

- Referring expressions are generated dependent on contextual influences such as task importance, salience and number of distractors;

- Additional informative properties, relations and pointing gestures always increase certainty;

- Properties and relations which have no additional value in identification are not included in overspecified referring expressions;

- In the case of a precise pointing gesture is generated no redundant information is included in the description;

- The notion of certainty provides for the inclusion of more specific locative expressions, for instance *upper left* instead of just *left*.

In Section 6.3, the predictions of the algorithm are compared with the referring expressions produced by humans. In the near condition, precise pointing is cheap, and hence always selected. In these cases, the algorithm does not overspecify and neither do human speakers. In the far condition, precise pointing is prohibitively expensive, and hence not done. To meet the Certainty Threshold, the algorithm produces overspecified descriptions. For three targets, these are compared to the overspecified descriptions produced by human speakers; which result in 40 % exact matches and 77 % lenient matches (including the exact matches). These findings are representative for the other multimodal referring expressions produced by humans.

In NLG, the overspecification of referring expressions has so far received virtually no attention. One exception is the work by Jordan, which is based on the Incremental Algorithm by Dale and Reiter (1995), as shown in Chapter 3. The Incremental Algorithm is capable of generating overspecified referring expressions, however not on a well-founded basis. By chance, that is by the interaction between the preferred attributes and the distractor set, overspecification might appear in the generated referring expressions. It can be inferred that, dependent on the variability of the objects in the domain, the more objects in the distractor set the higher the degree of overspecification. Accordingly, one way to generate overspecified descriptions is to alter the definition of the set of distractors. In this respect Jordan (2002) (see also Jordan, 1999; Jordan, 2000) reports on the overspecification of object descriptions in the COCONUT corpus, which consists of 24 dialogues in each of which two people collaborate on buying furniture. Jordan performs a comparative study of object descriptions that have already been mentioned and generated descriptions that just distinguish the objects from their distractors. The latter descriptions are based on several distractor set definitions. These definitions range from the inclusion of all discourse objects mentioned so far, to including only the objects from the previous utterance in the set. Besides using recency in this way, Jordan also tries the distractor set definition by Passonneau (1996) based on Centering Theory (Grosz et al., 1995). In the COCONUT corpus, all objects have only four properties: *owner*, *color*, *price* and *type*. In a comparison of the corpus descriptions to the distinguishing descriptions determined by the various distractor sets, Jordan finds that on average the degree of overspecification of the descriptions in the corpus is higher than the degree of overspecification in the generated descriptions, regardless of which distractor set definition is used. This suggests that including all discourse objects in the distractor set performs best in mimicking overspecification as produced by human speakers. Including all properties in the distractor set might be adequate when the number of properties is very small, like in the COCONUT corpus; however, the kind of overspecification realized is not controlled by the algorithm when solely dependent on the objects in the distractor set and a list of preferred attributes. Furthermore, although the redundancy in the descriptions generated by the Incremental Algorithm may seem psychologically plausible, they lack the fundamental basis in linguistic behavior which is advocated in the approach presented here by considering the notion of uncertainty.

From a different perspective Horacek (2005) also looked at uncertainty. In an attempt to select properties on the basis of more specific criteria, Horacek (2005), very recently proposed to implement a notion of uncertainty in the Incremental Algorithm. He describes three kinds of uncertainty: (1) Uncertainty about knowledge (i.e., the user might not be familiar with terms used in a description); (2) Uncertainty about perceptual capabilities (i.e., the user might not be able to perceive particular aspects of a target); and (3) Uncertainty about conceptual agreement, with respect to vague properties (i.e., the user might interpret relative properties like *large* differently). These three kinds of uncertainty are described in terms of probabilities, multiplied to obtain the probability of recognition of a referring expression. The Incremental Algorithm is altered in order to generate referring expressions adapted to the user that needs to interpret them. Each attribute-value pair is enriched with a probability of recognition that indicates if the user is able to understand the property. Hence the algorithm selects properties, that result in the certain improvement in terms of probability of identification. A repair mechanism with which more or other properties are selected, is used to increase this probability in the case that the likelihood of the correct identification of a referring expression is not satisfactory. The three kinds of uncertainty observed by Horacek (2005) can be employed to gain control over the linguistic information to be added in a multimodal referring expression. Especially in the selection of more specific properties like *vermillion*, instead of *red*, this notion may help to decide whether the user is familiar with the term and is able to interpret it correctly. As a result it can be decided to what extent a referring expression will be understood by the user in terms of linguistic and perceptual knowledge. It would be interesting to combine this approach with the algorithm that generates overspecified referring expressions as presented here.

As seen in Section 6.1, several factors play a role in the generation of overspecified referring expressions. In particular, context and discourse related factors are of high importance. Based on observations in human communication, the approach to generate overspecified referring expressions proposed in this thesis uses three independently motivated parameters (cost, certainty score and a Certainty Threshold). The algorithm balances the cost and the certainty score of the edges in order to find the cheapest graph that meets the Certainty Threshold. In this way a wide range of referring expressions can be generated, from minimal to highly overspecified ones. Manipulation of the Certainty Threshold ensures not only successful reference grounding but also provides a way of expressing additional information affecting dialogue in a broader sense. For instance, a high

Certainty Threshold can be due to the newness of the referent in the discourse or to the importance of some aspect in the dialogue, as in the case of a commitment or a focus shift. A low Certainty Threshold may follow from the fact that the target is very salient or has a low number of distractors, which causes less overspecified or minimal referring expressions to be generated.

A possible drawback of formalizing the cost, certainty score and Certainty Threshold as independent parameters is that this might result in a complex interaction between the cost and the certainty score. For instance, from the examples described in Section 6.3, it can be inferred that The Certainty Threshold needs to be set not only dependent on context and discourse factors, but also dependent on the type of objects to be referred to. While the costs of all edges are standardized in the examples, it turns out that different values for the Certainty Threshold are necessary to reach satisfactory results even within one object domain (e.g., compare the settings for Brazil and Chile). Moreover, the worked examples presented in Section 6.3 show that the parameters are domain dependent. In the domain of geometrical objects, the algorithm performs rather well, whereas the algorithm has more difficulties in the world map domain, in which the variety of the referring expressions produced by the participants is much higher. In the latter domain speakers tend to use not more than one property, while locative expressions are highly favored. Adjusting the parameter settings in order to reach better matches can be done, for example, by lowering the cost of relational edges compared to the linguistic edges that represent the properties. In summary, the proper setting of the algorithm's parameters is an empirical issue. It could be interesting to try to determine optimal settings automatically on the basis of empirical data.

Chapter 7

General Discussion

This final chapter is organized as follows: Section 7.1 presents a concise overview of the work discussed in this thesis. Section 7.2 zooms in on the multimodal GRE algorithm. In Section 7.3 some unsettled issues are considered. The chapter is closed with an inventory of the matters that might be addressed in future work in Section 7.4.

7.1 Overview

With the design of more advanced application systems, the questions arise not only how such systems should generate descriptions in which linguistic information and gestures are combined but also, how such multimodal referring expressions are produced by humans. The research in this thesis has focused on two aspects of the need of more advanced multimodal presentations: (1) In what way is the generation of multimodal utterances directed by the context? and (2) Which factors determine what modality or which combination of modalities is used under which conditions? This thesis has formulated an answer to these questions by means of an algorithm which generates multimodal referring expressions which reflects their occurrence in human communication.

The background for this algorithm was presented in Chapter 2, where the multimodal communication in HCI and human communication were discussed as modeled in current times. Subsequently in Chapter 3 a detailed study was presented of NLG algorithms that generate referring expressions, whereby the algorithms were presented in a uniform format which facilitates comparison. Additionally, a new multimodal notion of salience was proposed in which linguistic and perceptual salience are combined. The multimodal algorithm has built on this research in that it uses and extends on aspects that have been resolved in the development of GRE, whilst mending flaws and filling gaps at the same time. The multimodal algorithm was evaluated by the results of two production experiments where human production of multimodal referring expressions was studied. On the basis of this evaluation, the algorithm was adjusted and refined.

As seen in Chapter 2, algorithms that generate multimodal referring expressions tend to produce rather simple descriptions, based on relatively straightforward context-independent criteria. Furthermore, in these algorithms the decision to point is not connected to the process in which the linguistic referring expression is generated. The model for pointing proposed in this thesis provides for a close coupling between the linguistic information and the pointing gesture used. The algorithm in which this model was formalized generates various pointing gestures, precise and imprecise ones. The type of pointing gesture is closely linked to the perceptual context in that the scope of an imprecise pointing gesture contains more objects than the scope of a precise pointing gesture. A direct consequence of this model for pointing is that the number of linguistic properties required to generate a distinguishing multimodal referring expression is predicted to co-vary with the kind of pointing gesture used.

In Chapter 4, the model for pointing was implemented as a multimodal graph-based algorithm for the generation of referring expressions. This algorithm approaches the generation of referring expressions as a graph construction problem using subgraph isomorphism. The decision to point is made on the basis of cost functions which are grounded in a fundamental law about the human motor system (Fitts, 1954). The proposed algorithm is in more than one sense context-sensitive. The algorithm generates the referring expressions based on a three-dimensional notion of salience, proposed in Section 3.5, which acknowledges the linguistic context, the focus of attention and the inherently salient objects in the domain. The output of the algorithm is based on a trade-off between the costs of a pointing gesture and the costs of the linguistic information needed to single out a target object.

Evaluation of these kinds of NLG algorithms is difficult, because in linguistic corpora the objects and their properties that are referred to are unknown; only the surface realization is present. Evaluation of multimodal referring expressions is even harder, because multimodal corpora are scarce and the basis on which speakers decide which modality to use is concealed. In Chapter 5 it was shown that these problems can be resolved by using production experiments in which participants identify stimuli by speech and gesture. In this way, spontaneous multimodal data is gathered on controlled input. Two studies were carried out in which participants refer to objects that differ in shape, size and color. One study had a very strict setting; pointing was forced and no feedback was given (Section 5.3). The other study was performed in a more natural and interactive setting (Section 5.4). The analysis of the data resulting from these experiments, indeed reveal a strong interplay between pointing gestures and linguistic material. Moreover the studies provide information about which linguistic properties are used for which kind of target. Interestingly, the referential strategies differ for different kinds of targets. Objects that are easy to identify are described by their properties, whereas objects that are relatively difficult to identify are referred to by the use of locative expressions.

Another conclusion that can be drawn from the experimental data is that the descriptions that are used to identify objects in a domain may be overspecified, in particular when referring to targets that are far away or difficult to describe. Chapter 6 offered a more detailed survey on the redundant information contained in referring expressions, considering both unimodal and multimodal overspecification. In order to automatically generate overspecified multimodal referring expressions as occurring in human communication two questions were considered: (1) Why and when do speakers overspecify? and (2) How do speakers overspecify? While the notion of overspecification is understudied in the field of NLG, it has been addressed by a number of researchers in (cognitive) linguistics. Inspired by observations by among others Pechmann (1989), Maes et al. (2004) and Arts (2004) and the findings of the production experiments described above, a variant of the multimodal algorithm was proposed that is able to generate overspecified descriptions based on a notion of certainty. The algorithm uses an estimation of the probability that the user might misunderstand a particular referring expression. The adapted algorithm outputs the cheapest distinguishing expression that meets a pre-defined Certainty Threshold. By varying a predefined Certainty Threshold the algorithm is capable of generating referring expressions with a variety of overspecification levels (from minimally specified to substantially overspecified).

7.2 Generating Multimodal Referring Expressions

7.2.1 Multimodal GRE Algorithm

In contrast to existing algorithms, the algorithm presented in this thesis approaches the generation of multimodal referring expressions as a compositional task in which linguistic and gestural resources are combined. The linguistic means included to distinguish a target are properties and spatial relations, whereas the gestural part of a multimodal referring expression is restricted to deictic pointing gestures. The approach taken to pointing gestures is visualized with the Flashlight Model for pointing (Section 4.2.1), which allows different gradations of pointing precision, ranging from precise and unambiguous to imprecise and ambiguous. A precise pointing gesture has a high precision for both speaker and hearer. Its scope is restricted to the target object, and this directly rules out the distractors. On the other hand an imprecise pointing gesture generally includes some distractors in its scope because of a larger distance between the speaker and the target.

