Why a Hill Can't be a Valley: Representing Gestalt and Position Properties of Objects with Object Schemata

Kai-Uwe Carstensen/Geoff Simmons
University of Hamburg
Computer Science Department

Introduction*

What, one might ask, is the difference between a hill and a valley? There must be a crucial one if we believe our intuition and agree with the objection registered by Lewis Carroll's Alice:

"When you say 'hill'," the Queen interrupted, "*I* could show you hills, in comparison with which you'd call that a valley."
"No, I shouldn't," said Alice, surprised into contradicting her at last: "a hill *can't* be a valley, you know. That would be nonsense——" [1]

When, one might ask further, does such a difference become important for a text comprehension system?
In this paper we want to argue that an answer to both questions is provided if one considers the use of *dimensional adjectives* with respect to the object nouns:

(1)	a.	high hill	b.	deep valley
	c.	*high valley	d.	*deep hill

The examples in (1) show

- that these adjectives are sensitive to gradable *gestalt* and *position* properties of spatial objects in that they *designate* respective object axes as distinctive *dimensions*
- that the representation of these features is relevant for a text comprehension system as the expressions in (1) are natural language expressions which such a system should accept or reject.

Obviously we need an adequate semantic theory of dimensional adjectives which is related to our perception-based knowledge of objects and their spatial contexts.
In the following we present the core of such a theory (developed by Lang 1989a) which treats *dimensional designation* of spatial objects and in which the relevant object features are comprised in structures called *object schemata*. We then describe some extensions of the theory gained by working with OSKAR[2] (Lang / Carstensen 1989,1990) and outline the realization of this theory in the representational formalism of LILOG.

---

* Our thanks to Ewald Lang for his meticulous scrutiny and valuable advice.

[1] L. Carroll, Through the Looking Glass. In: M. Gardner (ed.), The Annotated Alice. Penguin Books Ltd.: Harmondsworth, Middlesex 1987, p. 207.

[2] OSKAR is the acronym of **O**bjekt-**S**chemata zur **K**onzeptuellen **A**nalyse **R**äumlicher Objekteigenschaften (Object schemata for the conceptual analysis of spatial properties of objects) - a Prolog-program developed by Ewald Lang and Kai-Uwe Carstensen.

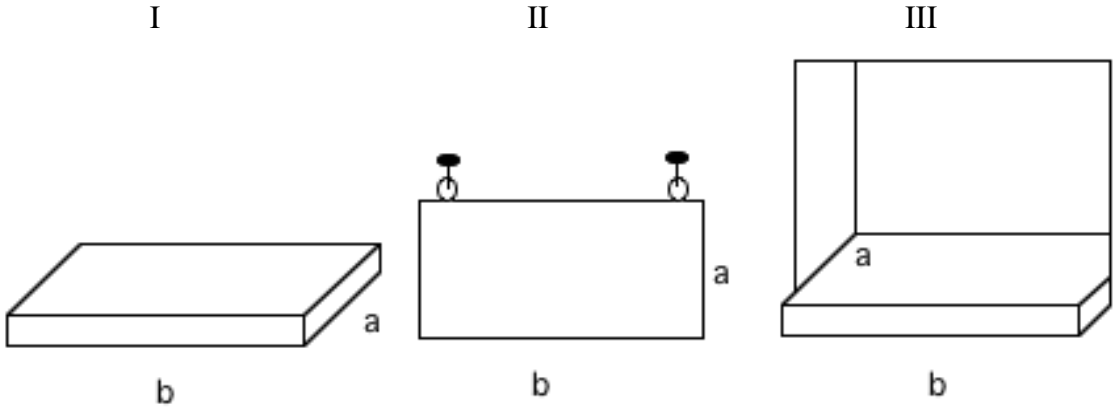
1. Overview: Relevant Aspects of Dimensional Designation

Before we describe Lang's theory in some detail we have to elaborate on the need for a sophisticated semantics of dimensional adjectives (DAdjs). Why not simply annotate object concepts with features for height, length, shortness etc.? A few remarks shall be made to answer this question:

First. There can be no one-to-one relation between DAdjs and object features as some of the former constitute pairs of antonyms, i.e., opposites on a scale belonging to only one object feature (long / short -> length). This kind of *polarity* is a general linguistic phenomenon (cf. good / bad) that should not be directly represented in object concepts.

