Incremental production of preverbal messages with INC

Abstract:

INC (*incremental conceptualiser*) is a cognitively motivated model of the first stage of language production, conceptualisation. Starting with basic conceptual representations of an observed moving object in a very reduced setting, provided by a perceptual pre-processing unit, INC produces *preverbal messages* that have the same structure as verbalisations of humans performing the same task. Architecture and processing mechanisms of INC are based on simple, cognitively motivated principles. Its behaviour is controlled by parameters that are set according to the available resources. After presenting the principles of the INC architecture and exemplifying the construction of a conceptual representation we go into the details of selecting conceptual entities for verbalisation and how the generation algorithm is determined by parameter settings.

Title:

Incremental production of preverbal messages with INC

Authors:

Markus Guhe (guhe@informatik.uni-hamburg.de)
Christopher Habel (habel@informatik.uni-hamburg.de)

Address:

Research Group Knowledge and Language Processing (WSV),
Department for Informatics, University of Hamburg,
Vogt-Kölln-Straße 30, D-22527 Hamburg, Germany

Corresponding author:

guhe@informatik.uni-hamburg.de

Incremental production of preverbal messages with INC

Markus Guhe (guhe@informatik.uni-hamburg.de)
Christopher Habel (habel@informatik.uni-hamburg.de)
Ladina Tschander (tschander@informatik.uni-hamburg.de)
Research Group Knowledge and Language Processing (WSV),
Department for Informatics, University of Hamburg,
Vogt-Kölln-Straße 30, D-22527 Hamburg, Germany

Abstract

INC (*incremental conceptualiser*) is a cognitively motivated model of the first stage of language production, conceptualisation. Starting with basic conceptual representations of an observed moving object in a very reduced setting, provided by a perceptual pre-processing unit, INC produces *preverbal messages* that have the same structure as verbalisations of humans performing this task. Architecture and processing mechanisms of INC are based on simple, cognitively motivated principles. Its behaviour is controlled by parameters that are set according to the available resources. After presenting the principles of the INC architecture and exemplifying the construction of a conceptual representation we go into the details of selecting conceptual entities for verbalisation and demonstrate how the generation algorithm is determined by parameter settings.

Introduction

In contrast to language comprehension (Lewis, 1999) only few proposals for the architecture of the human language production faculty follow the methodological idea of computational architectures (Anderson, 1983, Newell, 1990), viz. to combine empirical investigation and computational modelling. The *incremental conceptualiser* INC, which we described in Guhe & Habel (2001), is a computational model of the first part of the human language production architecture according to Levelt (1989). It is located between a perceptual *pre-processing unit* (PPU) and the *formulator* in which linguistic encoding takes place. In Guhe & Habel (2001) we demonstrated how the behaviour of INC depends on resource parameters. In the present paper we report on five major enhancements of the model supporting the fundamental hypotheses underlying the INC-model:

1. By changing to another domain, namely the description of motion events, we got evidence for the domain independency of the model.
2. In contrast to earlier versions of INC, that generated sequences of preverbal messages in the sense of Levelt (1989), INC now produces preverbal messages of a finer grade of incrementality, called *incremental preverbal messages* (Guhe, Habel & Tschander, 2003; Guhe, in prep.).
3. Each element in the internal *current conceptual representation* (CCR) now has an activation value. The CCR is the INC counterpart to working memory models, as used in ACT-R or Soar. The activation influences selection of the content to be verbalised. The algorithm for generating incremental preverbal messages contains a constraint that evaluates the activation value. Usually, the element is chosen if it is above the activation threshold (AT)—a new resource parameter—while it is not selected if it is below the threshold.[1]
4. INC operates in quasi-real time. (Even though it still reads its input from a previously generated file it now makes it computations with respect to the time intervals in which the events took place.)
5. INC supports another selection strategy that attempts to retain the granularity level it has started with. Only if nothing else is selected, items of another than the current level of granularity is chosen.