The GRE algorithm used to generate multimodal referring expressions is an extension of the graph-based algorithm proposed by Krahmer et al. (2003) (Section 4.2.2). The multimodal graph-based algorithm uses a domain graph to represent the domain of conversation as a labeled directed graph. The objects in the domain graph are defined as the vertices (or nodes) in the graph. The properties and relations that can be used to identify the objects in the domain are represented as edges (or arcs). For every target object a gesture graph is generated that includes the vertices and edges representing the pointing gestures directed from different distances to the target. Subsequently, the gesture graph is fused with the domain graph in order to obtain a multimodal domain graph. To generate a multimodal referring expression the graph-based algorithm searches for the cheapest subgraph (i.e., a referring graph that represents the target in the multimodal domain graph). The algorithm does not use an a priori criterion to decide when to use a pointing gesture. The output modality is determined by a trade-off between the cost of pointing and the cost of a linguistic description, which have to be defined on an empirical basis. In the implemented model for the generation of multimodal referring expressions key terms are: effort, salience and certainty.

The algorithm accommodates the principle of minimal **effort** (Clark and Wilkes-Gibbs, 1986) by means of cost functions that determine the inclusion of properties and pointing gestures in the search for the cheapest referring expression. Linguistic properties and relations are assigned costs according to preference similar to the list of preferred attributes proposed by Dale and Reiter (1995) (Section 3.3.2). Because it takes less effort to perceive the object's absolute properties as opposed to its relative properties (i.e., relative properties depend on the other objects in the domain), it is assumed that absolute properties are preferred over relative properties. Relations are considered to be even more expensive because the relatum of the target needs to be described as well. From the speakers point of view an imprecise pointing gesture is intuitively less expensive than a precise pointing gesture. There is neurological evidence that the production of an imprecise pointing gesture indeed takes less effort than the production of a precise pointing gesture. The costs of the various pointing gestures are derived from an empirically motivated adaptation of Fitts' law (1954) (Section 4.3.5). By modeling the cost as a function that relates the size of the target to the distance of the target, the amount of linguistic information and the kind of pointing gesture used to identify a target are predicted to co-vary depending on the effort needed to produce and interpret the multimodal referring expression.

To generate context-sensitive referring expressions a multimodal notion of **salience** is implemented in the algorithm, which fuses the visual and the linguistic context and therefore guarantees context-sensitive descriptions both linguistically and visually (Section 3.5). Each object in the domain receives a salience weight which is the weighted sum of three salience weights: linguistic, inherent and focus space salience of the object. The different types of salience are weighted in correspondence with their importance; linguistic salience is more important than focus space salience, which is more important than inherent salience. By using salience weights the algorithm can restrict the context set in three ways: (1) The algorithm closely monitors the linguistic context to compute the linguistic salience; (2) The algorithm explicitly tracks the focus of attention to compute the focus space salience, whereby the focus space is extended with the objects that are in the scope of the pointing gesture used to indicate the last mentioned target object (Section 3.5.3); and (3) it acknowledges inherently salient objects. The salience function assigns salience weights to every vertex in the domain graph. On the basis of the salience weights of the vertices in the domain, the context set is constructed as a subgraph of the domain graph, which only contains the vertices that are at least as salient as the target. As a result, when an object is salient, in general reduced information can be used for identification.

Based on observations in human communication on why, when and how overspecified referring expressions are produced (Section 6.1), the algorithm is adapted in order to generate overspecified referring expressions, both unimodally and multimodally. A key feature in human communication is the notion of distant responsibility (Clark and Wilkes-Gibbs, 1986), which claims that a speaker must be certain that the information provided in an utterance is understandable for the user. This factor is implemented in the algorithm by modeling a notion of **certainty**, with which multimodal referring expressions can be generated that range from minimal to highly overspecified ones. For each referring graph generated by the algorithm a certainty score is computed (Section 6.2.1), which is derived from a summation of the certainty scores of the edges contained in the graph. The certainty score of the referring graph is compared to the Certainty Threshold, which is derived from context and discourse related factors. If the certainty score satisfies the Certainty Threshold, the algorithm judges the probability that the hearer might understand the referring expression sufficient and the graph is returned to be realized in natural language.

Both linguistic edges and pointing edges increase certainty when added to a referring graph (Section 6.2.2). A domain dependent method is used to determine the certainty scores of the properties and relations, in which the occurrence of a property or the relation between two objects is related to the variety of properties and relations in the whole domain. The certainty scores of the pointing gestures is defined as a relation between the number of objects in the scope of the pointing gesture and the total number of objects in the domain. The algorithm balances the costs and the certainty scores of the properties, relations and pointing gestures in order to find the cheapest graph that meets the Certainty Threshold. Thus the algorithm generates overspecified referring expressions based on an independent method derived from observations in human communication.

7.2.2 Discussion

The algorithm presented in this thesis allows for a compositional approach to the generation of multimodal referring expressions, in which the target is adequately distinguished by a suitable selection of pointing gestures and linguistic information in a given context. The graph-based algorithm presents an appropriate framework to implement the Flashlight Model for pointing outlined in Section 4.2.1. This model predicts an interaction between the linguistic information and the kind of pointing gesture used to distinguish a referring expression; the less precise the pointing gesture, the more distractors are contained in the scope of the pointing gesture and the more linguistic information is needed to single out the target. In contrast, an incremental strategy for the generation of multimodal descriptions does not seem to be straightforward, because its lack of backtracking entails that all selected properties, relations and pointing gestures are realized. As a result the composition within the referring expression to be generated is uncontrolled in that multiple gestures directed at the same object can be generated and precise pointing gestures may be produced with a lot of redundant linguistic information. This seems to suggest that the Flashlight Model is inherently non-incremental.

To generate multimodal referring expressions the graph-based algorithm does not need an a priori criterion to decide when to include a pointing gesture in a distinguishing description. Rather the decision to point is based on a trade-off between the cost of pointing and the cost of a linguistic property. As such, the amount of linguistic properties required to generate a distinguishing multimodal referring expression is predicted to co-vary with the kind of pointing gesture. This approach has at least three consequences: (1) An isolated object does not require precise pointing; there is always a graph containing a less precise and hence cheaper pointing edge which has the same objects in its scope as the more precise pointing gesture; (2) The algorithm never outputs a graph with multiple pointing edges to the same target, since there is always a cheaper graph which omits the less precise one; and (3) A precise pointing edge and a relational edge never occur together in a distinguishing graph, because a graph that contains a precise pointing gesture is distinguishing.

To allow for context-sensitive descriptions, the algorithm incorporates a multimodal notion of salience that accommodates linguistic and visual salience. The approach integrates three different types of salience: linguistic salience, inherent salience and focus space salience. Objects that have been talked about, objects that stand out in relation to the other objects in the domain because of certain properties and objects that are located in the focus of attention, are enriched with salience weights that reflect their prominence in the domain. Every object in the domain receives a salience score, on the basis of which the context set, the number of distractors from which the target has to be distinguished, can be reduced to the set of objects that have a salience score equal to or higher than the target. In this way salient targets can be described in a relatively concise way. Due to the salience function the selection of relata can be restricted to only suitable salient objects that are selected from the context set. Moreover, descriptions of objects that are salient are less likely to contain a pointing gesture, unless pointing is very cheap. If an object is salient, this generally implies that its distractors are relatively few. This in turn implies that fewer (or less expensive) properties are required to rule out the distractors. Contrastively, in the case of a focus shift a pointing gesture is more likely to be included, because the target is relatively less prominent and more edges have to be included to rule out a larger number of distractors.

In order to generate overspecified referring expressions, the multimodal algorithm uses a notion of certainty, which is defined in terms of an estimation of the probability that the user might misunderstand a particular referring expression. The algorithm selects a referring expression that has an adequate, context related certainty score for the lowest cost. In general, adding properties, relations or pointing gestures increase certainty. Certainty scores of properties and relations are computed in relation to their occurrence in the domain, which causes prominent features of an object to have a high certainty score, whereas features that are shared with other objects in the domain are considered to have a low influence on the referring expression. The certainty scores of pointing gestures are calculated from the number of objects in the scope of the gesture with respect to the number of objects in the whole domain. In the case that a precise pointing gesture is generated, no redundant linguistic information is included in the description; whereas in the case that an imprecise pointing gesture is selected redundant properties and relations can be selected as well. Properties and relations which have no additional value in identification are not likely to be included in the referring expression, because they do not increase certainty while they do increase the cost. Moreover, the notion of certainty provides for the inclusion of more specific locative expressions, for instance *upper left* instead of just *left*.

Even though the algorithm has various attractive properties, it has some disadvantages as well. One is its computational complexity. Although various ways exist to reduce this complexity (see Krahmer et al., 2003), when moving to the generation of multimodal referring expressions, there are two other ways to reduce the search space and thereby reduce runtime. First, with the generation of pointing gestures, it is always possible to single out one object from the others due to the presence of unambiguous pointing edges. A precise pointing gesture can always be generated even if it is very expensive. As an advantageous side effect of this a polynomial upper bound is obtained for the theoretical complexity of the algorithm. At least the cost of one distinguishing graph for the target object is known: the graph consisting of only a vertex for the target object and a precise pointing edge. This means that not all subgraphs of the merged multimodal graph have to be inspected, but only those subgraphs which do not cost more than the precise pointing graph. A second method to decrease runtime is the use of salience. By reducing the number of the distractors to the set of objects that have a salience weight equal to or higher than the target object the domain graph can be limited in order to reduce the number of referring graphs that have to be checked.

Another remark concerns the various parameters the algorithm uses. The selection of properties, relations and pointing gestures is determined by the cost of the edges, the additional certainty that is supplied by the edges and the Certainty Threshold. These three features provide for a complex interaction, which needs careful consideration in order to obtain optimal settings. Two kinds of interactions can be distinguished: (1) The interaction between the pointing gestures and linguistic information; and (2) The interaction between the costs, the certainty scores and the Certainty Threshold. In the following discussion these interactions are considered.

With respect to the interaction between the linguistic information and the pointing gestures, the algorithm uses two independent methods of determining the cost of the edges. On the one hand, the notion of preferred attribute list proposed by Dale and Reiter (1995) is adopted to define the cost of linguistic edges. The underlying idea of this approach, in which absolute properties are considered cheaper than relative properties, is extended to determine the cost of spatial relations. On the other hand, the costs of pointing gestures is determined by the use of Fitt's law (Fitts, 1954), which relates the distance to the target to the size of the target. These methods are used in a way in which the costs of the linguistic edges and the costs of the pointing gestures are aligned intuitively. To relate the costs of the linguistic properties and relations to the costs of pointing gestures, a more balanced cost function is called for. A first step in a more balanced approach could be a redefinition of the costs of linguistic edges in terms of the notion of discriminatory power as defined by Dale (1989), in which the occurrence of a property is related to the number of objects in the domain. In this way the cost of a property is valued with respect to the occurrences of the property in the whole domain. For instance, in a domain with only one green object, the property *green* might be relatively cheap. A second step in this approach relates the costs of the linguistic edges to the scope of pointing gestures. When a pointing gesture is selected to be produced in a referring expression, the costs of the linguistic edges could be redefined in correspondence to the occurrences of the various edges and the number of objects in the scope of the gesture. As such the use of a pointing gesture might be considered as a resize of the domain.

The interaction between the costs, the certainty scores and the Certainty Threshold is an empirical issue for which more data has to be collected. In Section 6.1, some aspects were discussed that influence the setting of the Certainty Threshold in a given situation, such as discourse goals, task importance, modes of communication and situational conditions. The Certainty Threshold has a very complicated effect on the number and the kind of edges that should be selected to describe a target. A high Threshold does not necessarily mean that the number of edges to describe an object increases. Dependent on the cost, a high threshold might also result in the selection of fewer edges that each have a relatively high certainty score (e.g., instead of some relatively cheap properties, a couple of spatial relations can be selected, or maybe a precise pointing gesture). The algorithm weighs the kind of referring expression to be generated based on the interplay between the costs, certainty scores and Certainty Threshold. As seen in the production experiments reported on in Chapter 5, the kind of linguistic material used depends on how difficult it is to describe a target. Targets that are easy to describe are often referred to by properties, whereas targets that are more difficult to describe are identified in terms of relations. Correspondingly, apart from the aspects mentioned above, the kind of target also has an influence on the Certainty Threshold. As shown in Section 6.3.2, the descriptions of Brazil and Chile require different threshold settings. Intuitively, objects which are easy to identify do not require precise pointing, whereas the more difficult objects, like for example Luxembourg, demand very overspecified descriptions. In some of the latter cases a precise pointing gesture might be used to decrease the amount of effort necessary for identification. It should be investigated to what extent, the interaction between costs, certainty scores and Certainty Threshold can be controlled in order to generate appropriate referring expressions.

