Generating Referring Expressions in a Multimodal Context

An empirically oriented approach

Ielka van der Sluis and Emiel Krahmer

Computational Linguistics, Tilburg University
IPO, Center for User-System Interaction, Eindhoven University of Technology

Abstract

In this paper an algorithm for the generation of referring expressions in a multimodal setting is presented. The algorithm is based on empirical studies of how humans refer to objects in a shared workspace. The main ingredients of the algorithm are the following. First, the addition of deictic pointing gestures, where the decision to point is determined by two factors: the effort of pointing (measured in terms of the distance to and size of the target object) as well as the effort required for a full linguistic description (measured in terms of number of required properties and relations). Second, the algorithm explicitly keeps track of the current focus of attention, in such a way that objects which are closely related to the object which was most recently referred to are more prominent than objects which are farther away. To decide which object are 'closely related' we make use of the concept of perceptual grouping. Finally, each object in the domain is assigned a three-dimensional salience weight indicating whether it is linguistically and/or inherently salient and whether it is part of the current focus of attention. The resulting algorithm is capable of generating a variety of referring expressions, where the kind of NP is co-determined by the accessibility of the target object (in terms of salience), the presence or absence of a relatum as well as the possible inclusion of a pointing gesture.

1 Introduction

The generation of referring expressions is one of the most common tasks in natural language generation. It is arguably also one of the most clearly defined ones: given a target object r and its properties, decide what is the best way to refer to r in the current context. In the past decade a number of algorithms for deciding on the form and/or content of a referring expression have been proposed (each given its own interpretation of what "the best way" is; computationally efficient, minimal, brief, etc.). Of these algorithms, the Incremental Algorithm of Dale & Reiter (1995) is generally accepted as the state of the art. The Incremental Algorithm is aimed at determining the content of a *distinguishing description*, that is: a definite description which is an accurate description of the target object r but not of any other object in the current domain of conversation. According to Dale & Reiter, the Incremental Algorithm has at least two important properties: (*i*) it is computationally attractive, because it has a polynomial complexity and is fast, and (*ii*) it is psychologically realistic, because it appears that humans produce distinguishing descriptions in a similar way as the Incremental Algorithm.

In recent years, various extensions of the Incremental Algorithm have been proposed. Horacek (1997) discusses a version which directly incorporates linguistic realization in the algorithm. Van Deemter (2001) presents a number of formal extensions to the Incremental Algorithm, concerned with, for instance, the generation of “vague” descriptions (‘the large mouse’) and the interaction with plurals (‘the large mice’). Krahmer & Theune (1998, 1999) (see also Theune 2000) offer an account of the role of context for the generation of referring expressions, and address a number of extensions required for the embedding of the Incremental Algorithm in a full fledged (spoken) natural language generation system. Most of these extensions explicitly aim at keeping the attractive properties of the Incremental Algorithm (in particular, speed, complexity and psychological plausibility).

In this paper, we discuss a further extension of the Incremental Algorithm, namely the generation of *multimodal* referring expressions: natural language referring expression which may include deictic, pointing gestures. There are at least two motivations for such an extension. First of all, in various situations a purely linguistic description can simply be too complex, for instance because the domain contains many highly similar objects. In that case, including a deictic, pointing gesture may be the most efficient way of singling out the target referent. The second reason is that when looking at human communication it soon becomes clear that referring expressions which include pointing gestures are very common (assuming, of course, that speaker and hearer can both directly perceive the domain of communication). Since our aim is to generate descriptions in a realistic way, it seems expedient to include such pointing gestures.

As the foundation of our enterprise we use the extended version of the Incremental Algorithm by Krahmer & Theune (1999). The multimodal extensions to this algorithm which we propose are based on empirical studies of how human speakers refer to objects in a shared work-space (Piwek et al. 1995, Cremers 1996, Beun & Cremers 1998). The main ingredients of our algorithm are the following. To begin with, we define a function which determines whether a pointing gesture is felicitous given the current context. This decision is based on two factors: the effort required for producing a pointing gesture and the effort required for a full linguistic description. Second, the algorithm explicitly tracks the focus of attention. Objects which are ‘closely related’ (in a way which we make precise below) to the most recent target object are taken to be more *salient* than objects which are not in the current focus space. Finally, we distinguish various reasons for which a particular object might be more salient than others. To do so, we define a three-dimensional notion of salience, combining focus space salience with linguistic salience and inherent salience. The last form of salience applies to objects which stand out perceptually with respect to the rest of the domain.[1]

[1] Various other algorithms for the generation of referring expressions in a multimodal setting have been proposed (for instance, Reithinger 1992, Claassen 1992, Huls et al. 1995, Cohen 1984, Salmon-Alt & Romary 2000). Of these, Salmon-Alt & Romary (2000) is closest in spirit to the current paper. Salmon-Alt & Romary also take the Incremental Algorithm as their starting point, and argue for an empirical, corpus-based approach. However, they concentrate on using information from different sources (discourse, perception, gestures) to restrict the context set of the Incremental Algorithm (in a similar way as done by Krahmer & Theune 1998, 1999). Contrary to the current paper, they do not address the question how the integration of such sources of information may be used for the actual *generation* of multimodal descriptions which combine language and gesture. Of the other cited works,

The outline of this paper is as follows. In section 2 we describe the Incremental Algorithm. In section 3 we summarize the main empirical findings regarding object reference in a multimodal environment and discuss the repercussions these have for the generation of multimodal descriptions. Then, in section 4 we show how the empirical rules can be captured in a formal and computational manner. Section 5 contains a sketch of the full algorithm, illustrated with a worked example.