The *incremental conceptualiser* INC

General overview

In this section we give an updated account of the main components and functionality of INC (Guhe & Habel, 2001). Its input is a stream of visuo-spatial percepts, called *perceived entities*[2]. The pre-processing unit operates according to the empirically founded *cut-hypothesis* of Avrahami & Kareev (1994, p. 239): 'A sub-sequence of stimuli is cut out of a sequence to become a cognitive entity if it has been experienced many times in different contexts.' Perceived entities are used to build up a hierarchical conceptual structure, from which elements are selected for verbalisation. In this way INC forms preverbal messages out of perceived entities. The newly introduced sorts of entities for the domain of describing motion events are presented in the next section.

The processes of incremental models are often depicted as a cascade of processes (Levelt, 1989): like water in a water cascade splashing down from one level to

[1] Elements that have only a low activation value are selected if they are conceptually/semantically required in the preverbal message.

[2] In Guhe & Habel (2001) we called these entities *basic entities*. However, in order not to confuse them with basic *level* entities and to emphasise the variability of the items they represent we decided to rename them.

the next, information is splashing down from one process to the next. This symbolises the simultaneous processing of information on subsequent stages. In INC four processes form this cascade: *construction*, *selection*, *linearisation*, and *PVM-generation* (preverbal message generation), cf. figure 1. All processes operate on the *current conceptual representation* (CCR), which contains a representation of the current state of affairs, e.g. the representation shown in figure 3. This representation is a network of interrelated concepts. It consists of so-called *referential objects* (refOs) and *descriptions*. See the next section for a more detailed explanation of the formalism.

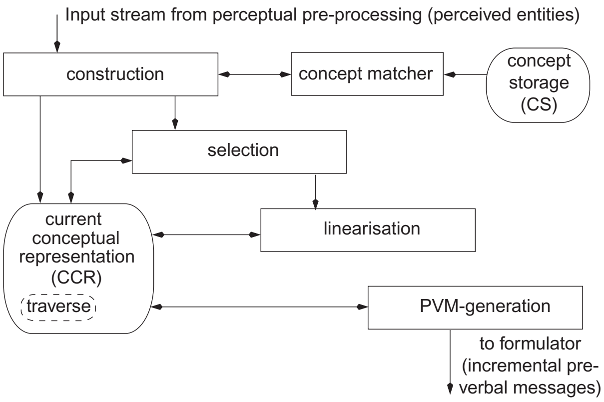

Figure 1: The incremental conceptualiser INC

The *construction* process builds up the CCR from which *selection* chooses the situations to be verbalised. Thus, it induces a path through the CCR, which we call the *traverse*. *Linearisation* brings the selected situations into an appropriate order, and *PVM-generation* generates preverbal messages for the chosen situations. Since we have no need for a linearisation here, because we do not investigate concurrent events up to now, e.g. two moving planes, linearisation is not yet implemented and plays only a minor role here.

Construction

The construction process reads the perceived entities provided by the pre-processing unit and builds up the CCR from them. It is supported by the *concept matcher*: each time a new perceived entity or a modification for an already existing one[3] is read from the pre-processing unit, construction calls the concept matcher to determine the best matching concept from the *concept storage*. The concept matcher has three possible results:

1. it finds an entity that subsumes the new entity and some 'surrounding' ones, which are determined by a heuristic,
2. it finds an analysis for the entity, i.e. it finds simpler concepts,
3. it finds nothing.

In the first two cases the concept matcher determines the *best match* by computing the *degree of agreement* (DoA) between the entity (entities) to be matched and the concepts stored in the concept storage. The best match is the concept with the highest DoA. A pair consisting of best match and DoA is given back to construction. The best match in case 1 need not be complete; that is, even if parts of this concept that are required for a complete match have not—yet—been perceived it can nevertheless be introduced into the CCR, with the missing parts marked as 'expected'.

The decision whether the CCR is modified at all depends on the comparison of the DoA with the *degree of agreement threshold* (DoAT). If DoA ≥ DoAT, construction carries out one of three operations: *generate*, *modify*, or *discard*, cf. Guhe & Habel (2001). Only *generate* is used for the purposes of this paper. It introduces new concepts into the CCR when (1) a new perceived entity comes in from the pre-processing unit or (2) DoA ≥ DoAT for the best match, and the best match is not yet contained in the CCR.

