Towards Generating Referring Expressions in a Mobile Robot Scenario

Hendrik Zender and Geert-Jan M. Kruijff*
Language Technology Lab, German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken Germany
{zender,gj}@dfki.de

Abstract

This paper describes an approach towards generating referring expressions that identify and distinguish spatial entities in large-scale space, e.g. in an office environment, for autonomous mobile robots. In such a scenario the dialogue context typically goes beyond the perceptual fields of the interlocutors. One of the challenges therefore lies in determining an appropriate contrast set. Another important issue is to have adequate models of both the large-scale spatial environment and the user's knowledge.

Introduction

In earlier work, we have presented a conversational autonomous mobile robot (Zender et al., 2007), emphasizing situated dialogue for teaching the robot about its environment. Besides understanding human-like concepts the robot must be able to express itself in a way that is understandable by humans. It is therefore crucial that the robot can produce expressions that successfully refer to entities in its environment.

Previous approaches to the generation of referring expressions (GRE) in the general domain of conversational agents have mainly focused on small-scale scenes or closed-context applications, (Kelleher and Kruijff, 2006), (Funakoshi et al., 2004), (Horacek, 1997), (Dale and Reiter, 1995). Although there are well-established methods for generation referring expressions from both explicit and implicit scene models, only limited research has focused on how to determine what part of a scene constitutes the current context. This is of special importance when conducting a situated dialogue about *large-scale space*, where large-scale space is defined as "a space which cannot be perceived at once" (Kuipers, 1977). For the dialogue this means that most potential referents and distractors are not in the visual fields of the interlocutors, but still they will want to talk about them.

In this paper, we present an approach to adapt the *incremental algorithm* (Dale and Reiter, 1995) to a scenario where a conversational robot has to refer to spatial entities in large-scale space. We will show how our approach of Conceptual Spatial Mapping (Zender and Kruijff, 2007) both provides a suitable knowledge base for the algorithm and serves as a basis for determining the context set.

* The research reported of in this paper was supported by the EU FP6 IST Cognitive Systems Integrated project *Cognitive Systems for Cognitive Assistants* "CoSy" FP6-004250-IP.

Background

The task of generating referring expressions can be paraphrased as finding a description for an entity in the world (the *intended referent*) that refers to the intended referent and only the intended referent. This implies that the description must be chosen in a way that prevents it from referring to another entity in the current *context set*. All entities in the context set except the intended referent form the *contrast set*. The referring expression must thus distinguish the intended referent from the members of the contrast set. A referring expression is a noun phrase (NP) of any degree of complexity. In order to provide enough information to uniquely identify the intended referent, further attributes of the referent need to be expressed, for instance with adjectives or prepositional phrases, which in turn might contain a referring expression NP.

The *incremental algorithm* of (Dale and Reiter, 1995) constitutes an approach to the GRE problem, which they rephrase in terms of the *Gricean Maxims*. Inherently, any referring expression should fulfill the Maxim of Quality in that it should not contain any false statements. The algorithm also ensures that only properties of the referent that have some discriminatory power are realized (Maxim of Relevance). Moreover, they try to fulfill the Maxims of Manner and Quantity in that the produced expressions are short and do not contain redundant information. The incremental algorithm provides a solution to the GRE problem with a reasonable run-time complexity. They support the fact that this is achieved by not attempting to find an optimal referring expression by findings in psycholinguistics.

Since we are going to present our approach in terms of the incremental GRE algorithm (cf. Algorithms 1, 2, 3), it is important to briefly explain its relevant principles. The algorithm needs a knowledge base that describes the *proper-*

ties of the domain entities through *attributes* and *values*. A special attribute is an entity's *type*. The algorithm is initialized with the *intended referent*, a *contrast set* and a list of *preferred attributes*. The algorithm then iterates through the list of attributes in the order of preference. If a property that holds for the intended referent is false for at least one member of the contrast set, the property is added to the generated expression and the respective members of the contrast set are removed from it. When the algorithm has successfully eliminated all the original members from the contrast set, the algorithm terminates and returns the expression generated so far. If the contrast set is still non-empty after iterating over the last property in the list, the algorithm fails.