2 The Incremental Algorithm of Dale & Reiter

The aim of Dale & Reiter's Incremental Algorithm (henceforth referred to as the D & R algorithm) is to efficiently generate a *distinguishing* description; a description that is applicable to the current object and not to any other object in the domain of conversation. Objects in a domain can be characterized in terms of a set of attribute-value pairs corresponding to their properties. For example, the objects in Figure 1 can be characterized as follows:

d_1 ⟨ *type, block* ⟩ ⟨ *color, white* ⟩ ⟨ *shape, square* ⟩ ⟨ *size, small* ⟩
d_2 ⟨ *type, block* ⟩ ⟨ *color, black* ⟩ ⟨ *shape, square* ⟩ ⟨ *size, small* ⟩
d_3 ⟨ *type, block* ⟩ ⟨ *color, black* ⟩ ⟨ *shape, square* ⟩ ⟨ *size, large* ⟩

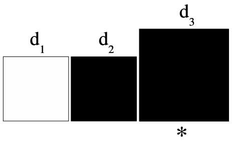

Figure 1: "the large black block"

One of the distinguishing properties of the Dale & Reiter algorithm is its use of a list of *preferred attributes*. In this list the properties relevant for the domain are ordered according to the preference that human speakers and hearers have when discussing objects in that particular domain. The exact ordering of properties for a particular domain is an empirical matter. However, some general trends exists. For instance, 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.[2]

The input of the D & R algorithm consists of the *target object* r and a *distractor set*, where r is the object to be described and where the distractor set contains all the objects in the domain except r itself. The D & R algorithm essentially iterates through the list of preferred attributes, adding a property to the description for r

the approach advocated here agrees most with Reithinger in that it focusses on multimodal generation in a way which models human behavior.

[2] For empirical evidence see e.g., Pechmann (1989) and Beun & Cremers (1998).

only if it rules out one or more of the objects in the distractor set not previously ruled out. Moreover, Dale & Reiter make the assumption that the property *type* should always be included in a distinguishing description even if it has no discriminating power (i.e., even if it did not rule out distractors). The D & R algorithm terminates when the distractor set is empty (success) or when all the properties of r have been checked (failure).

As an example reconsider Figure 1, and suppose that we apply the D & R algorithm to the object marked with a $*$ (d_3). This implies that the distractor set contains the other two objects in this particular domain, d_1 and d_2. For the time being, let us assume that the list of preferred attributes is ⟨ *type, color, shape, size* ⟩. First, the algorithm finds that the property *type* is not alone sufficient to distinguish $*$ (it rules out no distractors). Second, by including the property ⟨ *color, black* ⟩ in the set of selected properties, the algorithm can remove the white block d_1 from the distractor set. Still the set of remaining distractors is not empty. Third, the attribute *shape* has no effect on the set of remaining distractors (since d_2 and d_3 have the same shape). Fourth, the algorithm can use the relative attribute *size* to empty the distractor set. Finally, the D & R algorithm checks whether the *type* property was included, and since this was not the case, it is added to the set of distinguishing properties of $*$ after all. The set of selected properties can now linguistically be realized by the distinguishing description "the large black block". Note that the D & R algorithm does not itself output this linguistic description, rather it feeds the selected properties to a linguistic realizer.

Krahmer & Theune (1998, 1999) provide a number of extensions to the basic D & R algorithm. To begin with, they introduce a notion of linguistic context. 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. Linguistic salience is modelled using a *salience weight function*, according to which a salience weight is added to each object in the domain of conversation. With these additional salience weights the distractor set can be specified as the set that contains all the objects in the domain having a salience weight *higher than or equal to* the target object. This implies that when the target object is somehow salient, the search space is reduced. Hence, generally fewer properties will be required to empty the distractor set. Moreover Krahmer & Theune (1999) extend the D & R algorithm with the possibility of including relations, so that an object can be described in terms of its relations to other objects in the domain. Finally, following Horacek (1997), the algorithm directly produces linguistic descriptions. Only properties that rule out distractors *and* which can be realized within the constraints of the grammar are included in the final description. Here we take this extended version of the incremental D & R algorithm as our starting point.

The two main advantages of the D & R algorithm (which are essentially kept in the extensions of Krahmer & Theune) are its efficiency and its psychological realism. The efficiency of the algorithm is illustrated by its complexity, which is polynomial in time (Dale & Reiter 1995:247). Within the D & R algorithm there is no backtracking: once a property p has been selected, it will be realised in the final description, even if a property which is added later would render the inclusion of

p redundant. This is partly responsible for the efficiency, but Dale & Reiter claim that this is psychologically realistic because human speakers also often include redundant modifiers in their referring expressions (where they refer to Pechmann 1989).

On the other hand, the D & R algorithm also has its limitations. See for example Figure 2, where we want to single out one particular object in a domain where all the objects have most properties in common. 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 to use in natural communication. Moreover such descriptions contradict the principle of Minimal Cooperative Effort (Clark & Wilkes-Gibbs, 1986) stating that both the speaker's effort in producing the description and the hearer's effort in interpreting it should be minimal. Correspondingly, the most natural way to denote a particular object in Figure 2 is to add a pointing act to its description ("this block"). In the remainder of this paper, we describe the modifications and extensions required for the generation of such multimodal referring expressions.