Selection

Selection, like linearisation and PVM-generation, operates on the *traverse buffer*[4]. Its variable length is determined by the parameter *length of traverse buffer* (LoTB). The traverse buffer contains pointers to situation refOs. The refOs can be modified, because no preverbal messages have been generated for these situations. Such a buffering is necessary for two main reasons; first, linearisation needs some time for its computations, and some selected situations have to be available simultaneously in order for linearisation to be able to perform its operations. Second, selection needs a possibility for revising its choices, especially with the principle underlying the functioning of all processes: *Extended Wundt's Principle*. Levelt (1989) formulates *Wundt's Principle*: an incremental process starts computing as soon as it obtains some characteristic input. Our extension consists in not only starting to process input as soon as pos-

[3] A perceived entity needs to be modified if—for example—the movement of an object, say the movement of a plane on a straight path, is made known to INC by the pre-processing unit—due to the incremental character of this pre-processing—before the movement is finished: if the object stops moving, this information has to be sent to INC, if it continues moving this has to be done, as well, especially in case the path is segmented due to a change of orientation.

[4] The notions 'traverse' and 'traverse buffer' of the present paper differ slightly from the ones in Guhe & Habel (2001). The former conceptions had some difficulties: if a refO was verbalised more than once, it was present in the traverse more than once, which meant that the sequence in which refOs were verbalised could be reconstructed only with great difficulty. We kept the terms but decided to introduce a new kind of refOs— called *verbalisation refOs*—that contain exactly these information. The traverse is now the list of verbalisation refOs—it is a part of the discourse memory—, and the traverse buffer is a list of pointers to refOs in the CCR, which are selected for verbalisation but not yet verbalised. Thus, refOs in the traverse buffer are not verbalisation refOs!

sible but to produce output as quickly as possible, as well. The disadvantage of this method is that quite a lot of choices need to be revised. Yet, the advantage is that linearisation has the maximum amount of time to operate on the elements in the traverse buffer, before they are taken out by PVM-generation. Additionally, this method results in an *anytime* capability of INC: PVM-generation is capable of producing a preverbal message at any point in time. An additional reason is that from an empty traverse buffer no verbal output can be produced, which—in the end—leads to audible gaps in the verbalisation.

Selection uses two operations for modifying the traverse buffer: *append* and *replace*. Although we consider it likely that a deletion operation is needed someday, INC has none, because up to now we never needed it.

Selection applies its algorithms to each new or modified refO. If it decides to insert an element into the traverse buffer, the time of insertion—a time stamp—is also saved with the element. Apart from the selection algorithm presented in Guhe & Habel (2001), which always selects the most complex concept available that had not been verbalised up to that point, we now use also another one: *retain the granularity level if possible*. Since selection only operates refOs with the sort *situation*, each refOs is tested whether it contains a description with an *after* predicate that connects it to elements in the traverse buffer, because this relation is only present between situations on the same level of granularity (this is ascertained by construction).

As we argued when discussing Extended Wundt's Principle, it is advantageous to always have some elements in the traverse buffer. Thus, if the traverse buffer is empty refOs of another granularity level are selected. The next extension of the algorithm will be that in this case granularity levels are preferred that were already used before and the hierarchical structure is evaluated in a more sophisticated way.

PVM-generation

PVM-generation has access to the *head of traverse buffer*, the first element of the traverse buffer. It takes out this element when the specified *latency* (LT), computed as the difference between the current point of time and the time stamp that was attached at insertion time, has passed. Starting from this refO PVM-generation commences the generation of an incremental preverbal message that describes this situation. However, before we illustrate how this is done in detail, we first introduce the representations used by INC.