In order to determine appropriate discriminating properties, the algorithm requires a set of interface functions that provide further information, namely the *taxonomical specialization* of a given attribute, the *basic level category* of an entity's attribute, a model of the *user's knowledge*, and finally an ordered list of *preferred attributes*.

Algorithm 1 The Basic Incremental Algorithm for GRE

```
Require: r = intended referent; C = contrast set; P = preferred-attributes-list
  Initialize: DESC = {}
  for each A_i ∈ P do
    V = findBestValue(r, A_i, basicLevelValue(r, A_i))
    if RulesOut(⟨A_i, V⟩) ≠ nil then
      DESC = DESC ∪ {⟨A_i, V⟩}
    end if
    if C = {} then
      if ⟨type, X⟩ ∈ DESC for some X then
        return DESC
      else
        return DESC ∪ {⟨type, basicLevelValue(r, type)⟩}
      end if
    end if
  end for
  return failure
```

Algorithm 2 findBestValue(r,A,initial-value)

```
  if userKnows(r, ⟨, initial − value⟩) then
    val = initial-value
  else
    val = null
  end if
  if (more-specific-value = moreSpecificValue(r, A, val)) ≠ nil ∧
  (new-value = findBestValue(A, more-specific-value)) ≠ nil ∧
  (|rulesOut(⟨A, new-value⟩)| > |rulesOut(⟨A, val⟩)|) then
    val = new-value
  end if
  return val
```

Algorithm 3 rulesOut(⟨A, V⟩)

```
  if V = null then
    return nil
  else
    return {x : x ∈ C ∧ userKnows(x, ⟨A, V⟩) = false}
  end if
```

Our approach

A robotic office assistant that is supposed to interact with its users through spoken language will have to refer to things and places in their environment. It needs to do so in a way that is intuitively understandable by humans. There are conceivably many ways in which a robot might to refer to things in the world and most of them will fail to achieve their communicative goal. Consider the following set of examples:

1. "the location at position $(X = 5.56, Y = -3.92, \theta = 0.45)$"
2. "the mug to left of the plate to the right of the mug (...)"
3. "Peter's office no. 200 at the end of the corridor on the third floor of the Acme Corp. building 3 in the Acme Corp. building complex, 47 Evergreen Terrace, Calisota, Planet Earth, (...)"
4. "the area"

These referring expressions are valid descriptions of their respective referents. Still they fail to achieve their communicative goal, which is to specify the right amount of information that the hearer needs to uniquely identify the referent. First of all, robots are good at measuring exact distances, humans are not. So the robot should employ qualitative descriptions that make use of the same concepts as a human-produced utterance would. Second, specifying a referent with respect to another referent that is only identifiable relative to the first one leads to infinite recursion instead of the communicative goal. Finally, the robot might have a vast knowledge about facts and entities in the world, but it should not always try to uniquely separate the referent from all entities in the world. At the same time, it is necessary to provide enough information to distinguish the intended referent from those entities in the world that potentially distract the hearer. The following expressions *might* serve as more appropriate variants of the previous examples:

1. "the kitchen around the corner"
2. "the mug on the table in the laboratory"
3. "Peter's office"
4. "the large hall on the first floor"

The fact that they *might* (or *might not!*) be successful referring expressions points to the importance of knowing what the given context in a situation is. This is especially the case for a mobile robot that operates and interacts in large-scale space. It is thus an important basis to endow the robot with a spatial representation that resembles the way humans conceive of their environment. But it is not enough; the robot must also be able to determine which entities in the world might act as *potential distractors* with respect to the hearer's knowledge.

