INC – An incremental conceptualiser for language production

Markus Guhe

Cognitive Science Department, Rensselaer Polytechnic Institute
110 8th Street Troy, NY, 12180

Abstract

The first stage of language production in Levelt’s (1989) model is conceptualisation. Due to the complex, open-ended nature of conceptualisation developing computational models of conceptualisation proves difficult. I present a first cognitive computational model of the conceptualiser within Levelt’s framework. The incremental conceptualiser (INC) places particular emphasis on the incremental mode of operation, which means that the same information stream is processed on multiple stages at the same time. It reads perceived input about events it observes in the environment and generates preverbal messages (semantic structures) that describe the observed events and can be encoded linguistically by a subsequent formulator.

Key words: conceptualisation, language production, incrementality, cognitive modelling

1 Thinking while speaking

The observation that humans are able to think while they speak is not new. Heinrich von Kleist, for example, observes in his famous essay Über die allmähliche Verfertigung der Gedanken beim Reden that ‘many a great orator, in the moment that he opened his mouth, did not know what he would say’. (von Kleist 1805)

Email address: guhem@rpi.edu (Markus Guhe).
URL: http://www.cogsci.rpi.edu/guhem/ (Markus Guhe).

1 The research reported in this paper has been supported by the Deutsche Forschungsgemeinschaft in the project Conceptualising Events (ConcEv, grant Ha-1237-10). It was carried out while I was working at the Department of Informatics (group WSV), University of Hamburg. I want to thank Koenraad De Smedt, Christopher Habel, Holden Härtl, Kees van Deemter, and the people at the CogWorks lab for comments on earlier versions and Wayne Gray for giving me the time to write this paper.
Nowadays one would cast this in different terms: utterance production commences as soon as the communicative intention is conceived, although the utterance plan is not complete. After a speaker has started an utterance on an incomplete utterance plan, planning and speaking go on interdependently like ‘two wheels on one axis’ (von Kleist, 1805). In other words, thinking and speaking are tightly connected, and they are also temporally closely linked. Accordingly, speaking cannot be seen isolated from thinking. In particular, investigations aiming at cognitive plausibility must make this intermediary ‘thinking’ step in order to make the connection from the world to language.

Although von Kleist’s observations are not new (he was neither the first nor the only one to make such observations, even if his essay is among the most poignant and well-known ones), there is no attempt to capture them in a cognitively adequate computational model. Cognitively adequate means that the model not only produces adequate verbal output, but that it also models the way in which humans perform the task. With the incremental conceptualiser (INC, pronounce: [ɪŋk]) I present the first model of this kind. INC is a model that covers the whole stretch between perception and semantic structures. This means that its major contribution is to integrate different aspects of conceptualisation. (This also means that most problems cannot be accounted for in all their complexity.)

The general problem of the influence of language on thinking is the problem of linguistic relativity, cf. Gumperz and Levinson (1996) for an overview. Since this is not my major concern, it suffices to state that thinking and speaking are two different stages of the language production process. Nevertheless, humans think (conceptualise) in a particular way when they speak, for which Slobin (1996) coined the phrase thinking for speaking. Emphasising the temporal interleaving of both INC is a model of thinking while speaking. Thinking of this kind is called conceptualisation. The temporal interleaving of thinking and speaking put forward so eloquently by von Kleist is modelled in INC as generating a communicative (sub-)intention in a first step and then deciding on how this communicative intention is realised. The communicative intention is present throughout the generation of the utterance, but it is realised in a piecemeal fashion – increment by increment.

Levelt’s (1989) psycholinguistic framework for language production is my starting point for building INC. However, there is not only no computational model of a conceptualiser, there is hardly any research about conceptualisation as such. Psycholinguistics, for example, is mostly concerned with questions that are located further down in the language production system, eg lexical access. A major reason for lack of research on conceptualisation are the difficulties of its investigation. Firstly, it is never directly observable but only via a surface modality like language or motor movements (Nuyts, 2001). Secondly, conceptualisation is an open-ended task, more precisely I should say that INC generates the equivalent of a mental representation of a sub-intention. INC itself is very unlikely to ‘have’ intentions.
which means that it has no clear-cut boundaries. A consequence of this situation is that semantic structures are often regarded as given. Yet, semantic structures do not appear from nowhere, they must be generated. Hence, conceptualisation is vital for language production. INC is a step into the largely unexplored area of conceptualisation for language production. Vice versa, its cognitive processing principles can inspire other disciplines as well. (It may be similar to bionics: observations of principles occurring in nature are transferred to non-natural systems.)