Representing Motion Events

Referential Nets: Sorts and Descriptions

We represent conceptualisations of motion with *referential nets* (Habel, 1986, 1987). Referential nets were developed to model cognition-motivated linguistic processes, especially representations that change over time.[5] The referential net representations used in INC's CCR correspond to the conception of 'situation model' used by Lewis (1993) in an NL-Soar approach. In referential nets, all information about entities are associated with *referential objects* (refOs). Figure 3 depicts a referential net representing the moving plane of figure 2. RefOs are specified by their sort (as *object* or *path*) and by descriptions as *ηx chpos(r2, x, r3)* or *ηx plane(x)*.[6] The conceptual entities processed and stored in the CCR—including perceived entities—are of one of three sorts:

1. *object*: Objects are concrete real-world objects, like a plane, a car, or a person.
2. *spatial entity*: A spatial entity is an abstract entity representing *locations*, *directions*, or *paths of motion*. Note that the path of motion is not visually persistent, i.e. the paths we are talking about are no 'real-world' tracks. Paths are linear, directed, bounded entities, cf. Eschenbach, Tschander, Habel & Kulik (2000).[7]
3. *situation*: Situations cover other situation types—sorts in referential nets—as *state*, *event*, and *process*. States differ from the other two types in that they are static. Both, events and processes are dynamic but differ in containing a culmination point (events) or not (processes).[8]

Activation and Activation Threshold (AT)

The elements in the CCR all have an *activation* value, and the *activation threshold* (AT) is introduced as new parameter. Each description and each refO has an activation value that reflects its prominence in the conceptual representation (degree of attention of the speaker). The activation values are assigned to descriptions when they are introduced into the CCR. They are determined by the role the entities play in the perceptual preprocessing, i.e. this value is set by the PPU for perceived entities and by construction for elements that were computed with the help of the concept matcher. The function of activation values and the activation threshold will be described in the following section.

Producing Incremental Preverbal Messages

The INC version described in Guhe & Habel (2001) generated only increments on the level of complete

[5] This orientation to cognition distinguishes referential nets from kindred approaches of linguistic semantics, e.g. *discourse representation theory* (Kamp & Reyle, 1993).

[6] The η-operator—a logical counterpart to the indefinite article—is used in the referential net approach to link propositional representations with refOs (Habel, 1986; Guhe, Habel & Tschander, 2003). 'ηx chpos(r2, x, r3)' can be read as *'an object, that changes position in situation r2 along path r3'*.

[7] A path has two distinguished points: the starting point precedes every other point of the path; the final point is preceded by every other point of the path.

[8] For further details on the classification of situations see Bach (1986).

propositions, i.e. preverbal message representing situations. This was an adequate level of incrementality as long as we were mainly interested in the overall structure of sequences of event descriptions, i.e. while we were interested in which situations (events) were selected for verbalisations and which were not. Put differently, this was acceptable while we were mainly interested in the workings of the selection process. However, for a realistic language production system this will not do, because the information contained in one increment is no complete preverbal message.[9]

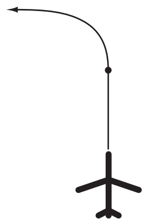

Figure 2: The motion event discussed as example

This becomes clear if we consider again the representation of motion events. Take for example the motion event depicted in figure 2—corresponding animations are presented in a current verbalisation study. There, a plane moves on a straight path until, starting at the location indicated by the dot, it follows a left curve. People describing the scene see neither the lines nor the dot; these items simply serve to illustrate the movement here. So, the plane moves on a white space. The conceptual analysis reveals that there are three kinds of entities. This means that an utterance like ‘A plane moves straight on.’ consists of—at least—three entities: one for the object, one for the path of motion and one for the overall situation. Figure 3 shows the representation in the CCR that is built up by construction for this scene.

INC uses an algorithm for the generation of incremental preverbal messages that bears some resemblance to the incremental algorithm proposed by Dale & Reiter (1995) and its extension of van Deemter (2002). However, these algorithms generate referring expressions, not complete preverbal messages.[10]

Deciding upon content to be verbalised is always a two-step process in INC. In the first step it decides to verbalise a refO. Usually, refOs contain lots of descriptions, some of which refer to other refOs. Hence, in the second step descriptions from the refO and further refOs (with further descriptions with further refOs and so on) are selected. The large number of descriptions makes constraints for the selection of descriptions indispensable, because not all information given by the descriptions of a refO are needed in a verbalisation.