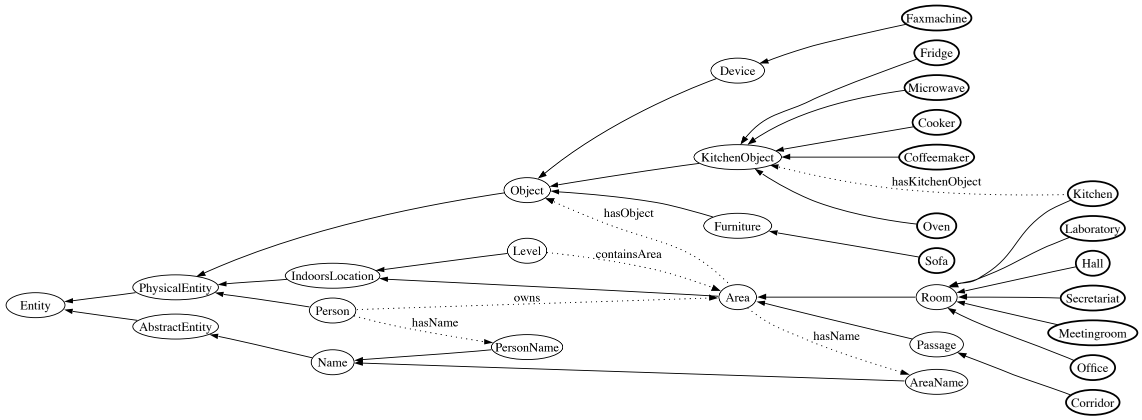

Figure 1: A subset of our office environment commonsense ontology, including relevant relations (dotted arrows) and basic level categories (thick lines).

In the following sections we will describe how the ontological representation of spatio-conceptual knowledge in our robot architecture serves as a natural knowledge base for the incremental GRE algorithm. Furthermore, we will propose a method for a proper construction of the *contrast set* for large-scale space.

The knowledge base

Our robotic system is endowed with a *multi-layered spatial representation*, ranging from a low-level metric map, via a topological abstraction layer, to an ontology-based conceptual map. We refer the reader to our previous publications for a more detailed account on the spatial representation (Zender and Kruijff, 2007; Zender et al., 2007). Here, we will focus on describing the relevant mechanisms for the GRE task in large-scale space.

The *conceptual map* layer consists of a Description-Logics based OWL-DL reasoner. It contains innate conceptual commonsense knowledge about an indoor office environment (Figure 1), i.e. types of spatial areas, objects and persons, and the relations that can hold between them. While the robot is exploring its environment, it populated its ontology with acquired and inferred instance knowledge.

The **instances** in the ontology are the *entities* of the world model. The **conceptual hierarchy** provides the taxonomical *type* information of the instances that the GRE algorithm requires. Furthermore, a number of concepts such as `Office`, `Kitchen`, `Corridor`, `Table`, etc. are marked as basic level categories, cf. (Brown, 1958) and (Rosch, 1978). The **relations between instances** are the *attributes* that the algorithm can use to further specify a referent. Note that the use of relations leads to a recursive call of the GRE algorithm with its well-known implications. An extension of the algorithm with heuristics to exclude reference to an already mentioned entity and to keep the recursion depth minimal can be used to cope with this issue. Currently, our approach supports the following properties (in the order of preference):

Type We represent an entity's type as the (asserted and inferred) concepts of the corresponding instance. Through ontological reasoning, we can retrieve an instance's most specific concept, its basic level category, and all the instances of a concept. Further, taxonomy traversal functions (e.g. *getSuperConcepts*,*getSubConcepts*) can provide additional type information if necessary.

Topological inclusion If the current context spans topological units at different hierarchical levels (cf. Figure 2) it is important to specify the intended referent with respect to the topological unit that contains the referent, e.g. when referring to "the kitchen on the 3rd floor", or "the table in the lab". The conceptual map represents topological position with the following relations: `hasObject(Area,Object)`, and `containsArea(Level,Area)`.