7.3 Loose Ends

This section addresses three aspects that have not been considered yet: (1) The output of multiple pointing edges per graph; (2) Pointing gestures directed towards sets; and (3) The generation of dynamic pointing gestures.

As seen in Chapter 5, speakers occasionally produce several pointing gestures per referring expression: when located at a distance to the target and when the target is somewhat difficult to identify with a linguistic description. After inspection it turned out that most of these pointing gestures are directed towards relata. Currently the algorithm, while searching for minimal descriptions, does not produce descriptions that contain more than one pointing gesture. To generate pointing gestures directed towards relata, the algorithm needs an extensive multimodal graph that contains gesture graphs for all objects in the context set, or at least for all objects in the focus space of the current target. These pointing edges are intuitively rather cheap, since they are only included when the target is indicated by a pointing gesture as well. When a pointing gesture is produced to indicate the target, a pointing gesture to indicate the spatial relation to the target demands only a minor distance to be crossed by the hand. Accordingly, as illustrated in Figure 7.1, it is assumed that a pointing gesture directed towards the relatum is produced with similar precision as the pointing gesture directed towards the target, which implies that the cost of a pointing gesture directed towards a relatum is dependent on the pointing gesture directed towards the target. To compute the cost of a pointing gesture directed towards a relatum, the cost function defined in Section 4.3.5 has to be interpreted slightly differently. To derive the cost for a pointing gesture directed at a relatum of the target, the distance can be computed from the position of the hand while pointing at the target to the required position of the hand for pointing to the relatum.

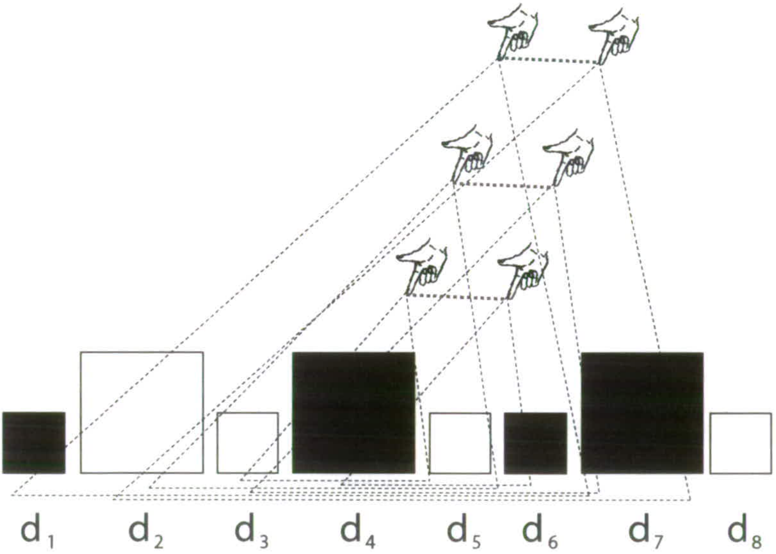


Figure 7.1: Pointing to a relatum, where the distance is measured from current position of the hand that points at the target to the position required to point at the relatum with similar precision.

Pointing gestures directed towards sets of objects can be included when the algorithm is extended to refer to sets of objects in the way proposed by van Deemter and Krahmer(to appear). An extension of the algorithm to refer to sets requires a change in the input, which defines the target not as a single object, v, but as a set of objects, S (where S of course can be defined as a singleton set to refer to one object). This revision implicates that the algorithm has to search for a referring expression that identifies each of the objects in the set S and not any of the objects in the domain graph outside S (see discussion of the Plural Algorithm, Section 3.4.1). This extension, especially, has an effect on the generation of precise pointing gestures. As indicated in Figure 7.2 the scope of a precise pointing gesture needs to include the objects contained in the set, which makes the pointing gesture somewhat less precise. In such a case, a dynamic pointing gesture, for example one that employs a circular movement which encloses the objects that are referred to might help to obtain more precision. However, in cases where the objects in the set are not located close to each other (e.g., all black objects in Figure 7.2) the generation of one precise pointing gesture is not sufficient. To solve this in the realization phase it can be decided that the objects contained in the set are pointed at separately, which yields multiple precise pointing gestures that need to be coordinated with the linguistic description. Of course, this can only be an option, when the target set is relatively small.

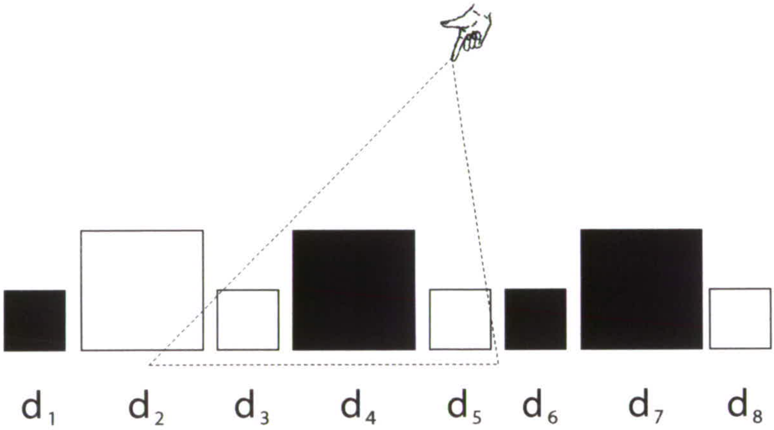


Figure 7.2: 'These objects'.

When looking at the generation of dynamic pointing gestures used to indicate single targets, it has been suggested in Chapter 6 to produce a movement in the stroke of the gesture when the property *shape* is included in the linguistic description. In line with the model proposed by Theune et al. (2005), such representational gestures can be generated in the Microplanner's lexicalization phase. However, evidence supplied by the observations in human communication presented by Kita and Özyürek (2003) as well as by the studies presented in this thesis, suggests that the occurrence of a representational gesture does not necessarily hinges on the verbalization of a property that indicates the appearance of an object (see also Kopp et al. (2004)). This implies that representational gestures can provide complementary information and have distinguishing qualities of their own. For instance, when identifying Chile on a world map, a vertical movement produced together with the property *long-stretched* might be used to rule out a horizontally long stretched distractor like Russia. To be able to use representational gestures to single out a target, they should be selected in the phase of the Microplanner in which the referring expressions are generated. The multimodal algorithm provides the architecture to generate these representational gestures by an extension of the multimodal graph with edges that represent the kind of movement to include in the accompanying gestures. For example, the vertex that represents Chile might have an edge *vertical*, with which a representational vertical gesture can be generated that indicates the north-south orientation of the target. The cost of such gestures is an empirical matter, for which other criteria than the occurrence of the property *shape* have to be discovered.

7.4 Future Research

Besides the complex relation between different parameters as mentioned in Section 7.2.2 and the role of representational gestures as discussed in Section 7.3, the following issues are planned to be addressed in future research. Although the Flashlight Model for pointing discussed in Section 4.2.1 presents an idealized picture of the effect of different kinds of pointing gestures, it has been acknowledged that the scope of an imprecise pointing gesture is vague, i.e., it is hard to tell which objects in the domain are located in its reach. In order to make exact predictions on the costs of properties and on the use of overspecification and to compute the focus space salience, a definition of the scope of a pointing gesture is needed. A way to obtain this definition might be through production experiments in which the participants have to identify objects that are located at a certain distance, where the linguistic descriptions produced are analyzed with respect to the visual context (see Kranstedt et al., 2003; Lücking et al., 2004; Kranstedt et al., 2005; Kranstedt and Wachsmuth, 2005; Kranstedt et al., to appear). By an evaluation of exactly which objects are ruled out by the linguistic description that accompanies a pointing gesture, the scope of the pointing gesture can be validated. In such experiments the use of eye-tracker techniques might be of valuable help (Metzing and Brennan, 2003).

The experiments reported in Chapter 5 address the relation between the kind of pointing gestures used and the amount of linguistic information that accompanies the gestures. However, it also needs to be established when speakers actually use a pointing gesture together with or instead of a linguistic description. Although the results of Study II indicate that speakers use pointing gestures all the time even if they are not obliged to do so, it can be argued that the pointing gestures produced by the participants were evoked by the nature of the task (i.e., identifying countries on a map). Closer inspection of the underlying principles might indicate the use of pointing gestures to be a speaker dependent issue; some people point a lot and some speakers do not, or speakers that at some time in an interaction started to point might keep on using pointing gestures.[1] Such a speaker

[1] Such behavior can be observed in videos obtained from the experiments conducted by Beun and Cremers (1998), (c.f., Lücking et al., 2004).

dependent modality choice implies that the interaction between the linguistic information and pointing gestures should be related to the attitude attributed to the system or to the preferences of the user. Evaluation of the algorithms performance demands the application of the algorithm to multimodal dialogue systems, so that experiments can be conducted similar to the ones reported by Reeves and Nass (1996).

As discussed in Chapter 1, the work presented in this thesis aims to support the development of more advanced HCI systems. In Section 2.5.1, two different types of multimodal dialogue systems are presented: (1) Systems using ECAs that identify objects by the combined use of linguistic descriptions and pointing gestures; and (2) Systems that indicate objects via highlighting or blinking of them, possibly combined with a linguistic referring expression. The algorithm presented in this thesis can be applied to both types of systems. When applied to an embodied conversational agent it should be expected that the ECA performs referring expressions in a way that is also observed in human communication, e.g., it might include a pointing gesture to identify a target; it does not necessarily have to move to a position where it can point directly to a target, and it might also produce overspecified descriptions. In the case that the algorithm is used in a system that has a graphical interface with spoken or written output that is not supplied by an ECA, a target can be indicated by a referring expression, which is possibly overspecified and which also might be accompanied by the blinking or highlighting of the target. In the latter case, the blinking or highlighting of a target can be interpreted as a precise pointing gesture. However, one can think of domains in which less precise pointing gestures are adequate. For example, an imprecise pointing gesture to indicate the Netherlands in‘ a world map might be visualized by encircling the target as depicted in Figure 7.3 where an imprecise kind of highlighting is used to indicate the target. The application and evaluation of the multimodal algorithm in different HCI systems is to be pursued in future work.

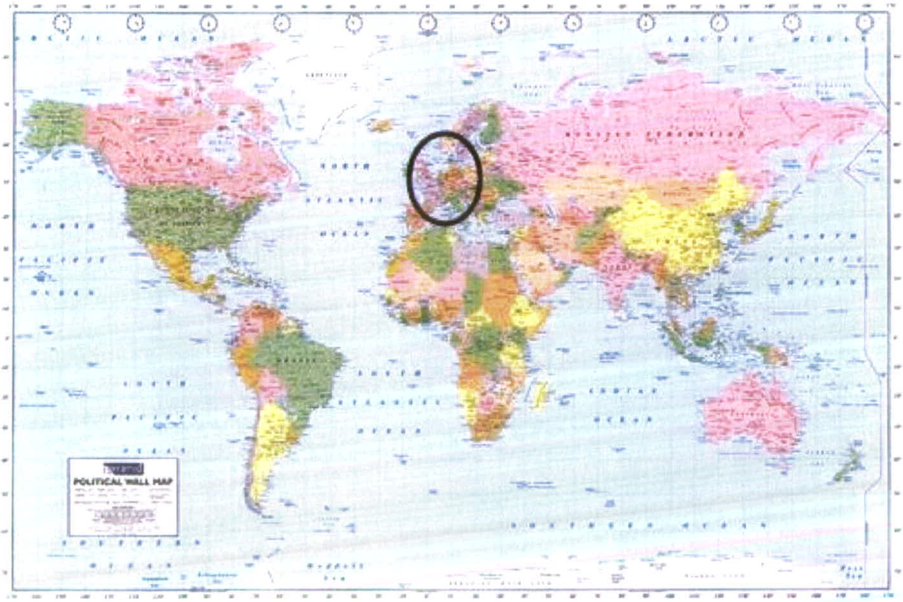


Figure 7.3: Political world map with an imprecise pointing gesture directed at the Netherlands.

Bibliography

Abeillé, A. and O. Rambow (2000). *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing.* CSLI, Stanford, USA.

Ahn, R., R. Beun, T. Borghuis, H. Bunt, and C. van Overveld (1995). The DenK architecture: A fundamental approach to user interfaces. *Artificial Intelligence Review 8*, 431–445.

Allgayer, J., R. Jansen-Winkeln, C. Reddig, and N. Reithinger (1989). Bidirectional use of knowledge in the multi-modal NL access system XTRA. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, Detroit, pp. 1492–1497.