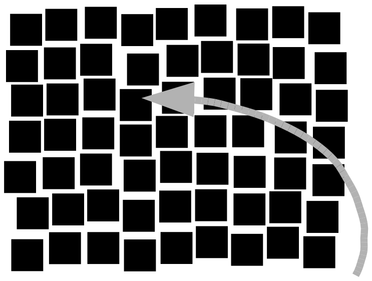

Figure 2: Disadvantage of the D & R algorithm

3 Empirical observations on multimodal object descriptions

In this section we discuss five rules for referring to objects in a multimodal setting, derived from the empirical results reported by Beun & Cremers (1998) and Cremers (1996). Beun & Cremers performed several experiments 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. This implied that participants could both talk about and point to the blocks in front of them. Below each of the rules is introduced and illustrated with an example. The first rule is concerned with *inherently salient objects*. Beun & 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.

Rule 1 If the target object is inherently salient within the domain of conversation, use reduced information.

According to Beun & Cremers' definition, the object labeled $*$ in Figure 3 is inherently salient, because it differs from the other objects by its color. Following rule 1, $*$ can therefore be referred to as "the block", even though this description is by itself not a distinguishing description since it is applicable to all the blocks.

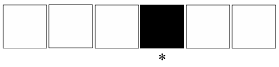

Figure 3: "the block"

It is worth stressing that this is the only rule for which Beun & Cremers found no significant evidence (due perhaps to the relatively small size of their corpus and the sparseness of inherently salient objects within a given domain). Interestingly, rule 1 is probably also the most controversial rule. 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 distractors. For example: a single pink elephant should be referred to as "the pink elephant" even when the contrast set entirely consists of flamingos. Arguably, world knowledge (that elephants are typically grey) plays an important role in this case, but not for the examples that Beun & Cremers discuss. This suggests that the respective positions of Beun & Cremers and Horacek do not really contradict each other, but apply to different cases. However, more research is required to test this hypothesis.

Rule 2 If the target object is located in the current focus area use only information that distinguishes the object from other objects in the focus area.

This rule can be illustrated by Figure 4. Assume that speaker and hearer are currently focussed on the two left most blocks in Figure 4. Within such a focus space, participants in the experiments of Beun & Cremers typically would describe the block marked with a $*$ simply as "the white block", even though there is another white block in the domain (but outside the current focus space).[3]

Rule 3 Use only information that distinguishes the target object from other objects that would be suitable for use in carrying out the current action.

This rule concerns functional expressions like "put the white block in between" in a situation in which there is only one object that fits in the intended space between two black objects (see $*$ in Figure 5). Notice that the referring expression crucially

[3] In fact, subjects typically produce *de witte* (English: 'the white (one)'). For the sake of simplicity, we follow Dale (1992) 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.

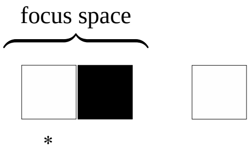

Figure 4: “the white block”

exploits the functional information expressed in the rest of the utterance. Only by considering the action, a hearer can decide which white block the speaker is referring to. Since the algorithm we propose in this paper is solely aimed at producing referring expressions and has no direct access to functional information expressed by the entire utterance, this problem will not be dealt with here.

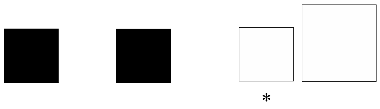

Figure 5: “put <u>the white block</u> in between”

> **Rule 4** Use absolute features as much as possible and use relative features only as necessary.

This rule is already implicit in the ordering of preferred attributes argued for by Dale and Reiter (see section 2). In the situation of Figure 6, the D & R algorithm would describe $*$ as “the black black” (not including *size*), but $**$ would be referred to as “the large white block”, since the inclusion of *white* (an absolute property) is not sufficient to rule out all distractors. For this, the relative property *large* is required.

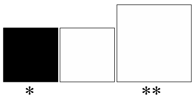

Figure 6: $*$ “the black block”; $**$ “the large white block”

> **Rule 5** If an explicit relatum is needed for referring to the target object, choose as a relatum an object that is salient.[4]

[4] This rule is a slight generalization of Beun & Cremers, who do not discuss linguistic salience and only mention relata which are inherently salient and/or in the focus of attention.

In other words, if another object is needed to describe the target object, then select one which is salient. An object can be salient for a number of reasons, for instance because it has just been talked about (linguistic salience, see Krahmer & Theune, 1999). Suppose that the black block in Figure 7 has just been talked about and the grey block has not been mentioned. In that context, $*$ is typically referred to as “the white block next to the black one”.

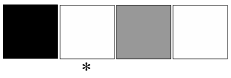

Figure 7: “the white block next to the black one”

4 Main ingredients of the multimodal algorithm

In this section we describe the main novelties of our multimodal extension of the D&R algorithm, based on the empirically motivated rules discussed above.

4.1 When to point?

According to the principle of Minimal Cooperative Effort (Clark & Wilkes-Gibbs, 1986), a balance should be found between on the one hand the speaker’s effort to produce a description and on the other hand the effort necessary for interpretation of this description by the hearer. Hence we assume that the decision to use a pointing act for distinguishing an object is determined by two factors: the effort of pointing and the effort required for a full, linguistic description.

