Incremental Generation of Multimodal Deixis Referring to Objects

Alfred Kranstedt, Ipke Wachsmuth
University of Bielefeld
Artificial Intelligence Group, Faculty of Technology
D-33594 Bielefeld
{akranste, ipke}@techfak.uni-bielefeld.de

Abstract

This paper describes an approach for the generation of multimodal deixis to be uttered by an anthropomorphic agent in virtual reality. The proposed algorithm integrates pointing and definite description. Doing so, the context-dependent discriminatory power of the gesture determines the content-selection for the verbal constituent. The concept of a pointing cone is used to model the region singled out by a pointing gesture and to distinguish two referential functions called object-pointing and region-pointing.

1 Introduction

Deixis anchors utterances in their spatio-temporal context and can therefore be seen as a central part of the *aboutness* of language. In face-to-face interaction deixis is typically expressed using several modalities. In this paper we describe an approach for the generation of multimodal deixis referring to objects. These expressions integrate two different kinds of referring to objects, indicating the location of an object by pointing or describing its properties by a definite description. Following McNeill [McNeill, 1992], we distinguish between abstract pointings and pointings into concrete domains. Here, we focus on pointings into concrete domains co-occurring with verbal expressions, typically definite noun phrases. As we will see further on, the interrelation between gesture and verbal expression is of a complex nature. Both are often under-specified; only together they identify the referent unambiguously.

In the growing number of applications which are characterised by an anthropomorphic human-computer interface there is an increasing need for robust mechanisms when referring to objects by speech and gesture. Emphasising the importance of deixis in the interaction with humanoid agents, [Lester *et al.*, 1999] introduced the expression *deictic believability*. In contrast, the generation of multimodal reference is an open issue until now, while the generation of referring expressions, which identify objects by description, is well investigated (several computational models have been proposed over the last years).

The approach proposed for the generation of multimodal deictic expressions is based on the incremental algorithm by [Dale and Reiter, 1995]. This algorithm for the generation of verbal referring expressions was adapted in that the spatial property location, which can be expressed either absolutely by pointing or relationally by verbal expressions (e.g. "the left object"), is evaluated besides other object properties in content-selection. Taking account of the inherent impreciseness of pointing gestures, two referential functions of pointing are distinguished, *object-pointing* and *region-pointing*. While object-pointing refers on its own, region-pointing is used to narrow down the set of objects from which the referent has to be distinguished by a definite description.

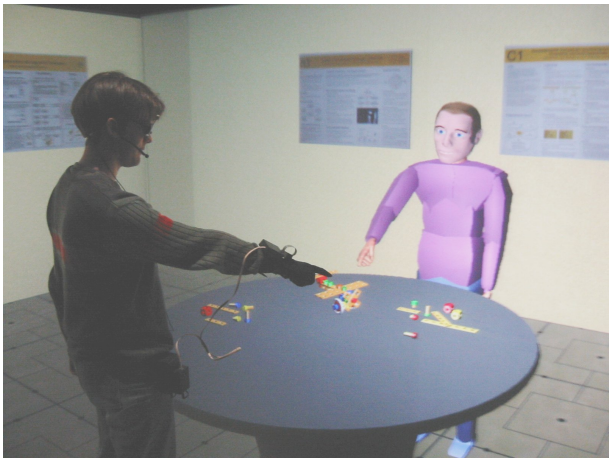


Figure 1: The interaction scenario

The described research is undertaken in the course of the development of human computer interfaces for natural interaction in Virtual Reality (VR). Conducting empirical investigations and developing computational models we focus on dialogues in a construction task domain, where a kit consisting of generic parts is used to construct models of mechanical objects such as a toy airplane. A typical setting consists of a human instructor and an anthropomorphic virtual agent interacting in face-to-face manner in VR realised in a three-side Cave-like installation. Our human-sized virtual agent called Max is able to interpret simple multimodal (speech and gesture) input from the human instructor on the one hand and to produce synchronised output involving synthetic speech, fa-

cial display and gesture [Kopp and Wachsmuth, 2004] on the other hand. As illustrated in Fig. 1, Max and the human dialogue partner are located at a virtual table with toy parts and communicate about how to assemble them. Speech and gesture are used by both interlocutors to specify tasks and select relevant objects.

On the way towards dialogue generation a setting we call *demonstration games* has been established to get at the understanding and generation of complex deictic expressions. These demonstration games which reduce interaction to two turns are based on the minimal dialogue games proposed by [Mann, 1988]. The setting consists of two interlocutors located at a table with some objects lying on it. One interlocutor has to indicate an object by speech and gesture, and the other interlocutor has to give feedback on which object was referred to. In a human-human realisation this setting is used to conduct empirical studies to investigate the referring behaviour of subjects [Kühnlein and Stegmann, 2003; Lücking *et al.*, 2004]. An annotated corpus was acquired which comprises 65 multimodal demonstrations uttered by several subjects. In a human-machine realisation the setting is used as a testbed for the developed communicative abilities of our agent concerning deictic reference. This enables us to directly link and compare the results of speech-gesture processing with empirically recorded data in a comparable setting [Kranstedt *et al.*, 2004].