Allwood, J. (2001). Cooperativity and flexibility in multimodal dialogue. In H. Bunt and R. Beun (Eds.), *Cooperative Multimodal Communication, Lecture Notes in Artificial Intelligence 2155*, pp. 113–124. Springer, Berlin.

Allwood, J. (2002). Bodily communication dimensions of expressions and content. In B. Granström, D. House, and I. Karlsson (Eds.), *Multimodality in Language and Speech Systems*, pp. 7–26. Kluwer Academic Publishers, Dordrecht.

Alshawi, H. (1987). *Memory and Context for Language Interpretation.* Cambridge University Press.

André, E. (2000). The generation of multimedia presentations. In R. Dale, H. Moisl, and H. Somers (Eds.), *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, pp. 305–327. Marcel Dekker Inc.

André, E. (2003). Natural language in multimedia/multimodal systems. In *Handbook of Computational Linguistics*, pp. 650–669. Oxford University Press.

André, E. and T. Rist (1993). The design of illustrated documents as a planning task. In M. Maybury (Ed.), *Intelligent Multimedia Interfaces*, pp. 94–116. AAAI Press.

André, E. and T. Rist (1996). Coping with temporal constraints in multimedia presentation planning. In *Proceedings of the 13th Conference of the American Association for Artificial Intelligence (AAAI'96)*, pp. 142–147.

André, E. and T. Rist (2000). Presenting through performing: On the use of multiple lifelike agents in knowledge-based presentation systems. In *Proceedings of the 2nd International Conference on Intelligent User Interfaces (IUI'00)*, New Orleans, USA, pp. 1–8.

André, E., T. Rist, and J. Müller. WebPersona: A life-like presentation agent for educational applications on the world-wide web. In *Proceedings of the workshop: Intelligent Educational Systems on the World Wide Web.*

André, E., T. Rist, and J. Müller (1998). WebPersona: A life-like presentation agent for the world-wide web. *Knowledge-Based Systems 11*(1), 25–36.

de Angeli, A., F. Wolff, P. Lopez, and L. Romary (1999). Relevance and perceptual constraints in multimodal referring actions. In *Proceedings of the Workshop on Deixis, Demonstration and Deictic Belief at (ESSLLI'99).*

Appelt, D. (1985). *Planning English Referring Expressions.* Cambridge University Press.

Appelt, D. and E. March (1982). Planning natural language utterances to satisfy multiple goals. Technical Note 259, SRI.

Ariel, M. (1991). The function of accessibility in a theory of grammar. *Journal of Pragmatics 16*(5), 443–463.

Ariel, M. (2001). Accessibility theory: An overview. In T. Sanders, J. Schilperoord, and W. Spooren (Eds.), *Text Representation: Linguistic and Psycholinguistic Aspects*, pp. 29–87. John Benjamins Publishing Company.

Arts, A. (2004). *Overspecification in Instructive Texts.* Ph. D. thesis, Tilburg University.

Baljko, M. (2001a). Articulatory adaptation in multimodal communicative action. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'01), Workshop on Adaptation in Dialogue Systems*, Pittsburgh PA, pp. 73–74.

Baljko, M. (2001b). The evaluation of microplanning and surface realization in the generation of multimodal acts of communication. In *Proceedings of the Workshop on Multimodal Communication and Context in Embodied Agents at the 5th International Conference on Autonomous Agents (AA'01)*, Montreal Canada, pp. 89–94.

Bangalore, S. and O. Rambow (2000a). Corpus-based lexical choice in natural language generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hongkong, China.

Bangalore, S. and O. Rambow (2000b). Exploiting a probabilistic hierarchical model for generation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, Saarbrucken, Germany.

Bateman, J. and M. Zock (2003). Natural language generation. In R. Mitkov (Ed.), *Oxford Handbook of Computational Linguistics*, pp. 284–304. Oxford university Press, London.

Bell, L., J. Boye, J. Gustafson, and M. Wirén (2000). Modality convergence in a multimodal dialogue system. In *Proceedings of Götalog 2000, 4th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 29–34.

Beun, R. (2001). On the generation of coherent dialogue. *Pragmatics and Cognition 9*(1), 37–68.

Beun, R. and H. Bunt (2001). Multimodal cooperative communication. In *Cooperative Multimodal Communication, Lecture Notes in Artificial Intelligence 2155*, pp. 1–10. Springer, Berlin.

Beun, R. and A. Cremers (1998). Object reference in a shared domain of conversation. *Pragmatics & Cognition 6*(1/2), 121–152.

Bizzi, E. and F. Mussa-Ivaldi (1990). Muscle properties and the control of arm movement. In D. Osherson, S. Kosslyn, and J. Hollerbach (Eds.), *Visual Cognition and Action*, Volume 2. MIT Press.

Bohnet, B. and R. Dale (2004). Referring expression generation as a search problem. In *Proceedings of the Australasian Language Technology Workshop (ALTW'04)*, Macquarie University, Australia.

Bos, E. (1993). *Easier Said than Done, Studies in Multimodal Human-Computer Interaction*. Ph. D. thesis, NICI, Nijmeegs Instituut voor Cognitie en Informatie.

Brennan, S. (1996). Lexical entrainment in spontaneous dialogue. In *Proceedings of International Symposium on Spoken Dialogue (ISSD'96)*, pp. 41–44.

Brøndsted, T. (1999). Reference problems in CHAMELEON. In *ESCA Tutorial and Research Workshop: Interactive Dialogue in Multimodal Systems*, Kloster Irsee, pp. 133–136.

Brøndsted, T., P. Dalsgaard, L. Bo larsen, M. Manthey, P. Mc Kevitt, T. Moeslund, and K. Olesen (1999). CHAMELEON: A general platform for performing intellimedia. In *Proceedings of the 8th International Workshop on the Cognitive Science of Natural Language Processing (CSNLP-8)*, Galway, Ireland, pp. 110–122.

Bühler, K. (1934). *Sprachtheorie: Die Darstellungsfunktion der Sprache.* Fischer, Jena.

Bunt, H. (1997). Dialogue context modelling. In *Proceedings of the International and Interdisciplinary Conference and Modelling and Using Context*, pp. 130–150.

Bunt, H. (1998). Issues in multimodal human-computer communication. In H. Bunt, R. Beun, and T. Borghuis (Eds.), *Multimodal Human-Computer Communication: Systems, Techniques and Experiments*, pp. 1–12. Springer, Berlin.

Bunt, H., R. Ahn, R. Beun, T. Borghuis, and C. van Overveld (1998). Multimodal cooperation with the DenK system. In H. Bunt, R. Beun, and T. Borghuis (Eds.), *Multimodal Human-Computer Communication. Sytems, Techniques and Experiments. Lecture Notes in Artificial Intelligence 1374*, pp. 39–67. Springer Verlag, Berlin.

Bunt, H. and W. Black (2000a). Dialogue pragmatics and context specification. In H. Bunt and W. Black (Eds.), *Abduction Belief and Context in Dialogue. Studies in Computational Pragmatics*, pp. 81–150. John Benjamins, Amsterdam.

Bunt, H. and W. Black (2000b). The ABC of computational pragmatics. In H. Bunt and W. Black (Eds.), *Abduction Belief and Context in Dialogue. Studies in Computational Pragmatics*, pp. 1–46. John Benjamins, Amsterdam.

Bunt, H. and Y. Girard (2005). Designing an open multimodal dialogue act taxonomy. In *Proceedings of the 9th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL'05)*, DIALOR'05, Nancy, France.

Bunt, H. and L. Romary (2002). Towards multimodal semantic representation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'02), Workshop on International Standards of Terminology and Language Resources Management*, Las Palmas, Spain, pp. 54–60. ELRA, Paris.

Bunt, H. and L. Romary (2004). Standardization in multimodal content representation: Some methodological issues. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Butterworth, B. and U. Hadar (1989). Gesture, speech, and computational stages: A reply to Mc Neill. *Psychological Review 96*(1), 168–174.

Byron, D. (2003). Understanding referring expressions in situated language, some challenges for real world agents. In *Proceedings of the 1st International Workshop on Language Understanding and Agents for the Real World*, Hokkaido University.

Calbris, G. (1990). *The Semiotics of French Gesture.* Indiana University Press, Bloomington.

Campana, E., M. Tanenhaus, J. Allen, and R. Remington (2004). Evaluating cognitive load in spoken language interfaces using a dual-task paradigm. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'04)*, Jeju, Korea.

Cassell, J., T. Bickmore, L. Campbell, H. Vilhjálmsson, and H. Yan (2000). Conversation as a system framework: Designing embodied conversational agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill (Eds.), *Embodied Conversational Agents*, pp. 29–63. MIT Press, Cambridge.

Cassell, J., M. Steedman, N. Badler, C. P. M. Stone, B. Douville, S. Prevost, and B. Achorn (1994). Modelling the interaction between speech and gesture. Technical report, University of Pennsylvania.

Cassell, J., M. Stone, and H. Yan (2000). Coordination and context-dependence in the generation of embodied conversation. In *Proceedings of the 1st International Conference on Natural Language Generation (INLG'00)*, pp. 171–178.

Cassell, J., J. Sullivan, S. Prevost, and E. Churchill (2000). *Embodied Conversational Agents.* MIT Press, Cambridge.

Claassen, W. (1992). Generating referring expressions in a multimodal environment. In R. Dale, E. Hovy, D. Rösner, and O. Stock (Eds.), *Aspects of Automated Natural Language Generation, Lecture Notes in Artificial Intelligence*, Volume 587, pp. 263–276. Springer Verlag, Berlin.

Claassen, W. and C. Huls (1991). DoNaLD: A dutch natural language dialogue system. Technical Report 11, SPINN/MMC Research Report, NICI, Nijmegen.

Clark, H. (2003). Pointing and placing. In S. Kita (Ed.), *Pointing, Where Language, Culture, and Cognition Meet.* Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Clark, H. and E. Clark (1977). *Psychology and Language. An Introduction to Psycholinguistics.* Harcourt Brace Javanovich, Inc., New York.

Clark, H. and J. FoxTree (2002). Using uh and um in spontaneous speaking. *Cognition 84*, 73–111.

Clark, H. and C. Marshall (1981). Definite reference and mutual knowledge. In A. Joshi, B. Webber, and I. Sag (Eds.), *Elements of Discourse Understanding*, pp. 10–63. Cambridge University Press.

Clark, H. and D. Wilkes-Gibbs (1986). Referring as a collaborative process. *Cognition 22*, 1–39.

Cohen, P. (1984). The pragmatics of referring and the modality of communication. *Computational Linguistics 10*(2), 97–146.

Cohen, P., A. Cheyer, A. Wang, and S. Baeg (1994). An open agent architecture. In *Proceedings of the American Association for Artificial Intelligence (AAAI'94) Symposium Series on Software Agents*, pp. 1–8. AAAI Press.

Cohen, P., M. Johnston, D. Mc Gee, and S. Oviatt (1998). The efficiency of multimodal interaction: A case study. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, Volume 2, pp. 249–252.

Cohen, P. and S. Oviatt (1995). The role of voice input for human-machine communication. In *Proceedings of the National Academy of Sciences*, Volume 92, pp. 9921–9927. National Academy of Sciences Press, Washington D. C.

Cremers, A. (1996). *Reference to Objects, an Empirically Based Study of Task-oriented Dialogues*. Ph. D. thesis, Eindhoven University of Technology.

Dale, R. (1988). *Generating Referring Expressions in a Domain of Objects and Processes*. Ph. D. thesis, Centre for Cognitieve Science, University of Edinburgh.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL'89)*, Vancouver BC.

Dale, R. and N. Haddock (1991). Generating referring expressions involving relations. In *Proceedings of the 5th Meeting of the European Chapter of the Association for Computational Linguistics (EACL'91)*, Berlin, Germany, pp. 161–166.

Dale, R. and E. Reiter (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science 18*, 233–263.

Dale, R. and E. Reiter (2000). *Building Natural Language Generation Sytems*. Studies in Natural Language Processing. Cambridge University Press.

van Deemter, K. (2000). Generating vague descriptions. In *Proceedings of the 1st International Conference on Natural Language Generation (ICNLG'00)*, Mitzpe Ramon, Israel.

van Deemter, K. (2001). Generating referring expressions: Beyond the incremental algorithm. In *Proceedings of the 4th International Workshop on Computational Semantics (IWCS IV)*, Tilburg, The Netherlands.

van Deemter, K. (2002). Generating referring expressions: Boolean extensions of the incremental algorithm. *Computational Linguistics 28*(1), 37–52.

van Deemter, K. and E. Krahmer (To appear). Graphs and booleans. In H. Bunt and R. Muskens (Eds.), *Computing Meaning*, Volume 3. Kluwer Academic Publishers.

van Deemter, K., E. Krahmer, and M. Theune (2005). Real vs. template-based NLG: A false opposition. *Computational Linguistics 31*(1), 15–23.