We assume that the effort of pointing is determined by two factors: the distance to and the size of the target object. The trade-off between these factors has been captured in Fitts’ law, the index of difficulty ID (Fitts, 1954). The index is computed from the size of the target object and the distance between the object and the position of the pointing device used, in our case the speaker’s hand. If this index is below a certain (task and domain dependent) threshold (i.e., it is easy to point), the algorithm includes a pointing act in the output.[5]

DEFINITION 1 (Index of Difficulty (ID))

$$\text{ID} = \log_2\left(\frac{2d}{w}\right)$$

[5] Piwek et al. (1995:13) contains some suggestive evidence supporting this idea. They found that builders are more likely to point than instructors. This might be explained by the fact that builders, by the very nature of their task, are forced to touch the blocks anyway, implying that the distance between their hands and the blocks is much smaller for builders than for instructors. Interestingly, there is also a certain amount of individual variation in that some subjects point very frequently while others never do. This suggests that the threshold is not only task and domain dependent, but also subjective.

where w is the width (or 'size') of the object and d is the distance from the pointing device to the object.

The second factor that contributes to the decision to point is the effort required for a purely linguistic description. Arguably, this effort is proportional to the number of attributes and relations needed to generate a distinguishing description. When the complexity of the linguistic description is above a certain threshold, the linguistic description generated so far is discarded and a pointing act is generated instead.[6]

Once a pointing act is included in a referring expression, we assume that the distractor set is immediately emptied and the target object is uniquely identified (it will be clear for the hearer which object is being referred to).[7] Finally, if an object cannot be uniquely identified in terms of a purely linguistic description, then the algorithm similarly provides for a pointing act accompanied by a short and general linguistic expression.

4.2 Computing the focus space

The second rule of Beun & Cremers described in section 3 concerns the objects in the current focus space from which the target object should be distinguished. The notion of a focus space is not only psychologically plausible, but is also beneficial from a computational point of view. By defining the focus space as a subset of the objects in the whole domain, the search space of the algorithm is reduced. In this section we will first give an insight in how we define the current focus space using an example, before we present a formal definition.

The focus space consists of the last mentioned object o and the set of objects directly related to o (such as to o's left or right, below o etc.). An object d is standing in a direct relation to an 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 8. If the object last mentioned is the black block, the focus space contains three blocks as shown in the picture (the black block itself plus the two white ones). The grey block to the right of the black block is excluded because there is a closer block which is also to the right of the black block. Once the algorithm has generated a referring expression for $*$, the focus space needs to be updated. The updated focus space contains $*$ and the set of objects that are directly related to

[6] We have our doubt whether the approach which first tries to generate a complex description only to discard it later in favor of a pointing act is psychologically realistic. In a sense, this problem is a specific instance of the general problem of finding a solution which requires minimal effort (Zipf 1949). Arguably, one needs to calculate the effort required for *each* solution to be able to determine which one is minimal. And this process certainly is not minimal.

[7] Mariët Theune (p.c.) notes that less precise (i.e., non distinguishing) gestures could also be useful. In particular, if all distractors close to the target object have been ruled out by the properties selected so far, then adding such a less precise gesture pointing to the region which contains the target object would also suffice. Such a gesture could typically also be used when the hand is relatively far away. Rather then postulating an absolute threshold, it would be interesting to assign a "denotation" to pointing gestures: if the hand is close, the denotation will consist of few objects, if it is father away, the denotation will consist of more objects.

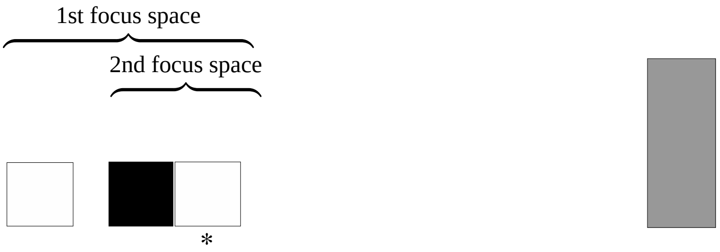

Figure 8: “the white block to the right of the black one”

$*$, which would be the black and the grey block. However, on a second note the grey block 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, we use *perceptual grouping* (Thorisson, 1994). Thorrison defines a proximity score for the distance of each object in the domain to a particular object o. The proximity score between the object o and an object d in the domain D is defined as follows:

$$F = \frac{\text{distance}(o, d)}{\max\limits_{y \in D} \; (\text{distance}(o, y))}$$

By setting a threshold to this fraction, we can exclude far away objects from the focus space of o. For example, consider Figure 8 again, and suppose, for the sake of illustration, that we set the threshold to 0.5. In Figure 8, the distance between $*$ and the black block is 1cm, the distance between $*$ and the white block to the left of the black block is 1.5cm and the distance between $*$ and the grey block is 6cm. Then the maximal distance in this domain is 6cm. By perceptual grouping with a threshold of 0.5 this will result in the following fractions (for $o = *$): $1/6 = 0.17$ ($d =$ black block), $1.5/6 = 0.25$ ($d =$ leftmost white block) and $6/6 = 1$ ($d =$ grey block). Hence we can exclude the grey block from the focus space of $*$.

Summarizing, the new, updated focus space contains $*$ (the last mentioned object) and the black block. In definition 2, the focus space of an object o is formally defined as the union of the object o itself with the set of objects in the domain that are closest to o for any relation of a given type (with *type* $\in$ { left_of, below, ... }) and which are not ‘too far away’ in terms of perceptual grouping (with F as defined above).