In the section to follow, the role of pointing in multimodal referring expressions is analysed in more detail. The concept of a pointing cone and two referential functions of pointing, *object-pointing* and *region-pointing*, are introduced. In Sec. 3 a short overview on related work concerning the generation of referring expression is given. The incremental algorithm proposed by [Dale and Reiter, 1995] underlying our approach is outlined in Sec. 3.1. In Sec. 4 the content-selection algorithm proposed for multimodal deictic expressions is described in detail. Sec. 5 illustrates its functionality giving an example. Sec. 6 describes the embedding of the algorithm in a generation framework and the current potentials and limitations of the approach. The paper concludes with a short discussion of the proposed approach.

2 Pointing in Multimodal Deictic Expressions

There is little doubt in the literature that pointing is tied up with reference as the following quotation from [Lyons, 1977, p. 654] shows:

When we identify an object by pointing to it (and this notion, as we have seen, underlies the term 'deixis' and Peirce's term 'index': cf. 15.1), we do so by drawing the attention of the addressee to some spatio-temporal region in which the object is located.

Pointing, then, is related to objects indicated and regions occupied. Lyons also emphasises that certain kinds of expressions, especially definite descriptions, are closely linked to pointing or demonstration (op. cit., p. 657):

[. . .] definite referring noun-phrases, as they have been analysed in this section, always contain a deictic element. It follows that reference by means of definite descriptions depends ultimately upon deixis, just as much as does reference by means of demonstratives and (as we saw in the previous section) personal pronouns.

Pointing and definite descriptions therefore represent on the one hand different kinds of referring to objects (indicating their location or describing their properties). On the other hand they appear to be intimately connected. Lyons does not discuss how exactly pointing and verbal expression are related. Following [Rieser, 2004], we pursue a line of thought associated with Peirce, who maintains the idea of gestures being part of more complex signs [Peirce, 1965]. Transferring that to deixis we call such complex signs, which are composed of a pointing gesture and a definite description, *complex demonstrations*. In other words, complex demonstrations are definite descriptions to which pointings add content, either by specifying an object independently of the definite description (Lyons' attention being drawn to some object) or by narrowing down the description's restrictor (Lyons' spatio-temporal region). Below, we refer to these two possibilities as the respective functions of demonstration, see [Rieser, 2004] for discussion. If a pointing gesture uniquely singles out an object, it is said to have *object-pointing* function. If the gesture draws the attention of the addressee to a region making the objects inside it salient it is ascribed a *region-pointing* function.

The distinction between object-pointing and region-pointing is closely connected with the observation that pointing gestures are inherently ambiguous, varying with the distance between pointing agent and referent. In the empirical data collected in our demonstration games we found object-pointing only in demonstrations to objects near to the demonstrating subject, while pointings to objects farther are accompanied by definite descriptions [Lücking *et al.*, 2004]. Two phenomena can be recognised (even though they are blurred by over-specification which we observe very often in complex demonstrations). First, pointing saves words; definite descriptions accompanied by a pointing gesture are shorter and less complex than definite descriptions without gesture. Secondly, length and complexity of the definite description in complex demonstrations depend on the distance between demonstrating subject and referent pointed to. Similar results can be found in literature, e.g. [Beun and Cremers, 2001; van der Sluis and Krahmer, 2004].

These results indicate that the discriminative power of pointing gestures influences the construction of definite descriptions and that in order to determine the set of entities delimited by a pointing gesture the distance to the referent has to be accounted for. As a first approximation we model the topology of the region singled out by a pointing gesture as a cone anchored at the index finger tip and directed along the vector defined by the stretched index finger.

It has to be stressed, however, that a cone is an idealisation of the pointing region. There are a lot of influencing parameters, which we can divide in perceivable parameters on the one hand (like spatial configuration of demonstrating agent, addressee, and referents as well as the clustering of the entities under demonstration) and dialogue parameters on the other hand. Determining the pointing cone in more detail is the issue of further empirical investigations currently undertaken. The concept of pointing cone we use is based on a set

of parameters which guarantees that the cone's form and size can be adjusted as further findings become available.

Observations we made in our corpus suggest that we have to acknowledge that each of the two referential functions of pointing, i.e. object-pointing and region-pointing, comes with a cone on its own. Therefore, the concept of pointing cone can be divided into two topologically different types for object- and for region-pointing respectively, with the former having a narrower angle than the latter. The cone of object-pointing represents the resolution of a pointing gesture visually perceivable to the dialogue participants, and therefore, defines the borderline up to which object-pointing can be conducted successfully. Preliminary findings [Kühnlein and Stegmann, 2003] indicate an apex angle of this cone of about 12 to 24 degrees. In contrast, region-pointing draws the attention of the addressee to a wider region making the objects inside this region salient. The cone representing this region has to be modelled with a wider apex angle than the cone for object-pointing to ensure robust reference and to fit empirical findings concerning over-specification.