Second. There is a distinction to be made between *object constitutive* and *contextually induced* properties which has to be accounted for in a principled and consistent way. For example, while both hills and towers are high by default, (only) towers can be said to be long (when they are lying down); while poles and streets are always long, (only) poles can be said to be high in certain contexts; poles cannot be deep and spherical objects can neither be deep nor high.

Third. There is no way to avoid modelling the gestalt properties of an object, as DAdjs can denote different object extents according to the context in which they are used (see the illustration below). For the same reason, such a referential access to an object model is neccessary to allow for quantification of the respective object extents, i.e. *graduation* of the DAdjs as in the board is 40cm wide.



wide = a; long = b | wide = b; high = a | wide = b; deep = a

2. Dimensional Designation of Objects

General framework

The semantics of dimensional adjectives concerns the following group of words:

(2) a. lang - kurz breit - schmal dick - dünn

long - short { wide / broad } - narrow thick - thin

b. hoch - niedrig tief flach

high - low deep shallow

c. *weit - eng* d. *groß - klein*

{ wide / broad } - narrow { big - small / large - small / tall - short }

According to Lang, these adjectives refer to *axes* of a spatial object, which are determined and qualified by its gestalt and position properties, thereby designating them as *dimensions* of that object. Leaving out the details, the simplified schema of the semantic form of DAdjs is the following:

(3) [[QUANT DIM x] = VAL[3]]

QUANT and DIM represent the two relevant semantic components of a DAdj, i.e., the graduation and the dimensional designation component, respectively. QUANT is a semantic prime for a scaling operation which assigns a scale value to some spatial object x relative to a dimension d (which again is the value of DIM x). DIM is a variable for a limited set of constants (MAX, SUB, DIST, VERT, OBS, ACROSS, see below), which are to be interpreted as functions yielding d.

This approach to the semantics of dimensional designation adheres to the paradigm of cognitive linguistics (cf. Bierwisch / Lang 1987, 1989)[4], where the modularity of cognitive behavior is explicitly reflected within the semantic theory for natural language expressions by taking into account the interaction of the various cognitive systems and subsystems.

One aspect of this is the interaction of the grammar G (the language system embodying linguistic knowledge) and the conceptual system C, which is intermodally accessible and therefore serves as a mediator between language and perception.

Theoretically, G is said to be in an *inherent* relationship with C, i.e., G contains parameters which are instantiated by structures of C. Since semantic representations like (3) constitute the interface between G and C, the constants mentioned above function as parameters (so-called *Dimensional Assignment Parameters* (DAPs)), and are instantiated by their corresponding conceptual values (so-called *Dimensional Assignment Values* (DAVs)). The latter can be viewed as those features of object concepts that represent prominent gestalt and position properties.

---

[3]According to Bierwisch 1989, VAL itself has the internal structure [v $\pm$ c] where v is a norm or comparison value, c is the difference of the scale value to v, and $\pm$ is a variable for the operations + and - which yield the different polarities of the DAdjs.

[4]Note that within this approach the meaning of an expression is characterized by the interaction of two different representational levels: the *semantic* and the *conceptual* level. See Maienborn (in this volume) for some aspects of this interaction. In the case of DAdjs, (3) corresponds to the semantic level.

Therefore, they are said to be *perceptually based* and *conceptually categorized* whereas the DAPs are *conceptually motivated* but *grammatically coded*, which means that the way DAPs are couched in lexical expressions allows for certain differences among languages (cf. groß vs. tall, large, big).
With this two-level semantic approach, we are able to model the (in)validity of the following inferences, which differ from grammatically coded converses (as for example, x is longer than y ⟷ y is shorter than x):

(4) a. The pole is 20m high/tall ⟶ The pole is 20m long
b. The tower is 20m high/tall ⇏ *The tower is 20m long

It is not within lexical semantics but on the conceptual level where the inferences in (4) come out to be valid or not. This is to be realized by rules that determine which DAP of a DAdj can be instantiated by which conceptual DAV. Obviously, the same holds for the interpretation of *the hill is deep and *the valley is high and for the aforementioned distinction between object constitutive and contextually induced spatial properties.
Note that with this approach, the semantics of natural language expressions is more than just a subdiscipline of linguistics; it has a well-defined interface to other cognitive sciences and therefore can be accessed, discussed, used and applied in the area of AI. Additionally, the primitives of the semantic representation no longer constitute a symbol system in its own right - and can only be interpreted in terms of the system itself - (as in the Katz/Fodorian Semantic Marker framework). Rather, they are anchored in the various cognitive modules, ultimately being connected to the external world.