DEFINITION 2 (Focus space)
focus_space(o) $= \{o\} \cup \{d \in D \mid$ relation(*type*, o, d) $\wedge \neg \exists d'$ (relation(*type*, o, d') $\wedge$ (distance(o, d') $\leq$ distance(o, d))) $\wedge F \leq$ 0.5 }

Notice that the target object r need not be an element of the current focus space around object o. When it is not, we speak of a *focus shift*.

4.3 A threedimensional notion of salience

As mentioned in section 2, Krahmer & Theune (1998, 1999) extended the D & R algorithm with a linguistic notion of salience. As we have seen in the previous section, other forms of salience are also relevant for the generation of referring expressions. In particular, objects can be inherently salient and/or they can be salient because they are in the current focus space. To model these differences, we define a threedimensional notion of salience. More precisely, each object in the domain receives three salience weights, one indicating whether or not the object is linguistically salient, one indicating whether it is inherently salient and one indicating its focus space salience. The total salience weight of an object is determined by taking the sum of the three separate salience weights (which is in line with the observation of Beun & Cremers (1998:141) that an inherently salient object in the current focus space is more salient than an inherently salient object outside the focus space). Arguably, some forms of salience are more important than others. We assume that linguistic context salience is primary, for instance, in the sense that an object r which was just described is more salient than an object which is in the current focus space (i.e., close to r) but has not itself been mentioned so far. In a similar vein, we take it that an object which is in focus is somewhat more salient than an object which is inherently salient but falls outside of the current focus space.

The first two rules of Beun & Cremers are immediately satisfied, when (following Krahmer & Theune 1998, 1999) we restrict the distractor set to objects which are at least as salient as the target object. This implies that if the target object is inherently salient and/or part of the focus space, this will generally lead to a reduction of the distractor set (assuming that not all objects are equally salient) and consequently fewer properties should suffice for emptying the distractor set. Thus, when an object is inherently salient we can use reduced information. When the target object is part of the current focus space (and is not linguistically salient), the distractor set will typically consist of the other objects that are in the current focus space, together with the objects that are linguistically salient. According to the fifth rule, when a relatum is needed the algorithm will select the most salient one.

Within the algorithm presented here, linguistic salience (L-salience) is modelled as it is done by Krahmer & Theune (1999), 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 10 (maximum salience). In the initial state, every object is assigned an L-salience weight 0. There are various ways to determine inherent salience (I-salience); see Cremers (1996:24) for references and discussion. Here we opt for a strong criterion, where an object is inherently salient only if for some attribute it has a particular value V_1 while the other objects in the domain all have a different value V_2 for that particular attribute. If an object is inherently salient, it has a constant I-salience weight of 1. Finally, focus space salience (FS-salience) is easily determined given definition 2. An object has an FS-salience weight of 2 iff it is part of the current focus space.

Definition 3 calculates the salience weight of each object d in a state s_i as the sum of the three kinds of salience associated with d in that state. In the initial state s_0 (the beginning of the discourse) no object has been described and we assume that there is no focus space. Thus, initially each object in the domain has an L-salience and an FS-salience weight of zero. In this definition, $C_f(U_i)$ is the ordering of the forward looking centers of U_i (the sentence uttered at time i) according to Centering Theory (Grosz et al. 1995). This ordering is such that the syntactic subject of U_i is the most salient object (mapped to salience weight 10) followed by the indirect object (mapped to 9) and the other objects (mapped to 8). Thus, more formally, $\mathsf{level}(d_i, \langle\, d_1, \ldots, d_n\rangle) = \mathsf{max}(0, 11 - i)$, where $\langle d_1, \ldots, d_n\rangle$ 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 1, unless it is already 0. The FS-salience weight 2 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-salience).

DEFINITION 3 (Threedimensional Salience)
For each object $d \in D$, the salience weight of d in state s_i is

$$\text{salience\_weight}(d, s_i) = \text{I-salience}(d, s_i) + \text{L-salience}(d, s_i) + \text{FS-salience}(d, s_i)$$

where:

Linguistic Salience

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

$$\text{L-salience}(d, s_{i+1}) = \begin{cases} \mathsf{level}(d, C_f(U_i)) & \text{if } d \in C_f(U_i) \\ \mathsf{max}(0, \text{L-salience}(d, s_i) - 1) & \text{otherwise} \end{cases}$$

Focus Space Salience

$\text{FS-salience}(d, s_0) = 0$

$$\text{FS-salience}(d, s_{i+1}) = \begin{cases} 2 & \text{if } d \in \text{focus\_space}(o) \wedge o = \underset{d'}{\mathsf{max}}\ \text{L-salience}(d', s_i)) \\ 0 & \text{otherwise} \end{cases}$$

Inherent Salience

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

4.4 Linguistic Realization

So far we have not said much about the actual linguistic realization; here we make up for this lack. To determine the form of the multimodal referring expressions we inspected the corpus collected by Beun & Cremers. For starters, we can decide on the list of preferred attributes for the block domain used in their experiments. Table 1 contains the distribution of the attributes in 141 initial, distinguishing descriptions from the corpus of Beun & Cremers. It is clear that *color* is by far the most preferred attribute in this domain. The attribute *type* is the least preferred attribute in this domain. This is not surprising since all objects in this domain are of the block type, which makes this is very uninformative property. However, the D & R algorithm stipulates that *type* should always be included in the final description, even if it is not discriminating. Table 1 clearly contradicts this. We conjecture that it should not be the *type* attribute which is always included, but rather the most preferred attribute for a particular domain.

attribute	$+$ *Point*	$-$ *Point*	total
Color	38	42	80
Location	4	19	23
Shape	3	10	13
Type	5	8	13
None	11	1	12

Table 1: Selected attributes as a function of pointing acts.