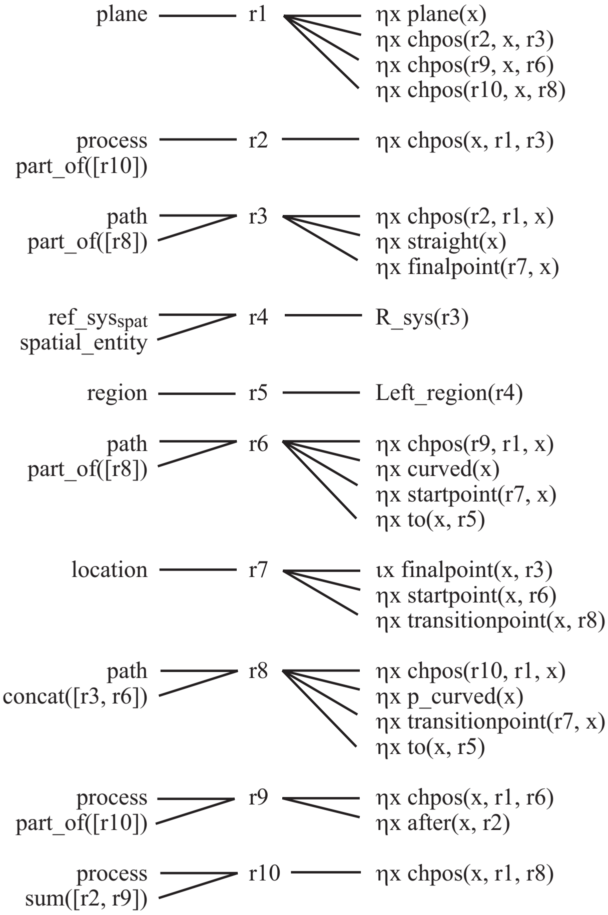

Figure 3: The referential net for the example

Descriptions are selected with regard to three kinds of constraints: structural, activation, and conceptual constraints. The *structural* constraint, which is the backbone of the algorithm, says that only *grounded descriptions* can be verbalised. If descriptions refer to no other refO they are *directly grounded*, e.g. *ηx plane(x)*. Descriptions pointing to other refOs are not grounded are evaluated whether they are *groundable*. In the chain of grounding all refOs contained by the description under evaluation are checked whether they have a grounded description. If a description is groundable by a chain of grounding the description is *indirectly grounded*. For example *ηx chpos(x, r1, r8)* of *r10* is such a description, because *r1* and *r8* contain grounded descriptions. If the grounding attempt does not succeed the description is *cyclic*, e.g. if description *ηx finalpoint(r7, x)* of *r3* is considered for verbalisation the description *ηx finalpoint(x, r3)* of *r7* is cyclic, because it contributes no new information: it is possible to reduce both to the same formula, *finalpoint(r7, r3)*. Note that at this point only the description from *r7* counts as cyclic. If *r7* has another grounded description it is still possible to ground the description of *r3*.

[9] For the domain of drawing sketch maps, though, we could regard this as sufficient, because most utterances we recorded in our studies were utterances like ‘A line is drawn.’ or ‘This will become a crossing.’ The information expressed by these utterances is more or less the information represented by the drawing event.

[10] We plan to integrate the Dale & Reiter / van Deemter algorithm into ours at the point at which referring expressions are generated. However, we have not done so up to now.

The *activation* constraint evaluates the activation of a description in question. It filters out descriptions with low activation values.[11] Whether or not the value is high enough for the description to be verbalised not only depends on the value of AT. If the value of the activation is above the value for AT the description is used in the preverbal message. On the other hand, if it is below AT it is not automatically ruled out. For obligatory elements, i.e., for elements that are necessary to build correct preverbal messages, INC can still decide to take the description despite the low activation value—or it has to generate *proforms*, as pronouns or similar constructions.