2 Levelt’s tripartite architecture

Willem Levelt proposes the most prominent model of language production that is based on cognitive and psycholinguistic considerations and that furthermore goes all the way from communicative intentions to articulatory output (Levelt, 1989, 1999; Levelt et al., 1999). INC is in particular a model that generates online descriptions of events (Habel and Tappe, 1999). That means, it generates semantic descriptions of events it observes happening in the world while they take place. One aspect of this task is that the visual input must be translated into verbal output. The underlying assumption is that mental representations are multimodal (Levelt, 1989, p 73). Other types of knowledge are most notably spatial, imagery, and kinaesthetic knowledge. On the semantic level, however, all representations must be propositional, because all linguistic representations are propositional. Thus, conceptualisation must convert knowledge that is to be verbalised into propositions. This includes in particular perceptual information and knowledge stored in long term memory, and the interaction of perceptual information and long term knowledge lies at the heart of INC’s functionality. In this fashion, conceptualisation is the mediator between the world and language. However, in order to keep the complexity of conceptualisation manageable INC uses propositional representations as input. There are three main reasons that this idealisation is valid. Firstly, not only linguistic knowledge is propositional, but it is the most important modality for language, because linguistic knowledge is propositional. Secondly, the representational formalism INC uses (referential nets) is capable of representing knowledge in other modalities as well, cf section 6.1. Therefore, adding modalities will only extend INC. Thirdly and most importantly, the input representations are non-linguistic, and INC generates semantic representations from them. This is the main function of the conceptualiser. The input is provided by a pre-processing unit (PPU) that takes the spatio-visual information and segments it into simple concepts, called perceived entities.

The three main components of Levelt’s architecture are conceptualiser, formulator,
and articulator, cf figure 1. The two interfaces between these components are the preverbal message between conceptualiser and formulator and the phonetic plan between formulator and articulator. The strict sequentiality of the architecture allows no feedback between components but only within (Levelt 1989 p 15f). Consequently, a component does not know how far its output is already processed by subsequent components. This is relevant, for example, in cases where output is erroneous or incomplete, because there is no possibility for a subsequent component to send back information about this to a preceding one. The Levelt model, therefore, contains a monitoring component that reads in the parsed speech of the language comprehension system. This is one of two sub-components of the conceptualiser. It enables the conceptualiser to keep track of what parts of the planned utterance(s) are already produced, and it can detect deviations. The second sub-components is message generation, which generates preverbal messages. INC is a model of the message generation component. Message generation performs four main tasks (Levelt’s terms are given in parentheses):

1. building up of an internal representation (bookkeeping),
2. selection of the content to be verbalised (macroplanning 1),
3. bringing the selected content into a linear order (macroplanning 2),
4. generation of preverbal messages (microplanning).

Each of these four tasks is performed by one process in INC. All processes work incrementally, i.e. as soon as a new input increment is available it is processed.

Preverbal messages are special conceptual representations. Mainly two properties set them apart from other conceptual representations. Firstly, they are sequential representations, while conceptual representations are hierarchical. Thus, preverbal messages contain neither part-of nor subsumption hierarchies, see below. Secondly, preverbal messages are those conceptual representations that contain meaning to be conveyed (Jackendoff 1987 1990 1997 2002 Wiese 2003). They hold the infor-
information that is necessary for the formulator to encode the intended meaning linguistically, ie to generate a phonetic plan. This means in particular that the components other than the conceptualiser have no access to the conceptualiser’s internal representations as this would be just another form of feedback. In this sense preverbal messages are pre-linguistic representation or semantic representations. However, this view is not shared by all semantic theories, eg two-level semantics (Bierwisch and Schreuder 1992).

The formulator takes preverbal messages and encodes them linguistically, resulting in the phonetic plan. Models of the formulator include IPG (Kempen and Hoenkamp 1987), IPF (De Smedt 1990a), SYNPHONICS (Günther et al. 1995), WEAVER (Roelofs 1997), and Performance Grammar (Kempen and Harbusch 2002). The phonetic plan is a program for articulation, and is also input to the language comprehension component. The articulator computes motor commands from the phonetic plan and executes them. The result is the production of overt speech – the output of the articulator and input to the language comprehension component. The output of this component is parsed speech, which is the input to monitoring and closes the loop.

3 Incremental conceptualisation: an example

Imagine you are sitting at an airport. You are looking out of the window while waiting to board, and watch what is happening on the manoeuvring area. Imagine further that you describe what you see to another person, who cannot see what you are seeing, say, a person you are talking to on the telephone. How do you accomplish this? Assume you observe the scene depicted in figure 2.

In this scene a plane docks onto a gate. Four phases can be identified and described verbally as:

(1) a. *Ein Flugzeug fährt auf ein Gate zu.*
   ‘A plane is moving towards a gate.’

4 There are two caveats. When looking out of the window, you will have a different perspective, ie you will not view the manoeuvring area from a bird’s-eye perspective. Also, planes move at a rather slow speed when docking, so the need incremental processing may not be that urgent. The participants performing the verbalisation studies against which INC was tested watched scenes like the one given in figure 2 on a computer screen from a bird’s-eye perspective, and the planes moved at a speed that induced sufficient verbalisation pressure for incremental processing to be crucial for coping with the task. The phrase *looking out of the window* should, therefore, be understood as an overall motivation.

5 The examples are given in German, because the recorded verbalisation data used for evaluating INC are German, cf. Guhe et al. (2003b); Guhe (2003a).