In the (Dutch) corpus used in the various studies of Cremers, Beun and Piwek, demonstrative determiners are preferred over articles. Piwek & Cremers (1996) claim that Dutch proximate demonstratives (deze/dit; 'this') are preferred when referring to objects which are relatively hard to access. The use of distal demonstratives (die/dat; 'that') is equally distributed over more and less accessible referents.[8] Piwek & Cremers' notion of accessibility can be defined for the purposes of the current paper as follows:

DEFINITION 4 (Accessibility)

$$\text{accessible}(r, s_i) = \begin{cases} \text{False} & \text{if} \quad \text{I-salience}(r, s_i) = 0 \vee \\ & \qquad \text{L-salience}(r, s_i) \leq 8 \vee \\ & \qquad \text{FS-salience}(r, s_i) = 0 \\ \text{True} & \text{otherwise} \end{cases}$$

The choice of determiner does not solely depend on the accessibility of the object, also the occurence of a relatum or a pointing act is important. In the data from Beun & Cremers' corpus, proximate demonstratives are never used in combination with a relatum, contrary to distal demonstratives. For the relatum itself a definite article is used. On the other hand, the data from the experiments by Beun & Cremers show that in all cases in which a proximate demonstrative is used it is accompanied by a pointing act. A distal demonstrative in combination with a pointing act occurs only in 35% of the cases in which a distal demonstrative is used. Piwek & Cremers

[8] We are aware of the fact that there are certain differences between English and Dutch where determiners are concerned. Our algorithm, primarily based on Dutch data, formalizes the findings of Piwek & Cremers for Dutch demonstratives. For the generation of English referring expressions, some minor changes in the selection of determiners are required.

(1996) conclude that distal demonstratives are preferred without a pointing act in case they are used to refer to accessible entities.

In sum, the resulting algorithm generates a variety of ‘multimodal’ NPs where the kind of NP is determined by the occurence of a pointing act, the presence or absence of a relatum and the accessibility of the target object described in terms of salience. In contrast to including the *type* of the target object (as the D&R algorithm stipulates) we include the most preferred attribute, which is in the block domain the property *color*.

5 Outline of the multimodal algorithm

In this section we present our expanded, multimodal version of the algorithms of Dale & Reiter and Krahmer & Theune, simply called *make_referring_expression*, and illustrate it with two examples.[9]

5.1 Sketch of the algorithm

Before *make_referring_expression* is actually called, certain variables are initialised using the procedure *initialise*. This procedure takes as input three arguments, namely the target object R for which a referring expression should be generated, the current focus space FS and the current state S. The first variable which is initialised is *SW_list*. This is a list of all the objects in the domain, ordered with respect to their salience weights. *PA_list* is a list with the attribute-value pairs of the object *R*, ordered according to human preference. In *PR_list* all the relations of *R* with other objects in the domain are listed (*next to*, *on top of*, *below*, etc.), ordered by the salience weights of the relata. *Access* is a boolean variable depending on the three kinds of salience of *R*. *RemDist* is the set of objects from which *R* has to be distinguished, containing all the objects with a salience weight higher than or equal to the salience weight of *R*. Finally *make_referring_expression* is called to generate a distinguishing expression for the target object *R* with the relevant, initialized parameters. Notice that the state S is also passed on as a parameter. This is required for the generation of relata, where all the relevant parameters need to be re-initialised for the generation of the relatum. We assume that the parameters *S* and the *FS* are both updated by the main generation algorithm (hosting *make_referring_expression)* as soon as it has produced a complete utterance.

```
initialise(R,FS,S)
SW_list = generate_salience_weights(R,FS,S)
PA_list = preferred_properties(R)
PR_list = preferred_relations(R,SW_list)
Access = accessible(R,SW_list)
RemDist =dist(R,SW_list)
make_referring_expression(R,FS,PA_list,PR_list,RemDist,Access,S)
```

[9] Krahmer & Theune’s modified version of the D & R algorithm has been implemented. We are currently working on extending this implementation to include the multimodal additions described in this section.