Deutsch, W. (1976). *Sprachliche Redundanz und Objekt Identifikation*. Ph. D. thesis, Unversity of Marburg.

Mc Donald, D. (1981). *Natural Language Generation as a process of Decision-Making Under Constraints*. Ph. D. thesis, MIT.

Mc Donald, D. (1992). Natural language generation. In S. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence*, pp. 983–997. J. Wiley and Sons, New York.

Duncan, S. (1972). Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology 23*(2), 286–288.

Edwards, A. (2002). Multimodal interaction and people with disabilities. In B. Granström, D. House, and I. Karlsson (Eds.), *Multimodality in Language and Speech Systems*, pp. 27–44. Kluwer Academic Publishers, Dordrecht.

Elhadad, M. (1993). *FUF the Universal Unifier, User Manual Version 5.2* (ftp://ftp.cs.bgu.ac.il/pub/people/elhadad/nlg/fufman.ps ed.). Israel: Department of Computer Science, Ben Gurion University of the Negev, Israel.

Elhadad, M. and J. Robin (1998). SURGE: A comprehensive plug-in syntactic realization component for text generation. In *http://www.cs.bgu.ac.il/research/projects/surge/*.

Eppstein, D. (1999). Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications 3*(3), 1–27.

Evans, R., P. Piwek, and L. Cahill (2002). What is NLG? In *Proceedings of the International Language Generation Conference (INLG'02)*, New York, USA, pp. 144–151.

Feyereisen, P. (1997). The competition between gesture and speech production in dual task paradigms. *Journal of Memory and Language 36*(1), 13–33.

Fitts, P. (1954). The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology 47*, 381–391.

Gaiffe, B., J. Pierrel, and L. Romary (2000). Referring in a multimodal environment: From NL to designation. In M. Taylor, F. Neel, and D. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue*, Volume II. John Benjamins Publishing Company.

Galley, M., E. Fosler-Lussier, and A. Potamianos (2001). Hybrid natural language generation for spoken dialogue systems. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech'01)*, Aalborg, Denmark.

Gardent, C. (2002). Generating minimal definite descriptions. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL'02)*, Philadelphia, USA.

Gardent, C., H. Manuélian, K. Striegnitz, and M. Amoia (2003). Generating definite descriptions: Non incrementality, inference, and data. In T. Pechmann and C. Habel (Eds.), *Multidisciplinary Approaches to Language Production*. Walter de Gruyter, Berlin.

Garey, W. and D. Johnson (1979). *Computers and Intractability: A guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.

Gibbon, D., I. Mertins, and R. Moore (2000). *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. Kluwer International Series in Engineering and Computer Science, 565. Kluwer Academic Publishers, Dordrecht.

Goldman-Eisler, F. (1968). *Psycholinguistics: Experiments in Spontaneous Speech*. Academic Press, London.

Goodman, B. (1986). Reference identification and reference identification failures. *Computational Linguistics 12*, 273–305.

Goodman, B. (1987). *Communication and Miscommunication*. Association of Computational Linguistics Series of Cambridge University Press. Cambridge University Press.

Goodwin, C. (1981). *Conversational Organization: Interaction between Speakers and Hearers*. Academic Press, New York.

Grice, H. (1975). Logic and conversation. In P. Cole and J. Morgan (Eds.), *Syntax and Semantics 3: Speech Acts*, pp. 41–58. Academic Press, New York.

Grosz, B., A. Joshi, and S. Weinstein (1995). Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics 21*(2), 203–225.

Grosz, B. and C. Sidner (1986). Attention, intention and the structure of discourse. *Computational Linguistics 12*, 175–206.

Gundel, J., N. Hedberg, and R. Zacharski (1993). Cognitive status and the form of referring expressions in discourse. *Language 69*(2), 274–306.

Gupta, A. and T. Anastasakos (2004). Integration patterns during multimodal interaction. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'04)*, Jeju, Korea.

Hajičová, E. (1993). Issues of sentence structure and discourse patterns. In *Theoretical and Computational Linguistics*, Volume 2. Charles University Prague.

Haviland, J. (2003). How to point in zinacantan. In S. Kita (Ed.), *Pointing, where Language, Culture and Cognition Meet*, pp. 109–137. Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Heritage, J. (1984). A change of state token and aspects of its sequential placement. In M. Atkinson and J. Heritage (Eds.), *Structures of Social Action: Studies in Conversational Analysis*, pp. 299–345. Cambridge University Press.

Hintikka, J. (1998). Perspectival identification, demonstratives and 'small worlds'. *Synthese 114*, 203–232.

Horacek, H. (1995). More on generating referring expressions. In *Proceedings of the 5th European Workshop on Natural Language Generation (EWNLG'95)*, Leiden, The Netherlands, pp. 43–58.

Horacek, H. (1997). An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics (ACL'97)*, Madrid, Spain, pp. 206–213.

Horacek, H. (2003). A best-first search algorithm for generating referring expressions. In *Research Notes of the 10th Meeting of the European Chapter of the Association for Computational Linguistics, Conference Companion (EACL'03)*, Budapest, Hungary, pp. 103–106.

Horacek, H. (2004). On referring to sets of objects naturally. In *Proceedings of the 3th International Conference on Natural Language Generation (INLG'03)*, Brockenhurst, UK, pp. 70–79.

Horacek, H. (2005). Generating referential descriptions under conditions of uncertainty. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG'05)*, Aberdeen, Scotland.

Huls, C., W. Claassen, and E. Bos (1995). Automatic referent resolution of deictic and anaphoric expressions. *Computational Linguistics 21*(1), 59–79.

Hutchins, E., J. Holland, and D. Norman (1986). Direct manipulation interfaces. In D. Norman and S. Draper (Eds.), *User Centered System Design: New Perspectives on Human Computer Design*. Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Jordan, P. (1999). An empirical study of the communicative goals impacting nominal expressions. In *Proceedings of the ESSLLI Workshop on the Generation of Nominal Expressions (ESSLLI'99)*.

Jordan, P. (2000). Influences on attribute selection in redescriptions: A corpus study. In *Proceedings of the Cognitive Science Conference (COGSCI'00)*, Philadelphia, Pennsylvania, USA, pp. 250–255.

Jordan, P. (2002). Contextual influences on attribute selection for repeated descriptions. In K. van Deemter and R. Kibble (Eds.), *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, pp. 295–328. CSLI Publications, Stanford.

Jörding, T. and I. Wachsmuth (2002). An anthropomorphic agent for the use of spatial language. In K. Coventry and P. Olivier (Eds.), *Spatial Language: Cognitive and Computational Aspects*, pp. 69–86. Kluwer Academic Publishers, Dordrecht.

Karttunen, L. (1976). Discourse referents. In J. Mc Cawley (Ed.), *Syntax and Semantics*, Volume 2: Notes from the Linguistic Underground, pp. 363–386. New York Academic Press.

Kato, T. and Y. Nakano (1997). Towards generation of fluent referring action in multimodal situations. In *Proceedings of the Workshop on Referring Phenomena in a Multimedia Context and their Computational Treatment.*, pp. 20–27. ACL SIGMedia.

Kehler, A., J. Martin, A. Cheyer, L. Julia, J. Hobbs, and J. Bear (1998). On representing salience and reference in multimodal human-computer interaction. In *Proceedings of the American Association for Artificial Intelligence (AAAI'98) Workshop on Representations for Multi-modal Human-Computer Interaction*, Madison.

Kelleher, J., F. Costello, and J. van Genabith (2005). Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. In E.Reiter and D. Roy (Eds.), *Special Issue of Artificial Intelligence Journal on Connecting Language to the World*, Volume 167. Elsevier,.

Kelleher, J. and J. van Genabith (2003). A false colouring real time visual saliency algorithm for reference resolution in simulated 3-d environments. In *Proceedings of the Conference on Artifical Intelligence and Cognitive Science (AICS'03).*

Kempen, G. and E. Hoenkamp (1987). An incremental procedural grammar for sentence production. *Cognitive Science 11*, 201–258.

Kendon, A. (1972). Some relations between body motion and speech. In A. Siegman and B. Pope (Eds.), *Studies in dyadic communication.* Pergamon Press.

Kendon, A. (1980). Gesticulation and speech: Two aspects of the process of utterance. In M. Key (Ed.), *The Relationship of Verbal and Nonverbal Communication*, pp. 207–228. Mouton, The Hague.

Kendon, A. (1994). Do gestures communicate? A review. *Research on Language and Social Interaction 27*(3).

Kendon, A. (2004). *Gesture, Visible Actions as Utterance.* Cambridge University Press.

Kendon, A. and L. Versante (2003). Pointing by the hand in neapolitan. In S. Kita (Ed.), *Pointing, where Language, Culture and Cognition Meet*, pp. 109–137. Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Kettebekov, S., M. Yeasin, and R. Sharma (2002). Prosody based co-analysis for continuous recognition of coverbal gestures. In *Proceedings of the 4th International Conference on Multimodal Interfaces (ICMI'02)*, pp. 161–166.

Mc Kevitt, P. (1998). Chameleon meets spatial cognition. In S. O' Nuallain (Ed.), *Spatial Cognition*, pp. 149–170. John Benjamins Publishing Company.

Kievit, L., P. Piwek, R. Beun, and H. Bunt (2001). Multimodal cooperative resolution of referential expressions in the denk system. In H. Bunt and R. Beun (Eds.), *Cooperative Multimodal Communication, Lecture Notes in Artificial Intelligence 2155*, pp. 197–214. Springer Verlag, Berlin.

Kita, S. (1990). *The Temporal Relationship between Gesture and Speech: A study of Japanese-English Bilinguals.* MS, Department of Psychology, University of Chicago.

Kita, S. (1993). *Language and Thought Interface: A Study of Sponteneous Gestures and Japanese Mimetics.* Ph. D. thesis, University of Chicago.

Kita, S. and A. Özyürek (2003). What does cross-linguistic variation in semantic coordination of speech and gesture reveal?: Evidence for an interface representation of spatial thinking and gesture. *Journal of Memory and Language 48*, 16–32.

Kopp, S., B. Jung, N. Lemann, and I. Wachsmuth (2003). Max – a multimodal assistant in virtual reality construction. *KI, Knstliche Intelligenz, special Issue on Embodied Conversational Agents 4*, 11–17.

Kopp, S., P. tepper, and J. Cassell (2004). Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proceedings of the 6th International Conference on Multimodal Interfaces (ICMI'04)*, State College, PA, USA, pp. 97–104.

Kopp, S. and I. Wachsmuth (2002). Model-based animation of coverbal gesture. In *Proceedings of Computer Animation*, Los Alamitos, CA, pp. 252–257. IEEE Press.

Krahmer, E., S. van Erk, and A. Verleg (2001). A meta-algorithm for the generation of referring expressions. In *Proceedings of the 8th European Workshop on Natural Language Generation (EWNLG'01)*, Toulouse.

Krahmer, E., S. van Erk, and A. Verleg (2003). Graph-based generation of referring expressions. *Computational Linguistics 29*(1), 53–72.

Krahmer, E. and I. van der Sluis (2003). A new model for generating multimodal referring expressions. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG'03)*, Budapest, Hungary, pp. 47– 54.

Krahmer, E. and M. Theune (1998). Context sensitive generation of referring expressions. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, Sydney, pp. 1151–1154.

Krahmer, E. and M. Theune (2002). Efficient context-sensitive generation of referring expressions. In K. van Deemter and R. Kibble (Eds.), *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, pp. 223–264. CSLI Publications, Stanford.

Kranstedt, A., P. Kühnlein, and I. Wachsmuth (2003). Deixis in multimodal human computer interaction: An interdisciplinary approach. In *Proceedings of the 5th International Gesture Workshop (GW'03)*, Genova, Italy, pp. 112–123.

Kranstedt, A., A. Lücking, T. Pfeiffer, H. Rieser, and I. Wachsmuth (2005). Deixis: How to determine demonstrated objects. In *Presented at the 5th International Gesture Workshop (GW'05)*, Ile de Berder, France.