Gestalt and Position Properties of Objects
According to Lang, the categorization of percepts as spatial objects can be described by two interacting sets of principles: the so-called *Inherent Proportion Schema* (IPS) which defines the gestalt properties of an object and the so-called *Primary Perceptual Space* (PPS) which defines, among others, a system of axes within which the gestalt properties of an object can be interpreted as position properties.

The relevant principles underlying the IPS include those responsible for
- the *delimination* of an object against the background,
- the perception of *symmetry axes* (which are referred to by DAdjs) and
- *axial disintegration* (how possible symmetry axes can be discerned: a brick has three disintegrated axes while a ball only has one integrated axis),
- *salience/prominence* (how axes are ordered within an object according to their size, yielding a *proportion schema* of the object) and
- *penetrability* (which accounts for the different ranges of the DAdjs thick and wide / weit)

We can now list and give an interpretation for the DAPs of the DAdjs relating to *gestalt properties* of objects:

MAX identifies the maximal disintegrated axis of some object *x* , which in turn presupposes that there is exactly one such axis of *x* available (long / short).

SUB identifies either a non-maximal disintegrated third axis (cf. thick board) or an integrated axis forming the diameter of a circular section (cf. thick pole).

DIST identifies an object axis perceived as inside diameter of a hollow body (cf. wide hole).

The PPS, an internal model of how we reconstruct external physical space on the basis of perceptual information from upright walk, equilibrum and eye level, is defined by the following three axes:

- the *Vertical* axis (ubiquitous and constant)
- the *Observer* axis
- the *Horizontal* axis (dependent on, i.e. defined by orthogonality to, the other two axes)

An object, then, is assigned a *position property* if one of its axis extensions defined by IPS is redefined by being projected onto an axis of the surrounding space as determined by PPS. Here are the corresponding DAPs:

VERT selects exactly that disintegrated axis of an object which coincides with the Vertical of PPS (high / low).

OBS identifies any disintegrated axis of an object which coincides with the Observer axis of PPS (deep).

Additionally, there is the parameter ACROSS which selects an axis dependent on another axis that is identified by either MAX or VERT or OBS (wide / breit - narrow / schmal).[5] Taken together, VERT and OBS can be used to describe how objects are *oriented* and *perspectivised*, respectively.

Not surprisingly, DAVs are simply names representing the clusters of conditions defining the range of the DAPs.[6] Together with (representations of) the other gestalt properties described above, the DAVs are arranged into complex structures called *object schemata*.

Object Schemata

[5]Due to this inherent relativity there is no object constitutive DAV for ACROSS.

[6]To distinguish them from the DAPs, DAVs are written in small letters: max, sub, dist, vert, obs etc. The DAV ∅ represents an unspecified desintegrated axis of an object and can be regarded as the 'landing site' for the contextual DAPs VERT, OBS and ACROSS. There is another DAV, diam, which conceptually fills a lexical gap regarding dimensional designation. Yet, although it has no adjectival counterpart it might correspond to expressions like has the diameter of.

Very generally, object schemata (OS) are designed as matrices with rows and colums. The columns, called *sections*, contain relevant information about the *axes* of an object which can be designated by DAdjs. They are arranged according to the decreasing order of the axis extents, thereby modelling the *proportion* of that object. In the first row, the general gestalt properties are encoded: the *dimensions* of the object are named by the constants a, b and c, *boundedness* of an axis is represented by (the scope of) '<...>' and the *integratedness* of an axis is represented by (the scope of) '(...)'. The second row contains the primary ('object constitutive') DAVs, the third row - divided by a horizontal bar - the contextually induced DAVs.

(5)

'tower'	'pole'	'high pole'
< a (b c) >	< a (b c) >	< a (b c) >
max sub	max sub	max sub
vert		vert

The OS in (5) illustrate the similarity in the gestalt of towers and poles. Note that towers have a *canonical* vertical orientation while the orientation of a pole must always be contextually induced. This leads us back to the interaction of DAPs and DAVs, i.e., the interpretation of DAPs with respect to an OS: DAPs can either *identify* (cf. long pole) or *specify* (cf. high pole) a DAV in an OS. *Compatibility conditions*, which emerge from the inherent relationship between the *gestalt* and *position properties* determine possible OS and possible interpretations, while ruling out, for example, natural language expressions like (1)-(c)/(d) as ill-formed.
Finally, just a glimpse at the OS in (6) and (7) suffices to confirm Alice' suspicion that there is no simple way to compare hills and valleys, and proves that there is more to gestalt and position properties than 'height'- and 'depth'-features in object concepts.