In *make_referring_expression* it is first determined whether pointing to R does not require too much effort. For this the function *index_of_difficulty* is used (see definition 1), and if this index is below a certain threshold, a pointing act is included. If this is the case, the *RemDist* list is emptied, and an accompanying linguistic expression is put together by including the most preferred property of *R*. Finally, the appropriate determiner is inserted and the resulting tree is returned. If the effort of pointing is too high (the boolean variable *Point* is *False*), the algorithm first tries to find properties that rule out objects in the *RemDist* list; calling *find_properties* results in *Tree1*. If all the properties of *R* are considered and the *RemDist* still contains objects from which *R* is not distinguished, the algorithm selects the first relation on the *PR_list* which rules out distractors. If a relation between the target object R and a (salient) relatum R' is selected, then the algorithm also has to generate a referring expression for R'. For this purpose the main algorithm is called again, but now with R' as input (*initialise* (R', FS, S)). The resulting description for R' is stored in *Tree2*, after which *Tree1* and *Tree2* are combined resulting in *Tree* and the boolean *Relatum* (indicating whether or not a relatum was needed for the generation of R) is set to *True*. If no relation was needed to empty the *RemDist* list, only *Tree1* is stored in *Tree*. As argued in section 4.1, two factors contribute to the decision to point, namely the effort of pointing and the effort of a linguistic description. The second factor is modelled in the algorithm by counting the required properties and attributes included in the description of R under construction. If this number is above a certain Threshold, the algorihm classifies the linguistic description (stored in *Tree*) as too complex and discards it in favour of a pointing act. Finally, with *include_most_preferred_property*, *Tree* is enriched with the most preferred property of *R* if this property was not already present in *Tree*. (Notice that this marks a difference from the Incremental Algorithm for those domains where *Type* is not the most preferred attribute.) Finally, with *insert_det* (see below) a determiner is added to *Tree* and the complete description is returned.

```
make_referring_expression(R,FS,PA_list,PR_list,RemDist,Access,S)

Point = index_of_difficulty(R)
if (Point == true) RemDist = [ ]
if (Point == false)
    Tree1 = find_properties(R,PA_list,RemDist)
    if (RemDist $\neq$ [ ])
        Tree2 = find_relations(R,FS,PR_list,RemDist,S)
        if (Tree2 $\neq$ nil)
            Tree = add_tree(Tree1,Tree2)
            Relatum = True
        else Tree = Tree1
    if (Threshold $\leq$ determine_efficiency(Tree)) || (RemDist $\neq$ [ ])
        Point = True
        Relatum = False
        Tree = nil
Tree = include_most_preferred_property(Tree,R)
Tree = insert_det(R,Tree,Access,Point,Relatum)
return Tree
```

The functions *find_properties* and *find_relations* are (minimal variants of) functions found in Dale & Reiter (1995) and Krahmer & Theune (1999) respectively. The function *find_properties* determines which properties of the *PA_list* rule out any of the remaining distractors and should therefore be included in the referring expression of *R*, essentially as it is done in the Incremental Algorithm. The function *find_relations* looks for relations of R to be included in the referring expression, in essentially the same way as for properties. The function *insert_det* determines which determiner to add to the tree generated by the algorithm according to the Dutch data of Beun & Cremers (1998) and the rules for Dutch proposed by Piwek & Cremers (1996). When the referring expression for *R* contains no relatum, a pointing act is included and the object is inaccessible (it has a low salience weight), then a proximate demonstrative is inserted. A distal is used when the referring expression includes a pointing gesture or when the referent is accessible. In all other cases a definite article is selected.

5.2 Worked example

We end our presentation of the multimodal algorithm by discussing an example in which a sequence of two generated referring acts is considered. In the initial situation in Figure 9 below, there is no focus space, no inherently salient object and no linguistic salience. The salience weights of all the objects in the domain of conversation are zero. The task is to refer to $*$. The algorithm includes a pointing act because the index of difficulty for pointing to this object is below the threshold *C*. At this point, all distractors are ruled out. Next, the relevant value of the most preferred attribute is added (*color*). A proximate demonstrative determiner is chosen on the basis of the values of three boolean variables: *Point* = *True*, *Access* = *False* (since all objects are equally non-salient) and *Relatum* = *False*. The algorithm outputs a pointing act accompanied by the referring expression “this black block”.

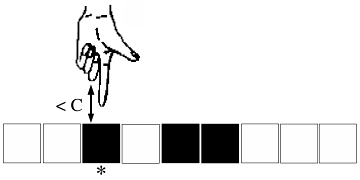

Figure 9: “this black block”

In the follow-up situation, presented in Figure 10, the focus space is updated (in the way defined above) and indicated by the curly bracket. Now the previously described object (the black one) is the only object in the domain with a non-zero linguistic salience weight. Within the focus space the black block has a total salience weight of 12 (10 for maximal linguistic salience plus 2 for focus space salience) and both the white blocks have a salience weight of 2. The task is to refer to $*$, and the distractor set contains all the objects with a salience weight higher than or equal to the target object (in this particular example, the distractor set coincides with the focus space). Because the hand is too far away from $*$, the index of difficulty is above the threshold and no pointing act is generated. Instead, the algorithm enters the *find_properties* routine. The algorithm adds the preferred property (*color*) to distinguish $*$ from the black block in the focus space. No other properties can be used to rule out the other white block. So, next the function *find_relations* is called. The function first tries the relation with the most salient relatum (the first element of the *PR_list*: the left-of relation between $*$ and the black block. Including this relation does empty the distractor set: the remaining white block in the distractor set does stand in the left-of relation to a black block, but that one falls outside the focus space and thus has a zero salience weight. The algorithm generates a description for the relatum: "the black one". This description is inserted in the description for $*$. Finally, a distal demonstrative determiner is inserted since the referent is accessible.[10]

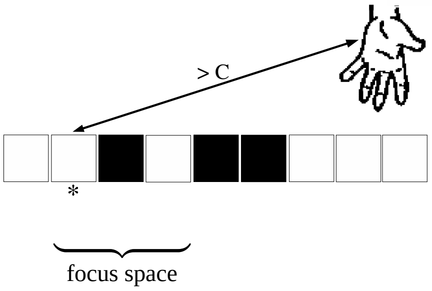

Figure 10: "that white block to the left of the black one"