Up to now, we make use of only one *conceptual* constraint in our algorithm, the *homogeneous-part-of constraint*. If a refO is tested in the chain of grounding that is part of other refOs in the current preverbal message and the refOs in question are of a *homogeneous sort*, no description from the refO will be used in the verbalisation. For example, the sort *path* is homogeneous, because parts of paths are paths.[12]

Incremental Preverbal Messages Generated with the Described Algorithm

Since the three resource parameters DoAT, LoTB, and LT were the main topic of Guhe & Habel (2001) we now give an example where the newly introduced parameter AT causes different behaviour, i.e. where it causes differences in the generated incremental preverbal messages. We focus on the activation values assigned to descriptions and do not use activation of refOs in the following. The output of an incremental preverbal message generated by INC[13] is given in figure 4.

Since numbers of refOs are assigned automatically by the system, the numbers differ from the ones in figure 3. In this notation {…} stands for the activation value of a description. The attribute `at_time` represents the time interval of the situation, e.g. `at_time(0 200)` says that `r1` happened in the time frames 0 to 200; since each time frame spans 50 ms, `r1` lasted 10 s. The `status` attribute has one of three values: `expected`, `discarded`, or `regular`. This reflects the status of a refO in the sense whether it was actually perceived (`regular`), whether it is `expected` because construction and concept matcher generated it during the construction of a complex concept, or whether construction and concept matcher `discarded` an expected refO, because later input revealed that the expectation is wrong. The

```
DoAT=0.5; LoTB=6; LT=125; AT=0.6
Selection strategy: retain granularity
_____________________________________________
New PVM starting with r1

situation -- r1 --- eta x: chpos(x 0 2) {0.5}
part_of([6])
at_time(0 200)
process
status(regular)
p_entity(2)

object ----- r0 --- eta x: plane(x) {0.9}
plane
status(regular)
p_entity(1)

path ------- r2 --- eta x: straight(x) {0.6}
part_of([5])
status(regular)
p_entity(3)
_____________________________________________
New PVM starting with r3

situation -- r3 --- eta x: chpos(x 0 4) {0.5}
part_of([6])
at_time(201 320)
process
status(regular)
p_entity(4)

object ----- r0 ---
atom
plane
status(regular)
p_entity(1)

path ------- r4 --- eta x: curved(x) {0.6}
part_of([5])
status(regular)
p_entity(5)
```

Figure 4: Example output of INC (1)

`p_entity` (perceived entity) attribute contains a number that allows pre-processing unit and INC to exchange information about perceived entities. In particular, this allows the pre-processing unit to inform INC of changes of a perceived entity, e.g. that the plane is now involved in another situation when it follows the left curve.[14]

Two situation refOs, `r1` and `r3`, are chosen by selection. Since after these decisions are made, they *have* to be verbalised, their descriptions are selected although the activation is below AT. For the other refOs, there exist descriptions above AT. Thus, the incremental preverbal message results in an utterance like:

(1) A plane moves straight on. It moves on a curve.

A contrasting verbalisation arises when the ‘old’ selection strategy is used so that the first incremental preverbal message—describing the straight movement—is identical to the one shown above, cf. figure 5. Then, however, the situation refO representing the whole

[11] In the current version of INC, the activation values do not change after they are set. For a more adequate model of the production process, values changing dynamically due to decay and reactivation are proposed. Yet, we do not need this for the simple example at hand. Decay rate and reactivation rate are two parameters to be introduced in the next version of INC.

[12] Note that in our approach paths are *extended*, and points *lying on a path* are not formalised as *parts* of a path but as *coinciding* with a path.

[13] The current version of INC with which these simulations were carried out is 0.2.1.