(6)

'hill'
< (a b) c >
diam vert

(7)

'valley'
< a b c >
max ∅ vert
obs

3. OSKAR

The Prolog-program OSKAR, developed in the 'rapid-prototyping'-style, was originally intended to be a means for testing the theory of dimensional designation with respect to consistency (no incorrect designations) and completeness (exhaustive applicability to spatial objects). This was achieved, and it turned out to be a useful proceeding for discovering some minor points to be improved. To name just two examples, *inherent* orientation and perspectivation had to be explicitly represented as additional DAVs, and *canonical* and *fixed* orientation had to be distinguished.
On the other hand, the aspect of 'sidedness' of objects was taken into account from the beginning, in that information about object constitutive ('intrinsic') or contextual ('deictic') sides became part of the OS. This gave rise to a number of possibilities and new ideas for representing conceptual knowledge inside and outside the theory of dimensional designation:

a) procedures for a principled simulation of *position variation* of objects were developed, i.e., the axis- and side-related aspects of 'tilting' and 'turning' were represented
b) contextual specification was redefined as a device for specifying an object's reference to the PPS (i.e., *positioning* or *perspectivising* an object): with this, 'setting upright' and 'laying down' could be identified as subtypes of the positioning of objects
c) *position properties* like those addressed by expressions like stands, lies, upside down, reversed were represented
d) moreover,the *movability* of an object was found to be a prerequisite for position variation and the position properties listed in c). The movability features of an object (immobile and movable) thus derive from the structure of the entries of the OS at issue.

The Prolog prototype OSKAR, having implemented, supplemented and tested the theoretical foundation (cf. Lang / Carstensen 1989, 1990), provides us with a specification for integrating dimensional designation and positional variation into LILOG.

4. Dimensional Designation and Positional Variation in LILOG

The integration of Lang's theory of dimensional designation and positional variation into the LILOG system proceeds in two steps:
(I) The theory must be realized in L_{LILOG}, the representational formalism of the LILOG system. With OSKAR as a prototype, this is a relatively straightforward process.
(II) The integration into LILOG must account for a number of new issues that are directly affected by dimensional designation, such as:
(a) Mechanisms of *inheritance* that regulate the relationship between OS defined for classes of objects and OS assigned to individiual RefOs.
(b) A treatment of *context* and context change.
(c) A realization of the scalar function QUANT (cf. (3) above), to be integrated into the *semantic form* of scalar adjectives.
In the remainder of this article, we will focus on the first of these two steps: the implementation of dimensional designation and positional variations in the representation language L_{LILOG}. It will be necessary to make decisions about the representation of inheritance and context dependence; problems related to the semantics of DAdjs and verbs of position will be remarked on briefly.[7]

5. OS and Object Ontology in L_{LILOG}

Object schemata (OS) categorize objects into classes with respect to their gestalt and position properties; we will exploit the feature logic of L_{LILOG} to reconstruct OS in LILOG. In L_{LILOG}, spatial objects are classified as subsorts of the sort OBJECT, distinguishing them from other kinds of entities like events. We will represent the gestalt and position properties for

[7]The syntax assumed in this paper for the language L_{LILOG} is implemented in the LEU/2 prototype; its semantics is loosely based on the specification in Pletat/von Luck 1989.

subsorts of OBJECT (e.g. HILL, VALLEY) by taking OS as complex feature structures. The sort declarations will define object constitutive OS information that is assumed by default for each object in a class; thus the ontology specifies the gestalt properties and canonical position properties for each class of spatial objects. However, because the interpretation of OS of some particular object within a class is context dependent, we will realize the assignment of the pertinent OS to a specific object by means of a temporally-indexed LLILOG function (see section 6).[8]

We begin by defining DAPs and DAVs in LLILOG. As one might expect, these are just atoms collected in a special sort called DIMDESIGNATION, which in turn is a subsort of the sort SPATIALCONCEPT, an extremely general and unspecified sort that merely serves to keep theoretical notions of spatial knowledge separate from everything else in the sort lattice.[9]

(8)
```
sort DimDesignation < SpatialConcept;
      atoms   max, vert, sub, dist, obs, across, imax,
              ivert, iobs, diam, empty.
```

The values prefixed with "i" are the DAVs mentioned in section 3 that represent "inherent" orientation and perspectivation, and `empty` corresponds to ∅.