3 Related Work

While much work concerning the generation of referring expressions has been published over the last 15 years, work on the generation of multi-modal referring expressions is rare. Most of the approaches which can be found in this field use idealised pointing in addition or instead of referring expressions. [Claassen, 1992] and [Reithinger, 1992] highlight the referent in two-dimensional settings by an idealised pointing gesture represented by an arrow or a schematic hand. [Noma and Badler, 1997] and [André *et al.*, 1999] introduce virtual agents in presentation tasks able to produce simple pointing gestures. [Lester *et al.*, 1999] and [Rickel and Johnson, 1999] generate pointing gestures expressed by an agent which moves to the referent, and therefore, achieve unambiguous pointing. Only [Krahmer and van der Sluis, 2003] integrate pointing and definite descriptions in a more natural way and account for vague pointing. They distinguish three types of preciseness, i.e. *precise*, *imprecise*, or *very imprecise* pointing, and integrate pointing into the graph-based algorithm proposed by [Krahmer *et al.*, 2003].

Examining the generation of referring expressions realised as definite descriptions one has to mention, first of all, that the problem of selecting the minimal set (in the sense of Grice's quantity maxim) of object properties needed for an unambiguous description of the referent has exponential computational complexity [Reiter, 1990]. Each combination of properties has to be tested whether it is true only for the referent, and the shortest one of these combinations has to be chosen. Especially for real-time applications in domains with high object density and objects with a high number of properties this computation is intractable with brute-force methods. Several approaches have been proposed to deal with this problem, namely [Dale, 1992; Krahmer *et al.*, 2003; Horacek, 1997; Gardent, 2002]. [Dale and Reiter, 1995] proposed an incremental algorithm which violates the quantity maxim in the strict sense, but achieves linear compute time and fits well with empirical findings.

3.1 The Incremental Algorithm by Dale and Reiter

To achieve linear compute time, [Dale and Reiter, 1995] propose a fixed sequence of property evaluation and avoid backtracking. This approach leads to over-specification, but they can show that the generation results fit well with empirical findings if the sequence of properties is chosen accurately w.r.t. the specific domain. Therefore, the content-selection algorithm (see Alg. 3.1) gets, in addition to the referent r and the context set C, also a sorted list of properties P as an input.

The functionality of this algorithm can be described in short as follows. In the ordering of P each property A_i in P is evaluated concerning its discriminatory power, that means it is checked if there is at least one object in C which has another value for A than the referent r has. These objects are *ruled out*. If the contrast set C is empty the algorithm terminates and returns a list with the discriminating properties L. W.r.t. observations in their corpora Dale and Reiter add the property *type* everytime. The task of FINDBESTVALUE is the search for the most specific value of an attribute that (1), discriminates the referent r from more elements in D than the next general one does, and (2), is known by the addressee.

We chose this algorithm as a starting point for our work and adapted it for multimodal expressions because of its appropriateness w.r.t. empirical data and its efficient compute time .

Algorithm 3.1: MAKEREFERRINGEXPRESSION(r, C, P)

$L \leftarrow \{\}$
for each member A_i of list P **do**
  $V =$ FINDBESTVALUE(r, A_i, BASICLEVELVALUE(r, A_i))
  if RULESOUT((A_i, V)) $\neq$ nil
    then $L \leftarrow L \cup \{(A_i, V)\}$
         $C \leftarrow C \setminus$ RULESOUT((A_i, V))
  if $C = \{\}$ **then**
    if $(type, X) \in L$ for some X
      then return (L)
      else return ($L \cup \{(type,$ BASICLEVELVALUE($r, type$))$\}$)
return (failure)

procedure FINDBESTVALUE(r, A, $initial\text{-}value$)
  if USERKNOWS(r, $(A, initial\text{-}value)$) = true
    then $value \leftarrow initial\text{-}value$
    else $value \leftarrow$ no-value
  if ($more\text{-}specific\text{-}value \leftarrow$ MORESPECIFICVALUE($r, A, value$)) $\neq$ nil $\wedge$
    ($new\text{-}value \leftarrow$ FINDBESTVALUE($A, more\text{-}specific\text{-}value$)) $\neq$ nil $\wedge$
    (|RULESOUT($(A, new\text{-}value)$)| $>$ |RULESOUT($(A, value)$)|)
    then $value \leftarrow new\text{-}value$
  return ($value$)

procedure RULESOUT((A, V))
  if $V =$ no-value
    then return (nil)
    else return ($x : x \in C \wedge$ USERKNOWS(x, (A, V)) = false)

4 Incremental Multimodal Content Selection

We integrate in the incremental algorithm by Dale and Reiter an evaluation of the spatial property *location*, either to be uttered absolutely by a pointing gesture or to be expressed verbally in relation to other objects in speaker-intrinsic coordinates.

Before presenting the algorithm we first have to clarify the terminology used. Analogous to [Dale and Reiter, 1995], we

define the context set C to be the set of entities (physical objects in our scenario) that the hearer is currently assumed to be attending to. We also define the set of distractors D to be the set of entities from which the referent r has to be distinguished further on. At the beginning of the content selection process the distractor set D will be the context set C except the referent r; at the end D will be empty if content selection was successful. R represents the set of restricting properties found, each composed of an attribute-value pair.

P represents the ordered list of properties which the algorithm gets as additional input. Based on observations in our data we assume that referring to objects by pointing is the first choice in face-to-face dialogues, while expressing relative location is only used after basic properties like type or colour. Therefore, we get *absolut location*, *type*, *colour*, *size*, and *relative location* to be the list of properties which have to be evaluated concerning their discriminatory power.