Ownership Areas in an environment are often referred to by identifying their owners, e.g. "Bob's office". In our ontology instances of `Area` can be related to a `Person` instance via the `isOwnedBy(Area,Person)` relation. The name of the person is represented by relating the `Person` instance with an instance of `PersonName` via `hasName(Person,PersonName)`.

Topological connectivity Information about neighboring areas can be a good cue for identifying spatial entities, e.g. "the room next to the lab". Our system represents adjacency of topological areas in the topological layer of the robot's multi-layered map, where the information can be accessed.

Name As names are usually (locally) unique, e.g. "the Occam meeting room", or "office 120", they are definitely a highly discriminating attribute for the GRE task. However, names don't seem to be a preferred category for referring to

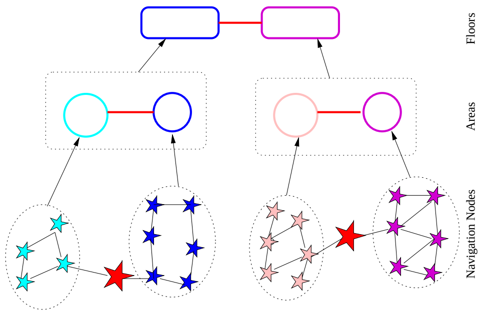

Figure 2: A topology of places, rooms and floors. Stars depict navigation nodes that denote free and reachable space for our robotic system. The set of navigation nodes is partitioned into distinct spatial areas, such as e.g. rooms. Areas in turn can belong to a floors, which are on the next level of abstraction. Using *topology traversal*, we construct an appropriate context set for the GRE task.

rooms as they seldom contain any more useful information than a generic NP + PP referring expression, e.g. "the meeting room on the first floor next to the large hall", that might bear additional useful information. Moreover, remembering the inherently artificial name for an entity might involve a higher cognitive load than processing the information encoded in a more generic referential description. For other scenarios though, such as an information desk agent at a hospital, or any other institution in which rooms are usually named by numbering them in logical order, the name feature can conceivably be placed in a higher-ranking position in the preference list. Names for areas are represented through the `hasName(Area,AreaName)` relation in our ontology.

Landmarks The conceptual map contains spatial areas in the environment as well as objects found there. The information about which objects are found there can be used to further specify a spatial entity. Currently, our model only provides the information which areas contain which objects (`hasObject(Area,Object)`). The GRE algorithm can thus form expressions like "the room with the mailboxes". Since most of these objects will only be perceivable once the person is at the location of the intended referent, we assume that this attribute should only be used with a low preference. It is noteworthy that our DL-reasoner is able to categorize spatial areas on the basis of the objects that are found there (Zender and Kruijff, 2007). The knowledge about which objects are found where is thus reflected already in the type information, which is always used in the process of generating a referring expression.

Determining the appropriate context set

In order to successfully identify a referent it is important to determine a correct and appropriate contrast set. If the contrast set is chosen too small, the hearer might find it difficult to uniquely identify the intended referent with respect to his or her knowledge. If, on the other hand, a too large contrast set is assumed, the generated referring expression might violate *Grice's Maxims*, here the Maxim of Quality, in that it contains too much unnecessary information.

We claim that the contrast set for generating referring expressions to entities in large-scale space can be determined on the basis of a topological representation. Assuming a (potentially recursively defined) topological hierarchy of places, the contrast set should include all sibling nodes of those topological units that are visited when following the search path between the current position and the intended referent (**topology traversal**). For instance, if the intended referent is an object located in the same room as the user and the robot, only local landmarks should be considered. Likewise, if the robot is to produce a referring expression to a room on a different floor, all entities on that floor and on the current floor will form the contrast set. Using topological inclusion as the most preferred attribute will then essentially function as a pruning of the hierarchically ordered context set. If the intended referent is within an area of the same topological hierarchy, however, this feature will not be used at all because it has no discriminatory power.