6 Concluding remarks

As noted in section 2, Dale & Reiter's Incremental Algorithm has two attractive properties: it is computionally attractive and psychologically realistic. To what extent has our proposed algorithm inherited these properties? Given that our extensions are to a large extent motivated by empirical research, the current algorithm can be claimed to model the way humans refer to objects in a multimodal setting. In this respect, the multimodal algorithm presented here is as psychologically realistic as Dale & Reiter's. It arguably also captures more of the variety found in human object references than the Incremental Algorithm does. The original Incremental Algorithm is efficient (polynomial) because there is no possibility of back-

[10] Recall that the algorithm is currently aimed at generating Dutch descriptions (*die witte links van de zwarte*). The use of a distal demonstrative determiner is probably less natural for English. See Piwek & Cremers 1996 for discussion.

tracking, there can be no ‘wrong’ decisions. Unfortunately, as soon as we include relations, this property cannot be kept: the generation of relational descriptions is NP complete (see e.g., Krahmer et al. 2001). In general, there is no guarantee that we always immediately select the right relatum, so either backtracking or multiple embeddings may be required. However, two factors should be noted. First, the use of salience guides the search for a relatum. In particular, following the empirical findings of Beun & Cremers, only salient relata are chosen, which in most cases offers a substantial reduction of the search space. Second, we define an upper bound to the number of properties and relations which can be included in the final description. When this maximum value is reached, the tree under construction is discarded and a pointing act is included. Fortunately, as soon as such an upper bound is defined, we regain polynomial complexity (see e.g., van Deemter 2001).

Acknowledgements

We would like to thank Paul Piwek, Mariët Theune and Harry Bunt for discussions and comments on an earlier version of this paper.

References

[1] Beun, R.J. & A. Cremers (1998), Object reference in a shared domain of conversation, *Pragmatics & Cognition* **6**(1/2):121-152.

[2] Cohen, P. (1984), The pragmatics of referring and the modality of communication, *Computational Linguistics* **10**(2):97-125.

[3] Claassen, W. (1992), Generating referring expressions in a multimodal environment, in: *Aspects of Automated Natural Language Generation*, R. Dale et al. (eds.), Springer Verlag, Berlin.

[4] Cremers, A. (1996), *Reference to objects: An empirically based study of task-oriented dialogues*, Ph.D. dissertation, Eindhoven University of Technology.

[5] Clark, H. & D. Wilkes-Gibbs (1986), Referring as a collaborative process, *Cognition* **22**:1-39.

[6] Dale, R. & E. Reiter (1995), Computational interpretations of the Gricean maxims in the generation of referring expressions, *Cognitive Science* **18**:233-263.

[7] Deemter, K. van (2001), Generating referring expressions: Beyond the Incremental Algorithm, *Proceedings of the 4th International Workshop on Computational Semantics*, Tilburg, The Netherlands.

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

[9] Grosz, B., A. Joshi & S. Weinstein (1995), Centering: A framework for modeling the local coherence of discourse, *Computational Linguistics* **21**(2):203-225.

[10] Horacek, H. (1997), An algorithm for generating referential descriptions with

flexible interfaces, *Proceedings of the 35th ACL/EACL*, 206-213, Mardrid, Spain.

[11] Huls, C. E. Bos & W. Claassen (1995), Automatic referent resolution of deictic and anaphoric expressions, *Computational Linguistics* **21**(1):59-79.

[12] Krahmer, E. & M. Theune (1998), Context sensitive generation of referring expressions. *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, 1151-1154. Sydney, Australia.

[13] Krahmer, E. & M. Theune (1999). Efficient generation of Descriptions in Context, *Proceedings of the ESSLLI Workshop on the Generation of Nominals*, R. Kibble and K. van Deemter (eds.), Utrecht, The Netherlands.

[14] Krahmer, E. S. van Erk & A. Verleg (2001). A meta-algorithm for the generation of referring expressions, *Proceedings of the Eight European Workshop on Natural Language Generation*, Toulouse, France.

[15] Pechmann, T. (1989), Incremental speech production and referential overspecification, *Linguistics* **27**:98-110.

[16] Piwek, P., R.J. Beun & A. Cremers (1995), *Demonstratives in Dutch cooperative task dialogues.*, IPO manuscript 1134, Eindhoven University of Technology.

[17] Piwek, P. & A. Cremers (1996), Dutch and English demonstratives: A comparison, *Language Sciences* **18**(3-4):835-851.

[18] Reithinger, N. (1992), The performance of an incremental generation component for multi-modal dialog contributions, in: *Aspects of Automated Natural Language Generation*, R. Dale et al. (eds.), Springer Verlag, Berlin.

[19] Salmon-Alt, S. & L. Romary (2000), Generating referring expressions in multimodal contexts, *Proceedings of the INLG 2000 workshop on Coherence in Generated Multimedia*, Mitzpe Ramon, Israel.

[20] Theune, M. (2000), *From data to speech: Language generation in context*, Ph.D. dissertation, Eindhoven University of Technology.

[21] Thorisson, K. (1994), Simulated perceptual grouping: An application to human computer interaction, *Proceedings of the 16th Annual Conference of Cognitive Science Society*, 876-881, Atlanta GA.

[22] Zipf, G.K. (1949), *Human behavior and the principle of least effort: An introduction to human ecology*, Addison-Wesley, Cambridge.