Kranstedt, A., A. Lücking, T. Pfeiffer, H. Rieser, and I. Wachsmuth (To appear in 2006). Deictic object reference in task-oriented dialogue. In G. Rickheit and I. Wachsmuth (Eds.), *Situated Communication*. Mouton de Gruiter.

Kranstedt, A. and I. Wachsmuth (2005). Incremental generation of multimodal deixis referring to objects. In *Proceedings of the 10th European Workshop On Natural Language Generation (ENLG'05).*

Kraus, R., P. Morrel-Sante, and C. Colasante (1991). Do conversational hand gestures communicate? *Journal of Personality and Social Psychology 61*(5), 743–754.

Krauss, R., Y. Chen, and P. Chawla (1996). Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us? In M. Zanna (Ed.), *Advances in Experimental Social Psychology*, Volume 28, pp. 389–450. Academic Press, Tampa.

Krauss, R., Y. Chen, and R. Gottesman (2000). Nonverbal behavior and nonverbal communication: A process model. In D. Mc Neill (Ed.), *Language and Gesture*, pp. 261–283. Cambridge University Press.

Krenn, B., M. Grice, S. Baumann, P. Piwek, K. van Deemter, M. Schröder, M. Klesen, and E. Gstrein (2002). Generation of multimodal dialogue for net environments. In *Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2002)*, Saarbrücken, Germany, pp. 91–98.

Kumar, S. and P. Cohen (2000). Towards a fault-tolerant multi-agent system architecture. In *Proceedings of the 4th International Conference on Autonomous Agents*, pp. 456–466. ACM Press, Barcelona, Spain.

Landragin, F., N. Bellalem, and L. Romary (2001). Visual salience and perceptual grouping in multimodal interactivity. In *Proceedings of the 1st International Workshop on Information Presentation and Natural Multimodal Dialogue*, Verona, Italy, pp. 151–155.

Landragin, F., A. Denis, A. Ricci, and L. Romary (2004). Multimodal meaning representation for generic dialogue systems architectures. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04).*

Langkilde, I. and K. Knight (1998). The practical value of n-grams in generation. In *Proceedings of the 9th International Workshop on Natural Language Generation (INLG'98)*, Niagara-on-the-lake, Ontario, pp. 248–255.

Lester, J. and B. Stone (1997). Increasing believability in animated pedagogical agents. In *Proceedings of the 1st International Conference on Autonomous Agents*, Marina del Rey, California, pp. 16–21.

Lester, J., J. Voerman, S. Towns, and C. Callaway (1997). COSMO: A life-like animated pedagogical agent with deictic believability. In *Workshop on Animated Interface Agents: Making Them Intelligent (IJCAI'97)*, Nagoya, Japan, pp. 61–69.

Lester, J., J. Voerman, S. Towns, and C. Callaway (1999). Deictic believability: Coordinating gesture, locomotion and speech in lifelike pedagogical agents. *Applied Artificial Intelligence 13*(4-5), 383–414.

Levelt, W. (1989). *Speaking: From Intention to Articulation.* MIT Press, Cambridge.

Levelt, W., H. Vorberg, and W. La Heij (1985). Pointing and voicing in deictic expressions. *Journal of Memory and Language 24*, 133–164.

Levin, E., S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker (2000). The AT&T - DARPA communicator mixed-initiative spoken dialogue system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'00)*, Beijing, China, pp. 122–125.

Levinson, S. (1983). *Pragmatics.* Cambridge University Press.

Liebers, A. (2001). Planarizing graphs. *Journal of Graph Algorithms and Applications 5*(1), 1–74.

van Linden, K. (2000). Natural language generation. In D. Jurafsky and J. Martin (Eds.), *Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing*, pp. 763–796. Prentice Hall, New Jersey.

Lücking, A., H. Rieser, and J. Stegmann (2004). Statistical support for the study of structures in multimodal dialogue: Inter-rater agreement and synchronization. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue, (Catalog'04)*, Barcelona, Spain, pp. 93–100.

MacKenzie, I. (1991). *Fitts' Law as a Performance Model in Human-computer Interaction.* Ph. D. thesis, University of Toronto, Canada.

Maes, A., A. Arts, and L. Noordman (2004). Reference management in instructive discourse. *Discourse Processes 37*, 117–144.

Malouf, R. (2000). The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong, pp. 85–92.

Mangold, R. and R. Pobel (1988). Informativeness and instrumentality in referential communication. *Journal of Language and Social Psychology 7*(3-4), 181–191.

Mann, W. and S. Thompson (1987). Rethorical structure theory: A theory of text organization. Technical Report ISI RS-87-190, University of Southern California, Marina Del Rey, CA.

Martin, D., A. Cheyer, and D. Moran (1999). The open agent architecture: A framework for building distributed software systems. *Applied Artificial Intelligence 13*, 91–128.

Martin, J., R. Veldman, and D. Béroule (1998). Developing multimodal interfaces: A theoretical framework and guided propagation networks. In H. Bunt, R. Beun, and T. Borghuis (Eds.), *Multimodal Human-Computer Communication: Systems, Techniques and Experiments*, pp. 158–187. Springer, Berlin.

Matsui, T. (1998). Pragmatic criteria for reference assignment: A relevance-theoretic account of the acceptability of bridging. *Pragmatics and Cognition 6*((1/2)), 47–97.

Maybury, M. (1993). *Intelligent Multimedia Interfaces.* AAAI/MIT Press, Menlo Park.

Maybury, M. (2000). Communicative acts for multimedia and multimodal dialogue. In M. Taylor, F. Néel, and D. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue II*, pp. 375–392. John Benjamins, Amsterdam.

Maybury, M. and J. Lee (2000). Multimedia and multimodal interaction structure. In M. Taylor, F. Néel, and D. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue II*, pp. 295–308. John Benjamins, Amsterdam.

Messmer, B. and H. Bunke (1995). Subgraph isomorphism in polynomial time. Technical Report IAM 95-003, University of Bern, Institute of Computer Science and Applied Mathematics, Bern, Switserland.

Messmer, B. and H. Bunke (1998). A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*(5), 493–504.

Metzing, C. and S. Brennan (2003). When conceptual pacts are broken: Partner-specific effects on the comprehension of referring expressions. *Journal of Memory and Language 49*, 201–213.

ter Meulen, A. (1994). Demonstratives indications and experiments. *The Monist 77*(2), 239–256.

Muskens, R. (2001). Talking about trees and truth conditions. *Journal of Logic, Language and Information*, 417–455.

Neal, J. and S. Shapiro (1988). Intelligent multi-media interface technology. In *Proceedings of the Workshop on Architectures of Intelligent Interfaces: Elements & Prototypes*, pp. 69–91.

Neal, J. and S. Shapiro (1991). Intelligent multimedia interface technology. In J.Sullivan and S.Tyler (Eds.), *Intelligent User Interfaces*, pp. 11–43. ACM Press, New York.

Neal, J., C. Thielman, Z. Dobes, S. Haller, and S. Shapiro (1998). Natural language with integrated deictic and graphic gestures. In *Readings in Intelligent User Interfaces*, pp. 37–52. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Mc Neill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, London, Chicago.

Mc Neill, D., E. Levy, and L. Pedelty (1990). Gestures and speech. In G. Hammond (Ed.), *Advances in Psychology: Cerebral Control of Speech and Limb Movements*, pp. 203–256. Elsevier/ North Holland Publishers, Amsterdam.

Mc Neill, D., F. Quek, K. Mc Cullough, S. Duncan, R. Bryll, X. Ma, and R. Ansar (2002). Dynamic imagery in speech and gesture. In B. Granström, D. House, and I. Karlsson (Eds.), *Multimodality in Language and Speech Systems*, pp. 27–44. Kluwer Academic Publishers, Dordrecht.

Nigay, L. and J. Coutaz (1993). A design space for multimodal systems: Concurrent processing and data fusion. In *Proceedings of the Conference on Human Factors in Computing Systems (Interchi'93)*, pp. 172–178.

Nijholt, A. and D. Heylen (2002). Multimodal communication in inhabited virtual environments. *International Journal of Speech Technology 5*, 343–354.

Noser, H., O. Renault, D. Tahlmann, and N. Magnenat-Thalmann (1995). Navigation for digital actors based on synthetic vision, memory and learning. *Computer Graphics 19*(1), 7–9.

Oh, A. and A. Rudnicky (2000). Stochastic language generation for spoken dialogue systems. In *Proceedings of the ANLP/NAACL 2000 Workshop on Conversational Systems*, Seattle, pp. 27–32.

Oviatt, S. (1997). Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction (Special Issue on Multimodal Interfaces) 12*, 93–129.

Oviatt, S. (1999). Ten myths of multimodal interaction. *Communications of the ACM 42*(11), 74–81.

Oviatt, S. (2003). Multimodal interfaces. In J. Jacko and A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Oviatt, S., A. de Angeli, and K. Kuhn (1997). Integration and synchronization of inputmodes during multimodal human-computer interaction. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'97)*, ACM Press, New York, pp. 415–422.

Oviatt, S. and P. Cohen (1991). Discourse structure and performance efficiency in interactive and noninteractive spoken modalities. *Computer Speech and Language 5*(4), 297–326.

Oviatt, S., P. Cohen, and M. Wang (1994). Toward interface design for human language technology: Modality and structure as determants of linguistic complexity. *Speech Communication 15*, 283–300.

Oviatt, S., R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson, and L. Carmichael (2003). Toward a theory of organized multimodal integration patterns during human-computer interaction. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, Vancouver, British Columbia, Canada, pp. 44–51.

Oviatt, S. and K. Kuhn (1998). Referential features and linguistic indirection in multimodal language. In *Proceedings of the International Conference on Spoken Language Processing*, Volume 6, ASSTA INC., Sydney, Australia, pp. 2339–2342.

Passonneau, R. (1996). Using centering to relax Gricean informational constraints on discourse anaphoric noun phrases. *Language and Speech 39*(2-3), 229–264.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics 27*, 98–110.

Pelachaud, C., V. Carofiglio, B. de Carolis, F. de Rosis, and I. Poggi (2002). Embodied contextual agent in information delivering application. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS'02)*, Bologna, Italy, pp. 758–765.

Piwek, P. and R. Beun (2001). Multimodal referential acts in a dialogue game. From empirical investigations to algorithms. In *Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue (IWIPNMD'01)*, Verona, Italy, pp. 127–131.

Ratnaparkhi, A. (2002). Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer, Speech & Language 16*(3/4), 435–455.

Reeves, B. and C. Nass (1996). *The Media Equation, How People Treat Computers, Television and New Media Like Real People and Places.* CSLI Publications.

Reiter, E. (1991). A new model of lexical choice for nouns. *Computational Intelligence 7(4)*, 240–251.

Reiter, E. and C. Mellish (1992). Using classification to generate text. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, pp. 265–272.

Reithinger, N. (1992). The performance of an incremental generation component for multi-modal dialog contributions. In R. Dale, E. Hovy, D. Rösner, and O. Stock (Eds.), *Aspects of Automated Natural Language Generation, Lecture Notes in Artificial Intelligence*, Volume 587, pp. 263–276. Springer Verlag, Berlin.

Reithinger, N., J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pfleger, P. Poller, M. Streit, and V. Tschernomas (2003). Smartkom - adaptive and flexible multimodal access to multiple applications. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI'03)*, Vancouver.

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

Rieser, H. (2004). Pointing in dialogue. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog'04)*, Barcelona, Spain, pp. 93–100.

de Ruiter, J. (1998). *Gesture and Speech Production.* Ph. D. thesis, Katholieke Universiteit Nijmegen.

de Ruiter, J. (2000). The production of gesture and speech. In D. Mc Neill (Ed.), *Language and Gesture.* Cambridge University Press.

Sacks, H. (1992). *Lectures on Conversation*, Volume II. Blackwell Publishers.

Salmon-Alt, S. and L. Romary (2000). Generating referring expressions in multimodal contexts. In *Proceedings of the International Workshop on Natural Language Generation (INLG'00), Workshop on Coherence in Generated Multimedia*, Mitzpe Ramon, Israel.

Schegloff, E. (1984). On some gestures' relation to talk. In J. Atkinson and J. Heritage (Eds.), *Structures in Social Action: Studies in Conversation Analysis*, pp. 266–296. Cambridge University Press.

Schmauks, D. and N. Reithinger (1988). Generating multimodal output-conditions, advantages and problems. In *Proceedings of the 12th conference on Computational linguistics (COLING'88)*, Volume 2, Budapest, Hungary, pp. 584 – 588.