Algorithm 4.1: CONTENTSELECTRE(r, P, C)

$R \leftarrow \{\}$
$D \leftarrow C$
$\alpha \leftarrow objectPointingConeApexAngle$
$\beta \leftarrow regionPointingConeApexAngle$
if REACHABLE?(r) (i)
then
- $R \leftarrow \{(location, \searrow)\}$
- $(\vec{h}, \vec{r}) \leftarrow$ GENERATEPOINTINGRAY(r)
- **if** GETPOINTINGMAP($(\vec{h}, \vec{r}), C, \alpha) = \{r\}$
  - **then return** $(R \cup \{type,$ GETVALUE$(r, type)\})$
  - **else** $D \leftarrow$ GETPOINTINGMAP($(\vec{h}, \vec{r}), C, \beta$)

for each $p \in P$ (ii)
do
- **if** RELATIONALPROPERTY?(p)
  - **then** $v \leftarrow$ GETRELATIVEVALUE(r, p, D)
  - **else** $v \leftarrow$ GETVALUE(r, p)
- **if** $v \neq null$ **and** RULESOUT(p, v, D) $\neq \{\}$
  - **then**
    - $R \leftarrow R \cup \{(p, v)\}$
    - $D \leftarrow D \setminus$ RULESOUT(p, v, D)
- **if** $D = \{\}$
  - **then**
    - **if** $(type, x) \in R$ for some x
      - **then return** (R)
      - **else return** $(R \cup \{type,$ GETVALUE$(r, type)\})$

return $(failure)$

procedure RULESOUT(p, v, D)
return $(\{x \mid x \in D \wedge$ GETVALUE$(x, p) \neq v\})$

The incremental content-selection in our algorithm (see Alg. 4.1) is organised in two main steps: First, see part (i), disambiguation of the referent by pointing is checked if the referent is visible for both participants. The decision, which kind of pointing, object-pointing or region-pointing, is appropriate is based on an evaluation of their discriminatory power. Object-pointing can only be used if the gesture is able to indicate the referent in an unambiguous manner. This is tested by generating a pointing cone with an apex angle of 12 degrees anchored in an approximated hand-position (covered in the functions GENERATEPOINTINGRAY(r) and GETPOINTINGMAP($(\vec{h}, \vec{r}), C, \alpha$) with the apex angle α). If only the intended referent r is found inside this cone, the algorithm terminates and referring can be done by object-pointing. Otherwise, region-pointing is evaluated using the same functions to narrow down the distractor set D to the objects found in the cone, now with the wider apex angle β.

For determining additional discriminating properties (see part (ii)) we use an adapted version of the incremental algorithm of Dale and Reiter described above. Each property p in P is evaluated concerning its discriminatory power. If it rules out some objects in D, these objects are deleted in D and p and its value v are added to R.

On the one hand we extend the original algorithm accounting for properties which are expressed in relation to other objects in the scene. On the other hand our algorithm is simplified in as much as in our prototypical implementation the FINDBESTVALUE function defined by Dale and Reiter is replaced by the cheaper function GETVALUE. We realise the search for the appropriate value on a specialisation hierarchy only for the special case *type* ("screw" instead of "pan head slotted screw" is used). If an appropriate value for *type* does not exist (this is the case for some aggregates under construction in our domain), *type* is uttered in an unspecific manner like "this part", the value v for the property *type* is then set to *object*, the most general value in the specialisation hierarchy. Analogous to [Dale and Reiter, 1995], *type* is added to R even if it has no discriminatory power. This complies with the most frequent kind of over-specification found in our empirical data.

For the other properties like *colour* we do not need such a sophisticated search on a specialisation hierarchy in our domain. We operate in a highly simplified domain with objects characterised by properties having only a few and well distinguished values perceivable by both dialogue participants. For the property *colour*, e.g., only the values *red*, *green*, *blue*, *yellow*, *purple*, *orange*, and *brown* exist.

In the following we describe the realisation of the essential modifications proposed in our approach in greater detail, the evaluation of the discriminating power of pointing and the consideration of relational properties.

4.1 Considering the Spatial Context: Object-pointing vs. Region-pointing

If we assume that the spatial context of the interaction determines the discriminatory power of pointing as described in Section 2 we have to anchor multimodal content-selection into this context. The central concept for this task is the pointing cone. It models the region which is indicated by the pointing gesture. The objects inside the cone can not be distinguished without further information.

In the course of our multimodal content-selection algorithm the generation of the pointing cone and the identification of the objects lying inside it is realised using the following functions:

- REACHABLE?(r): Tests if the referent r is visually available to both dialogue participants.
- GENERATEPOINTINGRAY(r): This function gets the referent r and computes a pointing ray which is represented by two vectors, its origin $\vec{h}$ located in the demonstrating hand and its direction $\vec{r}$ determined by the referent r.
- GETPOINTINGMAP($(\vec{h}, \vec{r}), C, \alpha$): This function (for details see Alg. 4.2) gets the pointing ray $(\vec{h}, \vec{r})$, a set of objects C, and an apex angle α and returns a sorted list of objects located inside the cone defined by $(\vec{h}, \vec{r})$ and α.