The LLILOG sort OBJECT has a feature `has_default_schema`, which is of the sort OBJECTSCHEMA. Thus a portion of our sort declaration for OBJECT is:[10]

(9)
```
sort Object < SpatialEntity;
  features has_default_schema : Objectschema,
           ....
```

The sort OBJECTSCHEMA is also a subsort of SPATIALCONCEPT; its features specify a dimensionality (an integer between 1 and 3) and a list of entities of the sort SECTION:[11]

---

[8]A note on typography: names of sorts will be given in SMALL CAPS in the running text; names of features and any portion of LLILOG code will be written in the text font `Courier`.

[9]Having DAVs and DAPs in the same sort is a technical expedient (it facilitates unification). It is up to the knowledge engineer to see to it that DAPs and DAVs are not confused (for example, `across` can only be contextually induced, and thus cannot appear in the ontology).

[10]We will elaborate on the "default" status of this OS in section 6. – Lang describes the notion of a "basic schema" (*Grundschema*), which is not reflected in this paper. Lang's basic schemata treat problems of *proportional variation* within an object class, which are important to our treatment of inheritance in LLILOG. Due to lack of space, we will not comment further on this point (cf. Lang 1987, 1989 a,b, Lang/Carstensen 1990).

[11]A list in LLILOG is inductively defined as in PROLOG; a list is either a special object called the "empty list" (represented in LLILOG with the constant `nil`), or it consists of a feature `head`, which can be of any sort, and a feature `rest`, which is another list.

```
(10) sort Objectschema < SpatialConcept,
     features   dimensions  : [1..3],
                sections    : List_of_Sections.
```

The `sections` feature of an OS in LILOG represents the disintegrated or integrated axes of an object (written in (5)-(7) above as "a", "b", "(a b)", etc.). At this level we have very little information about object axes; things get interesting when we look at the sort declaration for SECTION itself:

```
(11) sort Section < SpatialConcept;
      features   number_of_dims : [1..3],
                 davs           : List_of_Davs,
                 degree         : SpatialDegrees.
```

The sort SECTION defines the content of a section of an OS as shown in (5)-(7) above. In the ontology, the feature `davs` is assigned a list of the object constitutive primary entries of an OS; contextually induced DAVs may be appended to the list in the course of processing a text (cf. the second and third rows of (5)-(7) above).
The feature `number_of_dims` specifies the number of dimensions taken up by the axis represented by a section, thus indirectly representing the axis' integratedness. Finally, the feature `degree` relates the object axis to an entity of the sort SPATIALDEGREES, which may in turn be used in a realization of the function QUANT given in (3) above (cf. section 7).
Given these sort definitions, we can define the OS of an object class by assigning appropriate values to feature paths in the sort declaration for that class. Returning to our familiar examples, we can now give a portion of the sort declarations for HILL and VALLEY (somewhat simplified for expository economy), which are the LLILOG counterparts to (6) and (7) above (would Lewis Carroll have been startled, or intrigued?).

```
(12) sort hill < and (Object,
                  has_default_schema :
                    and(dimensions : {3},
                        <sections head> :
                            and( number_of_dims : {2},
                                 <davs head> : {diam},
                                 <davs rest> : {nil}),
                        <sections rest head> :
                            and( number_of_dims : {1},
                                 <davs head> : {vert},
                                 <davs rest> : {nil}),
                        <sections rest rest> : {nil} )).
```

```
(13) sort valley < and (Object,
                        has_default_schema :
                         and(dimensions : {3},
                             <sections head> :
                                 and( number_of_dims : {1},
                                      <davs head> : {max},
                                      <davs rest> : {nil}),
                             <sections rest head> :
                                 and( number_of_dims : {1},
                                      <davs head> : {empty},
                                      <davs rest> : {nil}),
                             <sections rest rest head> :
                                 and( number_of_dims : {1},
                                      <davs head> : {obs},
                                      <davs rest head> : {vert},
                                      <davs rest rest> : {nil}),
                             <sections   rest   rest   rest>   :
                  {nil})).
```