[14] Since the representation and processing of spatial reference systems is outside the scope of the present paper, concepts as `left_region` and linguistic expressions as *left* do not appear in the current examples. (But see, Eschenbach et al, 2000; Guhe et al., 2003.)

```
DoAT=0.5; LoTB=6; LT=125; AT=0.2
Selection strategy: most complex

[... first PVM same as above ...]

_____________________________________________
New PVM starting with r5

Situation -- r5 --- eta x: chpos(x 1 6) {0.5}
at_time(0 321)
sum([1 3])
process
status(regular)

object ----- r1 ---
plane
status(regular)
p_entity(1)

path ------- r6 --- eta x: trans_pt(7 x)
{0.5}
                    eta x: p_curved(x) {0.6}
sum([2 4])
status(regular)
p_entity(7)

location ---- r7 ---
status(regular)
p_entity(8)
```

Figure 5: Example output of INC (2)

movement (`r5`) is selected. More important, though, AT is only 0.2 this time. This output is generated if AT < 0.6. The important difference here is not that the more complex refO is selected—this is not surprising—but that the description *ηx transitionpoint(r7, x)* is also generated. This has an impact on the lexemes that are selected by the lexical access component: with this additional description a lexeme like *turn off* is chosen:

(2) A plane moves straight on. It turns off.

Conclusion and Future Work

With INC we propose a cognitively motivated model of the first stage of the language production process, the conceptualiser. In addition to three resource parameters (DOAT, LOTB, LT) described in Guhe & Habel (2001) we discussed the new parameter *activation threshold* (AT), which operates on activation values assigned to refOs and descriptions, i.e. the internal counterparts to entities perceived or conceived. Furthermore, we generated verbalisations for a new domain, motion events, with a new selection algorithm.

For the next version of INC two enhancements are planned. First, an activation value that is set to its initial value at the time of insertion into the CCR will be treated as a dynamic entity, influenced by additional parameters. In this version of INC, activation values will—on the one hand—decay according to the *decay rate* (DR), on the other hand, each time a refO or description is visited by a process it is reactivated according to the *reactivation rate* (RR). Second, the selection algorithm will be made 'more intelligent': if the traverse buffer is full, selection switches to the next higher level of granularity and replaces simpler elements, and if the traverse buffer is empty the next finer level of granularity is used.

Acknowledgements

The research reported here was conducted in the project ConcEv (Conceptualising Events), which is supported by the DFG in the priority program 'Language Production' under grant Ha-1237/10 to Christopher Habel.

References

Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard UP.

Avrahami, J. & Kareev, Y. (1994) The emergence of events. *Cognition*, 53, 239–261.

Bach, E. (1986) The Algebra of Events. *Linguistics and Philosophy, 9*, 5–16.

Dale, R. & Reiter, E. (1995) Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 18, 233–263.

Eschenbach, C., Tschander, L., Habel, Ch. & Kulik, L. (2000) Lexical Specification of Path. In Ch. Freksa, W. Brauer, Ch. Habel & K.F. Wender (Eds.) *Spatial Cognition II.* Berlin: Springer.

Guhe, M. (in prep.) *Incremental preverbal messages.*

Guhe, M. & Habel, Ch. (2001) The Influence of Resource Parameters on Incremental Conceptualization. Proceedings of the 2001 Fourth International Conference on Cognitive Modelling (pp. 103–108). Mahwah, NJ: Lawrence Erlbaum.

Guhe, M., Habel, Ch. & Tschander, L. (2003) *Describing Motion Events: Incremental Representations for Incremental Processing*. Proceedings 5th International Conference on Computational Semantics (IWCS-5), Tilburg, Netherlands.

Habel, Ch. (1986) *Prinzipien der Referentialität*. Berlin: Springer.

Habel, Ch. (1987). Cognitive linguistics: The processing of spatial concepts. *T. A. Informations (Bulletin semestriel de l'ATALA, Association pour le traitement automatique du langage)* 28. 21–56.

Kamp, H. & Reyle, U. (1993). *From Discourse to Logic*. Dordrecht: Kluwer.

Levelt, W.J.M. (1989) *Speaking: From Intention to Articulations*. Cambridge, MA: MIT Press.

Lewis, R. L. (1993). *An Architecturally-based theory of Human Sentence Comprehension*. PhD thesis, Carnegie Mellon University.

Lewis, R.L. (1999). Specifying architectures of language processing: Process, control, and memory in parsing and interpretation. In M. Crocker, M. Pickering & C. Clifton, eds., *Architectures and mechanisms for language processing.* Cambridge: CUP.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard UP.

van Deemter, K. (2002) Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm. *Computational Linguistics*, 28, 37–52.