The decision criterion is the apex angle α. If the vector originated in $\vec{h}$ directed to $o \in C$ spans with the pointing ray an angle less than α o is said to be located inside the cone, otherwise not.

- GETPOSITION$(o, \vec{h})$: Computes the position of object o w.r.t. the position represented by $\vec{h}$, in this case the hand position.
- GETANGLE$(\vec{x}, \vec{y})$: Computes the angle between the vectors $\vec{x}$ and $\vec{y}$.
- INSERT(o, M, α): Inserts the object o in the map M in increasing order w.r.t. the angle α.

Algorithm 4.2: GETPOINTINGMAP$((\vec{h}, \vec{r}), C, \alpha)$

$M \leftarrow \{\}$
for each $o \in C$
do
- $\vec{x} \leftarrow$ GETPOSITION$((o, \vec{h}))$
- $\beta \leftarrow$ GETANGLE$(\vec{x}, \vec{r})$
- **if** $\beta \leq \alpha$ **then** INSERT(o, M, α)

return (M)

In the course of evaluating pointing, it is tested first whether the referent is reachable by both participants. In our application domain this implies whether r is a visible object lying on the table, the construction area. If this is the case, pointing in general is appropriate, the property *location* with the value $\searrow$ indicating a pointing gesture is added to the list of restricting properties R.

To decide whether object-pointing or region-pointing is appropriate, the pointing cones for these two kinds of pointing have to be generated. This is achieved by generating the pointing ray first using the function GENERATEPOINTINGRAY. To determine the origin of the pointing ray without synthesising a pointing gesture at this early point of time an approximated hand position is computed located in a typical distance in front of the body on a straight line between a point in-between the shoulders of the demonstrating agent and the referent r.

The pointing ray is used as an input for the function GETPOINTINGMAP which stores all objects inside the cone in a sorted map. First, this is done for a cone with the apex angle α, the cone for object-pointing. If this map contains at least one object besides the referent r, disambiguation based only on a pointing gesture is not possible. Region-pointing is then chosen to narrow down the set of distractors. Again the function GETPOINTINGMAP is used to determine the set of objects which are indicated by pointing, now by region-pointing. The wider apex angle β for the pointing cone of region-pointing is used to ensure robust reference.

4.2 Relational Object Properties

In our corpus we often found properties which are typically expressed in relation to other objects. The most frequent examples concern the properties *size* and *location* leading to descriptions like "the big object" respectively "the left object". The function RELATIONALPROPERTY?(p) tests for each property p if it is a property which can be expressed relationally. To evaluate these properties we use the function GETRELATIVEVALUE. This function (see Alg. 4.3) compares the absolute value of the referent's property p with the corresponding values of the objects in D. If the referent r holds the maximum or minimum of the values the function returns the according max or min value, e.g., *big* or *small* if the property is *size*. To do so, GETRELATIVEVALUE needs a partial order for each property. In our system this is implemented for *size* and *relative location*.

In the case of *size* we relate the property to the shape of the objects under discussion. *Shape* is a property often used on its own if the type of an object is unknown but it is difficult to handle in generation because the description of shape, especially for complex shapes, is highly ambiguous and subjective. However, in our corpus data aspects of shape can be often found as part of descriptions of *size*. This can be found if the shape of an object is characterised by one or two designated dimensions. For these objects *size* is substituted by, e.g., *length* respectively *thickness* ("long screw" is used instead of "big screw").

In the case of *relative location* we use a similar kind of substitution. The relative location is evaluated along the axes defining the subjective coordinate systems of the dialogue participants (left-right, ahead-behind, and top-down). E.g., GETRELATIVEVALUE returns left if the referent r is the leftmost located object in $D \cup \{r\}$.

The function GETVALUE(o, p) returns the absolute value v of the property p of the object o fetched from the knowledge-base. The search for an appropriate value on a specialisation hierarchy for the property *type*, as described above, is realised within this function.

Algorithm 4.3: GETRELATIVEVALUE(r, p, D)

$v_r \leftarrow$ GETVALUE(r, p)
if $min\{v \mid v =$ GETVALUE$(x, p) \wedge x \in (D \cup \{r\})\} = v_r$
then
- $v_{min} \leftarrow$ typically used $minValue(p)$
- **return** (v_{min})

if $max\{v \mid v =$ GETVALUE$(x, p) \wedge x \in (D \cup \{r\})\} = v_r$
then
- $v_{max} \leftarrow$ typically used $maxValue(p)$
- **return** (v_{max})

return $(null)$

5 Example

The following example illustrates the process of content-selection as it is realised by the described algorithm (Fig. 2): The starting point is a query concerning the reference to a specific object with the technical name *five-hole-bar-0* (Fig. 2a). This object lying on the table is visible to both dialogue participants, therefore pointing is appropriate and the property location with the value $\searrow$ indicating a pointing gesture is added to R. Now it has to be decided which kind of pointing is appropriate (Alg. 4.1, part (i)), that means whether pointing alone (object-pointing) yields the referent in an unambiguous manner. To do so, the pointing cone for object-pointing is generated. In this example the object density is high and more than one object is found inside this cone. Therefore, pointing alone does not yield the referent and region-pointing is evaluated next. This is illustrated in Fig. 2b) schematically:

The two ellipses mark the intersection of the pointing cones with the table, the smaller ellipse w.r.t. object-pointing, the bigger one w.r.t. region-pointing. The smaller ellipse covers two objects, that means pointing alone can not distinguish between these two objects, an additional definite description is needed. Region-pointing is used to narrow down the set of distractors C for the construction of the definite description. To make the multimodal reference consisting of pointing and definite description more robust (in analogy to the empirical findings) now a wider apex angle is used resulting in the bigger ellipse. The objects inside this bigger ellipse, the two bars *five-hole-bar-0* and *three-hole-bar-0*, a block, a screw, and a disc constitute the distractor set.