Searle, J. (1969). *Speech Acts.* Cambridge University Press.

Sharma, R., J. Cai, S. Chakravarthy, I. Poddar, and Y. Sethi (2000). Exploiting speech/gesture co-occurrence for improving continuous gesture recognition in weather narration. In *Proceedings of the International Conference on Face and Gesture Recognition (FG'00)*, Grenoble, France.

Shaw, J. and V. Hatzivassiloglou (1999). Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL'99)*, College Park, Maryland, pp. 135–143.

Skantze, G. (2003). Coordination of referring expressions in multimodal human-computer dialogue. In *Proceedings of the International Conference on Spoken Language Processing.*

van der Sluis, I. (2001). An empirically motivated algorithm for the generation of multimodal referring expressions. In *Proceedings of the Student Research Workshop of the 39th Annual Meeting of the Association of Computational Linguistics (ACL'01)*, Toulouse, France, pp. 67–72.

van der Sluis, I. and E. Krahmer (2001). Generating referring expressions in a multimodal context. In *Selected Papers of the 11th CLIN Meeting*, pp. 158–176. Rodopi, Amsterdam.

van der Sluis, I. and E. Krahmer (2004a). Evaluating multimodal NLG using production experiments. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

van der Sluis, I. and E. Krahmer (2004b). The influence of target size and distance on the production of speech and gesture in multimodal referring expressions. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'04)*, Jeju, Korea.

van der Sluis, I. and E. Krahmer (2005). Towards the generation of overspecified multimodal referring expressions. In *Proceedings of the Symposium on Dialogue Modelling and Generation at the 15th Annual meeting of the Society for Text and Discourse*, Amsterdam, The Netherlands.

Smyth, M. and A. Wing (1984). *The Psychology of Human Movement.* Academic Press, New York.

Sonnenschein, S. (1982). The effects of redundant communication on listeners: When more is less. *Child Development 53*, 717–729.

Sonnenschein, S. (1984). The effect of redundant communication on listeners: Why different types may have different effects. *Journal of Psycholinguistic Research 13*, 147–166.

Soudzilovskaia, N. and F. Jansen (2001). Visual presentation and interaction in a multi-modal user interface. TWAIO-report, Computer Graphics and CAD/CAM group, Information Technology and Systems, Delft University of Technology.

Sowa, T., S. Kopp, and M. Latoschik (2001). A communicative mediator in a virtual environment: Processing of multimodal input and output. In *Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue (IPNMD'01)*, pp. 71–74.

Sowa, T. and I. Wachsmuth (2001). Coverbal iconic gestures for object descriptions in virtual environments: An empirical study. Technical Report SFB360, Collaborative Research Center Situated Artificial Communicators, University of Bielefeld.

Stent, A. (1998). Aspects of natural language generation. Technical Report TR701, Computer Science Department University of Rochester.

Stone, B. and J. Lester (1996). Dynamically sequencing an animated pedagogical agent. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, pp. 424–431.

Stone, M. (1999). Describing sets with covers and sets of ordinary assignments. In R. Kibble and K. van Deemter (Eds.), *Workshop on the Generation of Nominal Expressions.*

Stone, M. (2000). On identifying sets. In *Proceedings of the 1st International Conference on Natural Language Generation (ICNLG'00)*, Mitzpe Ramon, Israel.

Stone, M., C. Doran, B. Webber, T. Bleam, and M. Palmer (2003). Microplanning with communicative intentions: The SPUD system. *Computational Intelligence 19*(4), 311–381.

Tenbrink, T. (2004). Identifying objects on the basis of spatial contrast: An empirical study. In *Spatial Cognition IV. Reasoning, Action, and Interaction: International Conference Spatial Cognition 2004*, Frauenchiemsee, Germany. Springer-Verlag GmbH.

Theune, M. (2000). *From Data to Speech, Language Generation in Context.* Ph. D. thesis, Eindhoven University of Technology.

Theune, M. (2001). ANGELICA: Choice of output modality in an embodied agent. In *International Workshop on Information Presentation and Natural Multimodal Dialogue (IPNMD'01)*, Verona, Italy, pp. 89–94.

Theune, M. (2003). From monologue to dialogue: Natural language generation in OVIS. In *Proceedings of the American Association for Artificial Intelligence (AAAI'03) Spring Symposium on Natural Language Generation in Written and Spoken Dialogue*, Palo Alto, CA, pp. 141–150.

Theune, M., D. Heylen, and A. Nijholt (2005). Generating embodied information presentations. In O. Stock and M. Zancanaro (Eds.), *Multimodal Intelligent Information Presentation*, pp. 47–70. Kluwer Academic Publishers.

Thorisson, K. (1994). Simulated perceptual grouping: An application to human computer interaction. In *Proceedings of the 16th Annual Conference of Cognitive Science Society*, Atlanta, pp. 876–881.

Varges, S. (2004). Overgenerating referring expressions involving relations. In *Proceedings of the 3rd International Conference on Natural Language Generation (INLG'04)*, pp. 171–181.

Varges, S. and K. van Deemter (2005). Generating referring expressions containing quantifiers. In *Proceedings of the 6th International Workshop on Computational Semantics (IWCS-6).*

Vernier, F. and L. Nigay (2000). A framework for the combination and characterization of output modalities. In *Proceedings of the 7th Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'00)*, Limerick, Ireland, pp. 32–48.

Wachsmuth, I. (1999). Communicative rhythm in gesture and speech. In *Proceedings of the 2nd International Gesture Workshop (GW'99)*, pp. 277–289. Springer, Berlin.

Wachsmuth, I. and S. Kopp (2001). Lifelike gesture synthesis and timing for conversational agents. In *Proceedings of the 3rd International Gesture Workshop (GW'01)*, pp. 120–133.

Wahlster, W. (2002). SmartKom: Fusion and fission of speech, gestures, and facial expressions. In *Proceedings of the 1st International Workshop on Man-Machine Symbiotic Systems*, Kyoto, Japan, pp. 213–225.

Wahlster, W. (2003a). SmartKom: Symmetric multimodality in an adaptive and reusable dialogue shell. In *Proceedings of the Human Computer Interaction Status Conference*, Berlin, Germany, pp. 47–62.

Wahlster, W. (2003b). Towards symmetric multimodality: Fusion and fission of speech, gesture, and facial expression. In A. Günter and B. N. R. Kruse (Eds.), *KI 2003: Advances in Artificial Intelligence*, pp. 1–18. Springer, Berlin, Heidelberg.

Wahlster, W., N. Reithinger, and A. Blocher (2001). SmartKom: Multimodal communication with a life-like character. In *Proceedings of Eurospeech*, Aalborg, Denmark.

Walker, M. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research 12*, 387–416.

Wilkins, D. (2003). Why pointing with the index finger is not a universal (in socio-cultural terms). In S. Kita (Ed.), *Pointing, where Language, Culture and Cognition Meet*, pp. 109–137. Lawrence Erlbaum Associates Publishers, Manwah, New Jersey, London.

Wilson, D. (1992). Reference and relevance. *UCL Working Papers in Linguistics 4*, 167–191.

Wilson, D. and D. Sperber (1984). On choosing the context for utterance interpretation. In *Foregrounding Background*. Doxa, Lund.

Wolff, F., A. de Angeli, and L. Romary (1998). Acting on a visual world: The role of perception in multimodal HCI. In *Proceedings of the American Association for Artificial Intelligence (AAAI'98), Workshop on Multimodal Representation*, Madison.

Zipf, G. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison Wesley Publishing Company, Cambridge.

Zoltan-Ford, E. (1991). How to get people to say and type what computers can understand. *International Journal of Man Machine Studies 34*(4), 527–547.

Summary

Advances in human-computer interaction (HCI) provide evidence that the use of multiple modalities, such as for instance speech and gesture, in both the input and the output, will result in systems that are more robust and efficient to use than systems that interact with only one modality (Oviatt, 1999). In this thesis the focus is on multimodal output generation. A task that is addressed in many multimodal systems is that of identifying a certain object in a visual context accessible to both user and system. This can be done for example by blinking or highlighting the object, or by using an Embodied Conversational Agent (ECA) that points to the object, possibly in combination with a linguistic referring expression. Characteristically, this implies the coordinated generation of language and gesture. With the design of more advanced application systems, not only does the question arise about how such systems should generate descriptions in which linguistic information and gestures are combined, but also about how such multimodal referring expressions are produced by humans. The research that is presented in this thesis focuses on two aspects of the need for more advanced multimodal presentations: (1) In what way is the generation of multimodal utterances directed by the context? (see Bunt (1997), Bunt and Black (2000a) and Bunt and Girard (2005) for an elaborated notion of context) and (2) Which factors determine what modality or combination of modalities to use in what conditions? Preceded by a critical discussion of algorithms for the generation of referring expressions that have been proposed before, a new algorithm for the generation of multimodal referring expressions is presented. The algorithm is based on findings in human communication, where referring expressions which include pointing gestures are rather common (Beun and Cremers, 1998). The output of the algorithm is founded on three important notions that underly the human production of referring expressions: salience, effort and certainty.

The algorithm generates linguistic referring expressions (used on their own or in combination with a pointing gesture) based on a three-dimensional notion of **salience**, which acknowledges the linguistic context and the perceptual context of the interaction. To determine the linguistic context, the discourse history with a notion of recency is taken into account. The perceptual context is determined by

two factors: (1) the inherent salience of certain objects, that stand out because they have a particular property that is not present in the rest of the domain; and (2) the visual focus of attention, which centers around the last mentioned target in the discourse, where the scope of possibly generated pointing gestures is incorporated as well. Another important factor is the *principle of minimal effort* (Clark and Wilkes-Gibbs, 1986), which states that in cooperative dialogue a speaker tries to minimize both her own and the hearer's **effort**. Consequently a speaker's goal is to make identification by the hearer as easy as possible by providing enough but not too much information, while at the same time minimizing her own effort in producing the referring expression. Besides balancing the amount of information, the principle determines the kind of information that is used as well: in some cases a pointing gesture is the optimal way to refer to an object, whereas in others a linguistic description is more appropriate, or a combination of the two. The third factor that plays a role in object identification is the speaker's objective of making sure that the hearer can interpret the referring expression. This notion is formalized in the *principle of distant responsibility* (Clark and Wilkes-Gibbs, 1986), which says that a speaker must be **certain** that the information provided in an utterance is understandable to the addressee. Correspondingly, the speaker might be tempted to overspecify a referring expression or use a very precise pointing gesture, in order to gain certainty on correct identification by the hearer.

The model for pointing presented in this thesis provides for a close coupling between the linguistic information and pointing gestures used. The algorithm in which this model is formalized generates various pointing gestures, precise and imprecise ones. The type of pointing gesture is closely linked to the perceptual context in that the scope of an imprecise pointing gesture contains more objects than the scope of a precise pointing gesture. A direct consequence of this model for pointing is that the amount of linguistic properties required to generate a distinguishing multimodal referring expression is predicted to co-vary with the kind of pointing gesture used. The model for pointing is implemented in a multimodal extension of a new algorithm for the generation of referring expressions. This algorithm, proposed by Krahmer et al. (2003), approaches the generation of referring expressions as a graph construction problem using subgraph isomorphism. The decision to point is made on the basis of cost functions which are grounded in a fundamental law about the human motor system (Fitts, 1954). The output of the algorithm is based on a trade-off between the costs of a pointing gesture and the costs of the linguistic information needed to single out a target object. As such, referring expressions are generated with respect to a notion of effort, which balances the kind of information that should be presented in order to identify the target at the lowest cost.

The algorithm is evaluated using production experiments in which participants identify items by speech and gesture. In this way, spontaneous multimodal data is gathered on controlled input. This thesis presents a report of two studies in which participants refer to objects that differ in shape, size and color. One study has a very strict setting; pointing is forced and no feedback is given. The other study is performed in a more natural and interactive setting. The participants in the two studies are divided into two groups: one group located close to the object domain (i.e., the subjects can touch the targets by using precise pointing gestures) and one group located further away (i.e., the subjects can only use pointing gestures that vaguely indicate the location of the target). The data resulting from the two studies shows some clear differences with respect to the kind of target objects that were referred to and with respect to the distance from the object domain at which the speakers were located. When located near to the object domain, most speakers reduce the linguistic material almost to zero (i.e., a precise pointing gesture suffices in such cases), whereas when located further away from the object domain, subjects tend to produce more overspecified descriptions (i.e., an imprecise pointing gesture is too vague). From a detailed analysis of the data resulting from both studies, it is concluded that the co-variation of the linguistic material and the kind of pointing gesture corresponds well with the output of the algorithm. However, the algorithm, generating only minimal descriptions, makes different predictions when it comes to overspecification.