In our implementation, the lowest topological level is the navigation graph. The set of navigation nodes is then partitioned into topological areas that correspond to basic spatial units, such as rooms and corridors. Our ontology additionally contains a representation for dividing areas into storeys to which they belong, cf. Figure 2. The topological unit that is considered the current position need not necessarily be the robot's and/or the hearer's physical location. We claim that our approach will also yield plausible results when used in an incremental dialogue to generate route descriptions. In that case, the most recent dialogue referent is assumed as the initial position.

Representing the user's knowledge

In the incremental algorithm the *userKnows* function is used to ensure that the algorithm does not include descriptions that the hearer does not understand and also to prevent the algorithm from ruling out members of the contrast set that are no potential distractors with respect tot he user's knowledge. In our scenario, it is infeasible to fully specify the knowledge of all possible interlocutors. We therefore opt for *a priori* assuming an omniscient user. Using a dialogue model, we can explicitly mark information as not known by the user when, e.g. answering questions.

Moreover, the representation of the user's knowledge playS important role for example in the route description generation task. There, the *UserKnows* function should initially return false for any knowledge pertaining to referents that have not yet been introduced. The task of generating a route description is then redefined in terms of successively introducing new discourse referents that can then be used for the GRE task.

Natural language processing

In our system, we use a communication system for situated spoken dialogue between the robot and a user. Our implementation of the GRE algorithm collects information from the ontology that it will then represent as a Hybrid Logics Dependency Semantics (HLDS) logical form (Baldridge and Kruijff, 2002). This HLDS logical form is the processed by the OpenCCG realizer, which generates a natural language expression (Baldridge and Kruijff, 2003). The following list shows how information from the ontology is translated to HLDS. The logical form generated by the *type* attribute is the root node; logical forms representing other attributes are dependent structures of a root node.

- HLDS logical form for *type*:
  $@_{\{X:\mathsf{entity}\}}(\langle\mathbf{TYPE}\rangle$
  $\quad \& \ \langle Delimitation\rangle\mathbf{unique}$
  $\quad \& \ \langle Number\rangle\mathbf{singular})$

- HLDS logical form for *topological inclusion* (of areas):
  $\langle Location\rangle(\mathbf{on}$
  $\quad \& \ \langle Anchor\rangle(location \ \& \ \mathbf{floor}$
  $\quad\quad \& \ \langle Delimitation\rangle\mathbf{unique}$
  $\quad\quad \& \ \langle Number\rangle\mathbf{singular}$
  $\quad\quad \& \ \langle Property\rangle(q-position \ \& \ \langle\mathbf{ORD}\rangle))))$

- HLDS logical form for *topological inclusion* (of objects):
  $\langle Location\rangle(\mathbf{in}$
  $\quad \& \ \langle Anchor\rangle(\langle\mathbf{REFERRING\ EXPRESSION}\rangle))$

- HLDS logical form for *topological connectivity*:
  $\langle Location\rangle(\mathbf{next\ to}$
  $\quad \& \ \langle Anchor\rangle(\langle\mathbf{REFERRING\ EXPRESSION}\rangle))$

- HLDS logical form for *ownership*:
  $\langle GenOwner\rangle(person \ \& \ \langle\mathbf{NAME}\rangle)$

- HLDS logical form for *name*:
  $\langle Identifier\rangle(loc-name \ \& \ \langle\mathbf{LOCATION\ NAME}\rangle)$

- HLDS logical form for a number as *name*:
  $\langle Identifier\rangle(number \ \& \ \langle\mathbf{LOCATION\ NUMBER}\rangle)$

- HLDS logical form for *landmark*:
  $\langle Accompaniment\rangle(\langle\mathbf{REFERRING\ EXPRESSION}\rangle))$

Examples

Let us consider the example scenario depicted in Figure 3. For visualization purposes we have annotated a map sketch with the instance knowledge that is represented in the conceptual map. The knowledge base consists of a number of areas that are anchored in the topological map layer. The robot knows that the rooms in its environment are numbered (`<X>`), and that two of the meeting rooms additionally have names (`''Occam'' and ''Goedel''`). Additionally, the robot has learned the types of the areas through situated dialogue with its user (Kruijff et al., 2007). It knows about the presence of four objects, and, finally, the robot knows two persons, and in which offices they work.