The second part of the algorithm determines the properties needed for the definite description. It starts with testing the property *type*. The type *five-hole-bar* is too specific, so the super-type *bar* is chosen. This property rules out all objects except the two bars (now C = {*five-hole-bar-0, three-hole-bar-0*}) and *type* with the value *bar* is added to R. The property *colour* is tested next; it has no discriminatory power concerning the two bars. But the following property *size* discriminates the two objects. The shape of bars is characterised by one designated dimension. Therefore, *size* is substituted by *length*. In our case the referent r has the maximum length of all objects in C, the property *length* with the value *long* is added to R. Now C contains only r, the algorithm terminates and returns $R = \{(location, \searrow), (type, bar), (length, long)\}$ (Fig. 2b).

Based on R, a pointing gesture directed to r is specified, the noun phrase ”die lange Leiste” (the long bar) is generated, and both are inserted into an utterance template (see Fig. 2c)). The complete utterance is synthesised and uttered by the agent Max (Fig. 2d).

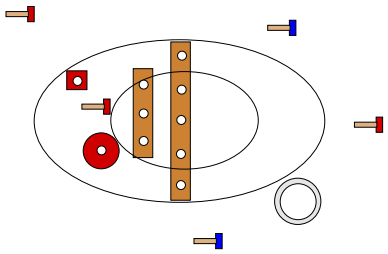
6 Application in the context of Human-Computer Interaction in VR

As explained in the introduction, the described approach was developed in the context of research on interfaces for natural interaction with an anthropomorphic agent in VR. The embodied agent Max should be enabled to produce believable deictic references to virtual objects in real-time interaction. Following [Dale and Reiter, 2000], the generation of natural language can be divided into three main steps, namely, macroplanning (document planning), microplanning, and surface realisation. Extending this, we add synthesis as a fourth step, including motorplanning and visualisation for gestural and a text-to-speech synthesis for verbal utterances. Content-selection for complex demonstrations is part of microplanning. The starting point is a logical representation of the performative of a planned utterance (as illustrated in the example above, see Fig. 2a)), which will be provided as result of the reasoning processes of the agent in future work.

The results of the content selection as represented by a list of attribute-value-pairs are fed into a surface realisation module generating a syntactically correct noun phrase. This noun phrase is combined with a gesture specification and both are inserted into a template of a multi-modal utterance fetched from a database and described in MURML [Kranstedt *et al.*,

a) **query(*five-hole-bar-0*)**

b)

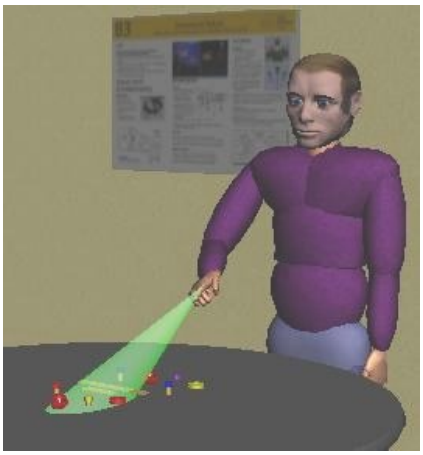


$R = \{(location, \searrow), (type, bar), (length, long)\}$

c)

```
<definition>
  <parameter name="NP"/>
  <parameter name="Object"/>
  <utterance>
    <specification>
     Meinst du <time id="t1"/> $NP? <time id="t2"/>
    </specification>
    <behaviorspec id="gesture_0">
     <gesture>
       <affiliate onset="t1" end="t2"/>
       <function name="refer_to_loc">
        <argument name="refloc" value="$Object"/>
        <argument name="frame_of_reference" value="world"/>
       </function>
     </gesture>
    </behaviorspec>
  </utterance>
</definition>
```

d)



“Meinst du die lange Leiste?”

(Do you mean the long bar?)

Figure 2: Example of the generation of a complex demonstration in four steps: a) A query concerning the object *five-hole-bar-o* constitutes the starting point; b) pointing cones for object-pointing and region-pointing are generated, the latter one specifies the distractor set for further property evaluation; c) the pointing gesture and the noun phrase are inserted in an utterance description template described in MURML; d) an appropriate animation (German speech, here with the visualised pointing cone) is synthesised.

2002] (see Fig. 2c) for illustration). MURML enables the specification of arbitrary co-verbal gestures. Cross-modal synchrony is established appending the gesture stroke to the affiliated word or sub-phrase in the co-expressive speech. Based on these descriptions, an utterance generator synthesises continuous speech and gesture in a synchronised manner (for details see [Kopp and Wachsmuth, 2004]).