6. Inheritance and Context Dependent Assignment of Object Schemata

The sort declarations in (12) and (13) define context invariant gestalt properties and canonical position properties of objects. But as shown in section 1, dimensional designation is context dependent, and thus so is the processing of OS; in particular, this concerns the assignment of *positional properties and variations* to individual objects. When an object enters discourse, the OS determined by its sort is assumed; for example, if a tree is mentioned in a text, we assume that the tree is standing (and hence that the OS appropriate for vertical orientation is valid) unless we have explicit evidence to the contrary. If the object undergoes a manipulation in its position, or if the assumption about its position is explicitly contradicted, then a new OS', based on the original OS and appropriate to the new position, is assigned to that object.

To cope with the context dependence and default status of OS in LILOG, we will create RefOs of the sort OBJECTSCHEMA in L_{LILOG}, which are assigned to objects by means of the temporally indexed L_{LILOG} function `has-os`. This function has the following arguments and sortal restrictions:

(14) `function has-os(O:Object, T:Interval)-> Objectschema.`

When a RefO of the sort OBJECT is introduced into discourse, it is assumed by default to be assigned an OS that is identical with its default schema. Modifications of that initial OS may be due to positional specification (e.g. the pole is 2m tall entails the pole's upright position), or positional change (e.g. the tree has been felled entails the loss of the tree's canonical verticality). Both result in an OS' reflecting the object's new position; in the latter case, the OS' is associated with a new temporal index.

In declaration (14), INTERVAL is the sort of temporal intervals. Temporal intervals are the entities proposed for LILOG for the treatment of tense and aspect (cf. Eberle 1988, 1989). In order to see how we arrive at a value for the argument T in `has-os`, we must take a closer look at verbs that are related to the positional properties of objects. In the LILOG proposal for

the semantics of *verbs of position*, stehen, liegen, sitzen, etc. (stand, lie, sit, etc.) are classified as *static*; stellen, legen, setzen etc. (put / place, lay, set, etc.) as their *causative* derivatives (cf. Maienborn 1989 a,b). In addition to specifying a local argument and tense and aspectual information, the meaning of each of these verbs determines a *mode of position* - a characteristic relation between the object's axes and the Vertical and/or Observer axis of the surrounding space, as reflected in the occurrence of vert and/or obs in the object's OS.[12]
For the static verbs, the mode of position is realized as an evaluation of matching conditions with the OS of the object in question. Now it is clear that the OS reflecting a certain static position is constant for a certain object just as long as that object remains in that position. This means that the temporal index T for the usage of a static verb coincides with the index T associated with the object's position.
The causative verbs (e.g. legen, etc.) indicate a change of state, for which a new OS' is created and assigned to the object with a new temporal index T'. The new OS' reflects the resulting mode of position valid at T' (e.g. liegen). The modification of OS accounting for positional variation is implemented in OSKAR (cf. section 3), and these procedures are easily adapted as L_{LILOG} rules. The treatment of temporal intervals is determined entirely by the analysis of tense and aspect for the verbs in question.
Thus we have a context dependent assignment of OS to objects that makes use of temporal information inferred by the LILOG system. We can treat references to periods of time when different OS were valid for a given object. Consider the sentence:

(15) While the tree was still standing, it was 5m tall.

If the tree is lying on the ground "now" (t_o), this implies that the dimensional designation tall, which is applicable at some t_i before t_o, is no longer acceptable at t_o.

[12]This characterization of the OS that are associated with the modes of position is slightly oversimplified. See Lang/Carstensen 1990 for details.

7. Dimensional Designation and Scalar Functions

Now that we have seen how object schemata are defined for object sorts and bound to RefOs in LLILOG, we can take a very brief look at the proposed implementation of the process of dimensional designation in LLILOG. As described in section (2), dimensional designation is a process by a which a DAP is interpreted for a given object in a given context; a DAV in the OS of that object in the given context must be identified or specified, thus locating the axis of the object that has been designated. Borrowing the terminology of OSKAR, we will call this process the *evaluation* of a DAP. Since we have added a temporal index in LILOG as a contextual property, we will assume that evaluation is also dependent on a temporal location; that is, a DAdj will be evaluated with respect to the temporal information of the sentence in which it is uttered.
We now define in LLILOG a function `eval_DAP`. The evaluation of this function starts a set of LLILOG rules that identify or specify a DAV in the given OS, and return the section representing the axis that is designated by the DAP, much as this was done in OSKAR.