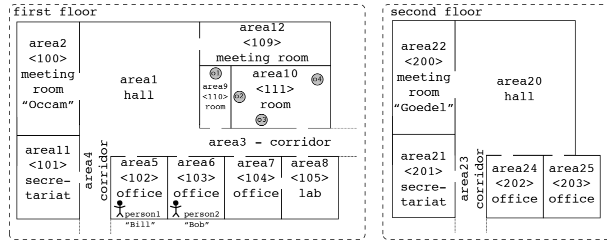

Figure 3: An example office environment knowledge base. The ontology contains several instances of areas (indicated by the `areaX` tag) on two floors, objects (`o1` is an instance of `Faxmachine`, `o2` and `o4` are instances of `Couch`, and `o3` is a `Coffemaker`), and persons (`Bill` and `Bob`).

The following examples are the results of applying the algorithm under varying circumstances. The initial position and the intended referent are denoted by $i =$`area`$_i$ and $r =$`area`$_r$, respectively.

(1) $i =$`area3`; $r =$`area1`;
Since initial position and target location are on the same floor, the expression "the hall" is produced.
$@_{\{area1:\mathsf{e-location}\}}(\mathbf{hall}$
$\quad \& \ \langle Delimitation\rangle\mathbf{unique}$
$\quad \& \ \langle Number\rangle\mathbf{singular})$

(2) $i =$`area3`; $r =$`area20`;
Since initial position and target location are on different floors, the expression "the hall on the second floor" is produced.
$@_{\{area20:\mathsf{e-location}\}}(\mathbf{hall}$
$\quad \& \ \langle Delimitation\rangle\mathbf{unique}$
$\quad \& \ \langle Number\rangle\mathbf{singular}) \quad \& \ \langle Location\rangle(\mathbf{on}$
$\quad\quad \& \ \langle Anchor\rangle(location \ \& \ \mathbf{floor}$
$\quad\quad\quad \& \ \langle Delimitation\rangle\mathbf{unique}$
$\quad\quad\quad \& \ \langle Number\rangle\mathbf{singular}$
$\quad\quad\quad \& \ \langle Property\rangle(q-position \ \& \ \mathbf{2})))$

The system is able to successfully generate a referring expression (Ex. 3) for the coffee maker (`o3`), but not for any of the couches (`o2` and `o4`) because the knowledge base does not contain any information that can properly distinguish between the two.

(3) $i =$`area3`; $r =$`o3`;
The position of the intended referent is anchored in the navigation graph topology. The context set thus spans two topological layers (the navigation graph and the area layer). Hence the algorithm includes the location information when generating the referring expression "the coffee maker in the kitchen".

Ex. 3 shows a weakness of the proposed algorithm. The basic level category `Kitchen` for `area10` is inferred on the basis of the presence of a `Coffeemaker` instance. However, when generating the referring expression for the "coffee maker" this is not taken into account. It remains an issue of further research to what extent this influences the acceptability of such a referring expression.

Another observation can be made when generating a referring expression for `area4` with different initial positions. However, if a recalculated contrast set is provided as input for the recursive call to generate an embedded referring expression, we can avoid redundant attributes. Here again, we make use of our principle to determine the context on the basis of the topological hierarchy. We simple assume the position of the most recent referent as the initial position when determining the contrast set. The result of this modification leads to the result in Ex. 6

(4) i =`area1`; r =`area4`;
This configuration yields "the corridor next to the secretariat".