The VR environment in which the interaction takes place is realised using the framework Avango [Tramberend, 1999] which is based on the common scenegraph representation of virtual worlds. With PrOSA (Patterns On Sequences of Attributes, [Latoschik, 2001] this framework was extended for interacting in immersive virtual reality by means of speech and gesture. The scenegraph is not only used to model the environment, it also builds the agent's knowledgebase of its environment. Each object represented in the scenegraph can be correlated with a so-called *semantic entity* [Latoschik and Schilling, 2003], which provides arbitrary semantic properties associated with this entity. During content-selection, the property values of the objects under discussion are fetched from these semantic entities.

The vocabulary used is geared to the ontology of the toy-kit, called *Baufix*, we use in our setting. It consists of a small number of generic parts like bars, screws, blocks, discs etc. (twelve different types, some of them in different size and colour). All the parts and the values of their properties can be named. Therefore, all possible descriptions in this small domain can be generated. Currently, deictic expressions as part of different types of speech acts can be generated, especially *query*, *request*, and *inform*. Only a small number of verb phrases can be used. In sum, the vocabulary currently available is very small. However, the focus of this work is not to generate a huge amount of speech output but to investigate the correlation between speech and gesture in the generation of multimodal reference.

Up to now we can generate in the course of deictic expressions pointing gestures synchronised with speech for all objects reachable for the agent without moving. In most cases moving will not be necessary, respectively more costly than generating a definite description. But we know that this is not adequate in all cases. The integration of moving in the course of content-selection will be an issue of future work.

7 Conclusion

In this paper an approach was presented which enables the generation of multimodal deictic expressions consisting of a pointing gesture indicating the location of an object and a definite noun phrase describing the object using its properties. Taking account of the inherent impreciseness of pointing gestures two referential functions of pointing are distinguished, object-pointing and region-pointing. With the increasing distance between demonstrating agent and referent the discriminatory power of the gesture decreases and more additional properties are needed to identify the referent. A pointing cone for each referring function of pointing gestures was defined to model the distance dependency of pointing. An algorithm was presented that integrates pointing and definite descriptions by using the objects highlighted by the gesture as distractor set for the construction of the definite description. Drawing the attention to a spatial region and the objects lying inside this region region-pointing ensures that these objects are in the focus of attention of the addressee ([Dale and Reiter, 1995] speak in this context about a navigational function of the expression).

Dale and Reiter emphasise that their content-selection algorithm is defined domain independently while the property list P and the functions MORESPECIFICVALUE, BASICLEVELVALUE, and USERKNOWS define the interface to the domain of application, especially to the knowledge about this domain shared by the interlocutors. Analogously, the functions REACHABLE?, GENERATEPOINTINGRAY, and GETPOINTINGMAP in our approach can be seen as a link between the content-selection algorithm and the spatial context in which the interaction takes place. Implementing the concept of the pointing cone they provide an interface between the geometrical aspects of pointing gestures and their referential semantics.

The quality of the generation results using the described approach depends on the precision of the topology of the pointing cones and the knowledge about the parameters influencing this topology. We have started to conduct empirical studies using tracking technology to collect analytical data concerning the pointing behaviour of human subjects in varying pointing domains [Kranstedt *et al.*, 2005].

Up to now, we do not have a comprehensive evaluation of our approach. But if we compare the generation results with the empirical data collected in the demonstration games mentioned in Sec. 1 and with other corpora about instructor-constructor dialogues in the *Baufix*-world [Sagerer *et al.*, 1994] we notice a good correspondence with the empirical findings. A critical point we found in these comparisons is that the perceivable resolution of pointing in real world is not exactly the same as in VR. In the latter it depends massively on kind and quality of the display technology used. Therefore, mechanisms which adapt the pointing cone's size and form to the constraints of the interaction environment seem to be useful.

Acknowledgment

This research is partially supported by the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Centre SFB 360.

References

[André *et al.*, 1999] E. André, T. Rist, and J. Müller. Employing AI Methods to Control the Behavior of Animated Interface Agents. *Applied Artificial Intelligence*, 13:415–448, 1999.

[Beun and Cremers, 2001] R.-J. Beun and A. Cremers. Multimodal Reference to Objects: An Empirical Approach. In *Proceedings of Cooperative Multimodal Communication, Second International Conference*, pages 64–86, 2001.

[Claassen, 1992] W. Claassen. Generating Referring Expressions in a Multimodal Environment. In R. Dale et al., editors, *Aspects of Automated Natural Language Generation*. Springer, Berlin, 1992.

[Dale and Reiter, 1995] R. Dale and E. Reiter. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 18:233–263, 1995.

[Dale and Reiter, 2000] R. Dale and E. Reiter. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, UK, 2000.

[Dale, 1992] R. Dale. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA, 1992.

[Gardent, 2002] C. Gardent. Generating Minimal Definite Descriptions. In *Proceedings of the 40th Annual Meeting of the ACL*, 2002.

[Horacek, 1997] H. Horacek. An Algorithm for Generating Referential Descriptions with Flexible Interfaces. In *Proceedings of the 35th Annual Meeting of the ACL*, 1997.