In order to generate overspecified referring expressions similar to the ones occurring in human communication, a detailed survey of both unimodal and multimodal overspecification is carried out with respect to the data resulting from the production experiments as well as findings in cognitive linguistics (e.g., Pechmann, 1989; Arts, 2004; Maes et al., 2004). Two questions are considered: (1) Why and when do speakers overspecify? and (2) How do speakers overspecify? From research in cognitive linguistics it can be inferred that the occurrence of overspecification in human communication is due to an uncertainty on the side of the speaker about the hearer being able to interpret the referring expression. This finding is confirmed by the analysis of overspecification in the data resulting from the studies mentioned above. Subsequently, the graph-based algorithm presented in this thesis is adapted in such a way that overspecified referring expressions can be generated on the basis of an estimation of the likelihood that a user will be able to correctly interpret the referring expression in the current context. Both the pointing gestures and the linguistic information that can be included in a referring expression are enriched with certainty scores that estimate their effect on the referring expression as a whole in terms of certainty. The degree of overspecification necessary in any particular situation is based on discourse and context factors. As a result the algorithm selects linguistic information and pointing gestures by balancing their costs and certainty scores, in order to find the referring expression that satisfies the responsibility to make sure that the user can identify the target at the lowest cost. In this way a wide range of referring expressions can be generated, from minimal to highly overspecified ones.

To conclude, in contrast to earlier algorithms for the generation of referring expressions, the algorithm proposed in this thesis generates multimodal (possibly overspecified) referring expressions in a context-sensitive way, on the basis of a limited number of independently motivated contraints related to costs and certainty.

Samenvatting

De vooruitgang in het onderzoek naar mens-computer interactie bevestigt dat het gebruik van meerdere modaliteiten in zowel de input als de output zal resulteren in meer robuuste en efficiëntere systemen vergeleken met systemen waarbij de interactie met de gebruiker verloopt in slechts één modaliteit (Oviatt, 1999). Dit proefschrift richt zich op de generatie van multimodale output. Een taak die uitgevoerd moet worden in veel multimodale systemen is het identificeren van een bepaald object in een visuele context, die zowel voor de gebruiker als voor het systeem toegankelijk is. Zo'n identificatie kan plaatsvinden door bijvoorbeeld het object op te lichten (highlighting) of te laten knipperen (blinking) of door het gebruik van een Embodied Conversational Agent (ECA), die het object aanwijst eventueel in combinatie met een linguïstische referentiële uiting. Het gebruik van meerdere modaliteiten heeft tot gevolg dat in het generatieproces taal en gebaren gecoördineerd moeten worden. Het ontwerpen van meer geavanceerde systemen roept de vraag op hoe deze systemen acties moeten genereren, die de informatie van gebaren en linguïstische informatie combineren. Tegelijkertijd wordt het interessant om te weten hoe zulke multimodale referentiële uitingen door mensen worden geproduceerd. Het onderzoek in dit proefschrift concentreert zich op twee aspecten van de behoefte aan geavanceerdere multimodale presentaties: (1) Op welke manier wordt de generatie van multimodale uitingen bepaald door de context? (zie Bunt (1997), Bunt and Black (2000a) en Bunt and Girard (2005) voor een specifieke modellering van context) en (2) Welke factoren bepalen het gebruik van een modaliteit of een combinatie van modaliteiten? Voorafgegaan door een kritische bespreking van eerder voorgestelde algoritmes die referentiële uitingen genereren, wordt in dit proefschrift een nieuw algoritme gepresenteerd dat multimodale referentiële uitingen genereert. Het algoritme is gebaseerd op observaties in menselijke communicatie waarin referentiële uitingen die een wijsactie bevatten veel voorkomen (Beun and Cremers, 1998). De output van het algoritme steunt op drie factoren die van belang zijn in de menselijke productie van referentiële uitingen: prominentie, inspanning en zekerheid.

Het algoritme genereert een linguïstische referentiële uiting (zelfstandig gebruikt of in combinatie met een wijsactie) op basis van een driedimensionale notie van **prominentie**, waarin de linguïstische en de perceptuele context van de interactie gemodelleerd zijn. De linguïstische context wordt vastgesteld op basis van de recente discourse geschiedenis. De perceptuele context wordt bepaald door twee factoren: (1) de inherente prominentie van bepaalde objecten die opvallen omdat ze een eigenschap hebben die in de rest van het domein niet voorkomt; en (2) de visuele focus in het domein, die zich centreert om het object dat het laatst genoemd is in de discourse, waarin het bereik van een eventuele wijsactie ook is opgenomen. Een andere belangrijke factor is het *principle of minimal effort* (Clark and Wilkes-Gibbs, 1986), dat zegt dat in coöperatieve dialogen een spreker probeert om tegelijkertijd haar eigen **inspanning** en die van de geadresseerde te minimaliseren. Het doel van de spreker is de identificatie voor de geadresseerde makkelijk te maken door voldoende, maar niet teveel informatie te geven. Tegelijkertijd wil de spreker ook haar eigen inspanning minimaliseren bij het produceren van de referentiële uiting. Behalve op het in balans brengen van de hoeveelheid informatie heeft het principe ook betrekking op het soort informatie dat gebruikt wordt: in sommige gevallen is een wijsactie de optimale manier om aan object te referenen, terwijl in andere gevallen een linguïstische beschrijving of een combinatie van de twee meer geschikt is. De derde factor die een rol speelt in object identificatie is het streven van de spreker naar de **zekerheid** dat de hoorder de referentiële uiting kan interpreteren. Deze factor is geformaliseerd in het *principle of distant responsibility* (Clark and Wilkes-Gibbs, 1986), dat zegt dat een spreker er zeker van moet zijn dat de informatie in een uiting begrijpelijk is voor de geadresseerde. In overeenstemming met dit principe is de neiging van de spreker om referentiële uitingen te overspecificeren of om een preciese wijsactie te gebruiken, om de zekerheid op correcte identificatie door de hoorder te garanderen.

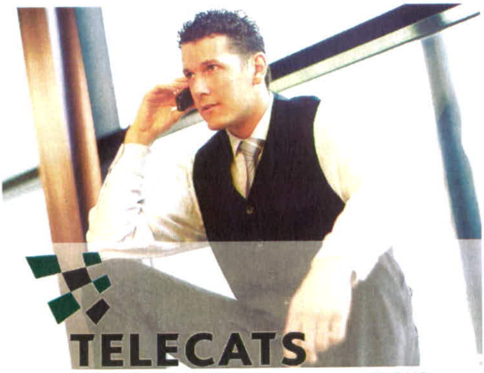
Het model voor wijsacties dat voorgesteld wordt in dit proefschrift beschrijft een nauw verband tussen de linguïstische informatie en de wijsacties die gebruikt kunnen worden. Het algoritme waarin dit model is geformaliseerd, genereert verschillende wijsacties variërend van nauwkeurige tot zeer onnauwkeurige. Het type wijsactie dat gegenereerd wordt is sterk afhankelijk van de perceptuele context, omdat het bereik van een onnauwkeurige wijsactie meer objecten bevat dan het bereik van een nauwkeurige wijsactie. Een directe consequentie van dit model is dat het aantal linguïstische eigenschappen dat nodig is om een onderscheidende multimodale referentiële uiting te genereren, varieert met het type van de gebruikte wijsactie. Het model voor wijsacties is geïmplementeerd in een multimodale uitbreiding van een nieuw algoritme voor de generatie van multimodale referentiële uitingen. Dit algoritme, voorgesteld door Krahmer et al. (2003), benadert de generatie van multimodale referentiële uitingen als een graafconstructieprobleem dat gebruik maakt van subgraaf-isomorfisme. Het besluit om een wijsactie te genereren wordt in het algoritme bepaald door een kostenfunctie die gebaseerd is op een fundamentele wet over het menselijke motorisch systeem

(Fitts, 1954). De output van het algoritme is gebaseerd op een afweging van de kosten van de wijsactie en de kosten van de linguïstische informatie die nodig zijn om een object te identificeren. Zodoende wordt het soort informatie dat gebruikt wordt in de referentiële uitingen bepaald op basis van de inspanning die het kost om een object te identificeren.

Het voorgestelde algoritme wordt geëvalueerd met behulp van productie-experimenten waarin participanten stimuli identificeren met behulp van spraak en gebaar. Op deze manier kan spontane multimodale data verzameld worden op basis van gecontroleerde input. Dit proefschrift beschrijft en analyseert twee studies waarin participanten objecten identificeren die verschillen in vorm, grootte en kleur. Eén studie is uitgevoerd in een heel beperkte omgeving: wijzen is verplicht en er wordt geen feedback gegeven. De andere studie heeft een meer natuurlijke, interactieve omgeving. De participanten in de twee studies zijn onderverdeeld in twee groepen: één groep bevindt zich dichtbij het objectdomein (de participanten kunnen de objecten aanraken bij het gebruik van wijsacties) en één groep bevindt zich op een grotere afstand (de participanten kunnen alleen wijsacties gebruiken die globaal de locatie van het object in het domein aangeven). De data resulterend uit deze twee studies laat duidelijke verschillen zien met betrekking tot het soort object waarnaar gerefereerd wordt en met betrekking tot de afstand tussen de spreker en het objectdomein. De meeste sprekers gebruiken geen linguïstische informatie om een object te identificeren dat zich op een kleine afstand van de spreker bevindt; een preciese wijsactie voldoet in zo'n geval. Sprekers gebruiken overgespecificeerde descripties wanneer ze verder van de referent afstaan; een onnauwkeurige wijactie is onzeker. Uit een gedetailleerde analyse van beide studies wordt geconcludeerd dat de co-variatie tussen het linguïstische materiaal en het soort wijsacties goed overeenkomt met de output van het algoritme. Het algoritme, dat alleen minimale descripties genereert, is echter niet ingesteld op het genereren van overspecificatie.

Om multimodale referentiële uitingen te genereren die vergelijkbaar zijn met die welke in menselijke communicatie voorkomen, is vervolgens een uitgebreid onderzoek gedaan naar zowel unimodale als multimodale overspecificatie met betrekking tot de resultaten van de twee productie-experimenten als ook de bevindingen in de cognitieve linguïstiek (o.a. Pechmann, 1989; Arts, 2004; Maes et al., 2004). Hierbij worden twee vragen gesteld: (1) Waarom en wanneer overspecificeren sprekers? en (2) Hoe overspecificeren sprekers? Uit de observaties in de cognitieve linguïstiek kan overspecificatie verklaard worden uit de onzekerheid aan de kant van de spreker over het feit of de hoorder in staat is om de referentiële uiting te interpreteren. Deze verklaring wordt bevestigd door de analyse met betrekking tot overspecificatie in de data van de bovengenoemde experimenten. Aan de hand van een nauwkeurige data-analyse en aan de hand van de observaties uit de cognitieve linguïstiek, wordt het graaf-gebaseerde algoritme aangepast, zodat overgespecificeerde referentiële uitingen gegenereerd kunnen worden. Dit gebeurt op basis van een schatting van de waarschijnlijkheid dat de gebruiker in staat is om de referentiële uiting correct te interpreteren in de huidige context. Aan zowel de wijsacties als aan de linguïstische informatie die in een referentiële uiting geïncludeerd kunnen worden, worden zekerheidsscores toegevoegd waarmee de gehele referentiële uiting in termen van zekerheid beoordeeld wordt. De mate van zekerheid die nodig is in een bepaalde situatie wordt hierbij vastgesteld met behulp van discourse en contextuele factoren. Het algoritme selecteert zodoende linguïstische informatie en wijsacties waarbij de kosten en zekerheidsscores zodanig tegen elkaar afgewogen worden dat de goedkoopste referentiële uiting gegenereerd wordt die met zekerheid door de gebruiker geïnterpreteerd kan worden. Met deze methode kan een groot aantal verschillende referentiële uitingen gegenereerd worden, van minimale tot uiterst overgespecificeerde.

In contrast met bestaande algoritmes voor de generatie van referentiële uitingen, kan het algoritme dat in dit proefschrift gepresenteerd wordt multimodale, contextgevoelige en mogelijk overgespecificeerde referentiële uitingen genereren op basis van een beperkt aantal onafhankelijk gemotiveerde parameters gerelateerd aan kosten en zekerheid.


TELECATS

ISBN
90.759
13.443