(5) i =`area20`; r =`area4`;
The unmodified algorithm yields "the corridor on the first floor next to the secretariat on the first floor".

(6) i =`area20`; r =`area4`;
After the modification the algorithm produces "the corridor on the first floor next to the secretariat".

Conclusions

In this paper we have presented an approach to applying the *incremental algorithm* for GRE to the domain of large-scale space, with an emphasis on its application in a mobile robot office assistant scenario. We have shown how our method of *conceptual spatial mapping* provides a knowledge base for the GRE algorithm. We have argued further that the construction of the *context* and *contrast* sets using our method for topology traversal is an important step towards generating appropriate referring expressions in large-scale space. More importantly, the same method can be used in scenarios where the robot has to provide a verbal route description from a given start position to a target location. The representation of the user's knowledge is another important parameter for the route description task where new discourse referents have to be introduced sequentially in order to allow for the generation of appropriate referring expressions.

Future work

In our current approach, the list of preferred attributes is static. Other approaches, e.g. (Kelleher and Kruijff, 2006), have shown that a dynamic ordering of attributes based on their (relative) salience yields better results. It remains an issue of future work to explore the effect and measurability of different kinds of salience (e.g. visual and discourse salience) in the context of GRE for large-scale space. A first approach could be to work with different preference lists for different types of referents (e.g. objects vs. areas).

The aforementioned approach of (Kelleher and Kruijff, 2006) provides an excellent opportunity for integrating qualitative spatial reasoning for small-scale space with the more allocentric conceptual spatial reasoning method of the approach presented in this paper. Using the method of topology traversal, the robot could conceivably produce referring expressions that incorporate entities and properties at different levels of abstraction, thus leading to a *"zooming-in-and-out"* behavior, like e.g. "the ball to the left of the box on the table in the kitchen on the third floor". Since both approaches are compatible in that they build upon on the same basic *incremental GRE algorithm*, we claim that the capabilities of our robot to refer to entities in the world can be significantly improved by combining these approaches.

References

Baldridge, J. and Kruijff, G.-J. M. (2002). Coupling CCG and hybrid logic dependency semantics. In *Proc. of the 40th Annual Meeting of the ACL*, pages 319–326, Philadelphia, PA, USA.

Baldridge, J. and Kruijff, G.-J. M. (2003). Multi-modal combinatory categorial grammmar. In *Proc. of the 10th Conference of the EACL*, Budapest, Hungary.

Brown, R. (1958). How shall a thing be called? *Psychological Review*, 65(1):14–21.

Dale, R. and Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Funakoshi, K., Watanabe, S., Kuriyama, N., and Tokunaga, T. (2004). Generation of relative referring expressions based on perceptual grouping. In *Proc. of COLING '04*, Geneva, Switzerland.

Horacek, H. (1997). An algorithm for generating referential descriptions with flexible interfaces. In *Proc. of the 35th Annual Meeting of the ACL and 8th Conf. of the EACL*, Madrid, Spain.

Kelleher, J. and Kruijff, G.-J. (2006). Incremental generation of spatial referring expressions in situated dialogue. In *In Proceedings of Coling-ACL '06*, Sydney, Australia.

Kruijff, G.-J. M., Zender, H., Jensfelt, P., and Christensen, H. I. (2007). Situated dialogue and spatial organization: What, where…and why? *International Journal of Advanced Robotic Systems*, 4(1):125–138.

Kuipers, B. J. (1977). *Representing Knowledge of Large-scale Space*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B., editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Zender, H., Jensfelt, P., Óscar Martínez Mozos, Kruijff, G.-J. M., and Burgard, W. (2007). An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. of AAAI-07*, pages 1584–1589, Vancouver, BC, Canada.

Zender, H. and Kruijff, G.-J. M. (2007). Multi-layered conceptual spatial mapping for autonomous mobile robots. In *Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems*, AAAI Spring Symposium 2007.