[Kopp and Wachsmuth, 2004] S. Kopp and I. Wachsmuth. Synthesizing Multimodal Utterances for Conversational Agents. *Comp. Anim. Virtual Worlds*, 15:39–52, 2004.

[Krahmer and van der Sluis, 2003] E. Krahmer and I. van der Sluis. A New Model for the Generation of Multimodal Referring Expressions. In *Proceedings of ENLG 2003*, 2003.

[Krahmer *et al.*, 2003] E. Krahmer, S. van Erk, and A. Verleg. Graphbased Generation of Referring Expressions. *Computational Linguistics*, 29(1):53–72, 2003.

[Kranstedt *et al.*, 2002] A. Kranstedt, S. Kopp, and I. Wachsmuth. MURML: A Multimodal Utterance Representation Markup Language for Conversational Agents. In *Proceedings of the Workshop Embodied Conversational Agents – let's specify and evaluate them, AAMAS 2002*, 2002.

[Kranstedt *et al.*, 2004] A. Kranstedt, P. Kühnlein, and I. Wachsmuth. Deixis in Multimodal Human Computer Interaction: An Interdisciplinary Approach. In A. Camurri and G. Volpe, editors, *Gesture-based Communication in Human-Computer Interaction*, pages 112–123. Springer (LNAI 2915), Berlin, 2004.

[Kranstedt *et al.*, 2005] A. Kranstedt, A. Lücking, T. Pfeiffer, H. Rieser, and I. Wachsmuth. Deixis: How to Determine Demonstrated Objects. Presented at Gesture Workshop 2005, Ile de Berder, France, 2005.

[Kühnlein and Stegmann, 2003] P. Kühnlein and J. Stegmann. Empirical Issues in Deictic Gesture: Referring to Objects in Simple Identification Tasks. Technical Report 2003/3, SFB 360, University of Bielefeld, 2003.

[Latoschik and Schilling, 2003] M. E. Latoschik and M. Schilling. Incorporating VR Databases into AI Knowledge Representations: A Framework for Intelligent Graphics Applications. In *Proceedings of the Sixth IASTED International Conference on Computer Graphics and Imaging*, pages 79–84, 2003.

[Latoschik, 2001] M. E. Latoschik. A General Framework for Multimodal Interaction in Virtual Reality Systems: PrOSA. In W. Broll and L. Schäfer, editors, *The Future of VR and AR Interfaces - Multimodal, Humanoid, Adaptive and Intelligent. Proceedings of the Workshop at IEEE Virtual Reality 2001*, pages 21–25, 2001.

[Lester *et al.*, 1999] J. Lester, J. Voerman, S. Towns, and C. Callaway. Deictic Believability: Coordinating Gesture, Locomotion, and Speech in Lifelike Pedagogical Agents. *Applied Artificial Intelligence*, 13(4-5):383–414, 1999.

[Lücking *et al.*, 2004] A. Lücking, H. Rieser, and J. Stegmann. Statistical Support for the Study of Structures in Multimodal Dialogue: Inter-rater Agreement and Synchronisation. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog 04)*, pages 56–64, 2004.

[Lyons, 1977] J. Lyons. *Semantics (2 Vols.)*. Cambridge University Press, Cambridge, UK, 1977.

[Mann, 1988] W. C. Mann. Dialogue Games: Conventions of Human Interaction. *Argumentation*, 2:512–532, 1988.

[McNeill, 1992] D. McNeill. *Hand and Mind*. University of Chicago Press, Chicago, Illinois, 1992.

[Noma and Badler, 1997] T. Noma and N. Badler. A Virtual Human Presenter. In *Workshop on Animated Interface Agents: Making them Intelligent, IJCAI'97*, pages 45–51, 1997.

[Peirce, 1965] C. S. Peirce. *Collected Papers of Charles Sanders Peirce*. Cambridge University Press, Cambridge, UK, 1965.

[Reiter, 1990] E. Reiter. The Computational Complexity of Avoiding Conversational Implicatures. In *Proceedings of the 28th Annual Meeting of the ACL*, pages 97–104, 1990.

[Reithinger, 1992] N. Reithinger. The Performance of an Incremental Generation Component for Multi-modal Dialog Contributions. In R. Dale et al., editors, *Aspects of Automated Natural Language Generation*. Springer, Berlin, 1992.

[Rickel and Johnson, 1999] J. Rickel and W. L. Johnson. Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control. *Applied Artificial Intelligence*, 13:343–382, 1999.

[Rieser, 2004] H. Rieser. Pointing in Dialogue. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog 04)*, pages 93–101, 2004.

[Sagerer *et al.*, 1994] G. Sagerer, H. Eikmeyer, and G. Rickheit. "Dies ist ein runder Gegenstand". Technical report, University of Bielefeld, SFB 360, 1994.

[Tramberend, 1999] H. Tramberend. Avocado: A Distributed Virtual Reality Framework. In *Proceedings of IEEE Virtual Reality 1999*, pages 14–21, 1999.

[van der Sluis and Krahmer, 2004] I. van der Sluis and E. Krahmer. The Influence of Target Size and Distance on the Production of Speech and Gesture in Multimodal Referring Expressions. In *Proceedings of the 8the International Conference on Spoken Language Processing*, 2004.