(16)
```
function eval_DAP(DAP:DimDesignation, OS:Objectschema)
                                        -> Section.
```

The function `eval_DAP`, together with the feature `degree` on SECTION, allows us to account for sentences like (17) and (18):

(17) This hill is 10m high.
(18) This hill is higher than that hill.

We will assume a theory of DAdjs as *degree adjectives*. In such a theory, sentence (17) may be treated by assigning a measurement value to a degree, and sentence (18) may be treated by placing the degrees of height of the mentioned objects in an ordering relation. There are quite a few theories of this kind on the market (cf. Bierwisch 1989 or von Stechow 1985). We can produce a LILOG representation that is amenable to all of these theories by interpreting the feature `degree` as a scalar function.
Leaving out the details, the essence of the lexical interpretations of DAdjs in such a theory is the QUANT function given in (3) above. If we add a temporal index T to that function, then we can define a LILOG interpretation for QUANT as follows:

(19)
```
forall X:Object, DIM:DimDesignation, T:Interval,
          D:Degree;
    QUANT (DIM, X, T) = D
    <->
    has-os(X, T) = OS
    and
    degree(eval_DAP(DIM, OS)) = D.
```

This means that the scale value assigned to an object extent for a DAP DIM at T is equal to the value of the feature `degree` for the OS section returned by `eval_DAP`.

In this paper, we have briefly outlined a three-step process that begins with linguistic theory, which is then confirmed and enhanced in a specific technical prototype (OSKAR), and is finally integrated into a comprehensive knowledge representation system (LILOG). All in all, the insights thus gained and rendered in formal representation may well account for the core of the conditions and principles organizing human knowledge of spatial objects.

References

Bierwisch, M.: The Semantics of Gradation. In: Bierwisch/Lang 1989 : 71 - 261.

Bierwisch, M. / Lang, E. (eds.): Grammatische und konzeptuelle Aspekte von Dimensionsadjektiven. Berlin: Akademie-Verlag 1987

Bierwisch, M. / Lang, E. (eds.): Dimensional Adjectives: Grammatical Structure and Conceptual Interpretation. Berlin-New York: Springer-Verlag 1989

Eberle, K.: Eine Prolog-Theorie für zeitliche Beziehungen zwischen Ereignissen. LILOG-Report 14. 1988

Eberle, K.: Quantifikation, Plural, Ereignisse und ihre Argumente in einer mehrsortigen Sprache der Prädikatenlogik erster Stufe. IWBS Report 67. 1989

Lang, E. (1987): Semantik der Dimensionsauszeichnung räumlicher Objekte. In: Bierwisch / Lang 1987 : 287 - 458

Lang, E. (1989a): The Semantics of Dimensional Designation of Spatial Objects. In: Bierwisch / Lang 1989 : 263-417

Lang, E. (1989b): Primärer Orientierungsraum und inhärentes Proportionsschema: Interagierende Kategorisierungsraster bei der Konzeptualisierung räumlicher Objekte. In: Habel, Herweg, Rehkämper (eds.), Raumkonzepte in Verstehensprozessen. Tübingen: Max Niemeyer Verlag, 1989

Lang, E. / Carstensen, K.-U. (1989): OSKAR - ein Prolog-Programm zur Modellierung der Struktur und Verarbeitung räumlichen Wissens. In: GWAI '89. Springer Verlag

Lang, E. / Carstensen, K.-U.: OSKAR - A Prolog Program for Modelling Dimensional Designation and Positional Variation of Objects in Space. IWBS-Report 109. 1990

Maienborn, C. (1989a): Zur Semantik von Bewegungs- und Positionsverben - Perspektiven der kognitiven Linguistik. LILOG Report 64. 1989

Maienborn, C. (1989b): Semantische und Konzeptuelle Analyse der Bedeutungskonstitution: Verbalphrasen der Bewegung und Lage. Magisterarbeit, Universität Hamburg 1989

Pletat, U. / von Luck, K.: Knowledge Representation in LILOG. IWBS Report 90. 1989

von Stechow, A. (1985): Comparing Semantic Theories of Comparison. In: *Journal of Semantics* 3, pp. 1 - 77