6 There is no English word that directly corresponds to the German *fahren*. Drive comes closest but has a different usage in English. For this reason, *fahren* always is translated as move, although move is more general (German: bewegen).
b. *Es stoppt beim Gate.*
   ‘It stops at the gate.’

c. *Der Laufgang bewegt sich auf das Flugzeug zu.*
   ‘The walkway is moving towards the plane.’

d. *Er erreicht das Flugzeug.*
   ‘It reaches the plane.’

I am concerned with the cognitive task how humans can get from visuo–spatial input to verbal output while the described events take place (not afterwards). This is a case of *incremental conceptualisation*. Conceptualisation is the task of producing pre-linguistic, semantic representations out of non-linguistic input, cf section 2. Incremental means that conceptualisation and its sub-tasks are performed in a *piecemeal* and *parallel* fashion. *Piecemeal* characterises the fact that you need not see the whole scene before you can start describing it. Instead, you can start talking, ie describe a part of the scene, before it ends. It also means that you can start talking before planning of an utterance is complete. *Parallel* means that you perform multiple tasks at the same time, eg processing the visual input, linking this input with your knowledge of such scenes, and producing a verbal description for it.

The four main tasks of conceptualisation listed on page 3 are realised in INC as follows. *Construction* builds up an internal representation of the external state of affairs from the perceived input by using knowledge about how the world is structured. *Selection* decides upon the content to be communicated to an (only implicitly assumed) hearer with respect to the current verbalisation goal. INC has a fixed verbalisation goal: *describe the scene you observe!* *Linearisation* brings the selected content into an appropriate order. For instance, linearisation decides whether two phases of the scene are described in the temporal order in which they occurred or whether the order is inverted. Finally, the *generation of preverbal messages* (PVM-generation) generates a preverbal message for the content to be verbalised. INC also has two auxiliary tasks. The first one is the perceptual pre-processing performed in the *pre-processing unit* (PPU). It takes perceptual input data and forms simple concepts from it that are the input to the conceptualiser. The second one is the *concept matcher*. It is an auxiliary task of construction for accessing long term knowledge, ie the knowledge that is present independently of the conceptualisation of an ob-
served scene. In particular, this task serves to access knowledge of how to build complex concepts from simpler ones.

The enormous complexity of conceptualisation can be reduced by focussing on the data-driven aspects of the task. Data-drivenness emphasises the need for incremental processing: as soon as a new piece of information (increment) is available it is processed with respect to the current conceptual representation. Processing the scene in figure 2 in an incremental fashion means, therefore, to use an approach in which not the whole scene is observed before conceptual processing commences, eg before boundaries between the phases are identified, but input is processed as it becomes available. This is called an online setting: language production starts before the scene has been observed completely. In contrast, in an offline setting all information required for a verbal description of the scene is available beforehand, ie the scene ended before verbalisation starts. The online setting makes it possible to correlate what is happening to its verbal description. In this way conceptualisation can be investigated although no direct observation is possible, cf section 1.

Verbalisation (1) of the scene illustrates two important points. Firstly, a human observer is capable of segmenting the input stream into sub-scenes (phases) – a prerequisite for piecemeal (incremental) processing. Secondly, the sub-scenes are part of the overall scene, ie the representation is hierarchically structured, cf figure 3. See Habel and Tappe (1999) on this issue. The latter is crucial for the human observer to be able to recognise the succession of the four phases as a DOCKING scene. The second, equally important hierarchy is the subsumption hierarchy, cf figure 4. While the part-of hierarchy establishes relations between an entity and its parts, eg a COCKPIT is part of a PLANE, the subsumption hierarchy relates kinds of entities. Therefore, it is also called is-a hierarchy, eg a PLANE is a kind of AIRCRAFT.

Both hierarchies are used to establish relations between concepts, the building blocks of conceptual representations. A concept is a system-internal representation of a concrete or abstract entity or of a set of entities[8]. What concepts are involved

---

7 When I refer to concepts in an informal sense I will use intuitively plausible names, like DOCKING or PLANE. These labels are not part of the formal representations.
8 This use of concept differs slightly from the standard use in psychology, where concepts are usually understood as mental representations of categories, ie as a mental representations of sets of entities [Murphy] 2002.
in the example scene? The answer this question requires a refinement. So far, scenes and phases were unanalysed wholes. However, it is advantageous to represent them in interrelated sub-structures. First of all, a distinction of object and situation structure must be made. This is a subdivision within the subsumption hierarchy. Part of the subsumption hierarchy representing the knowledge about how objects are hierarchically organised is given in figure 4.2. In the representations in figure 3 two types of situations are used, one where an object changes its position (MOVE) and one for its transition of motion to standstill (STOP). (The transition from standstill to motion is captured by a START concept, which is left out here; but see figure 9.)

Three object concepts are required for representing the example scene: PLANE, GATE, and WALKWAY. They are related by spatial entities, here are PATH (the trajectory an object is following) and LOCATION (the position of an object). Each phase of the scene is represented by a situation. Situation is a term comprising notions like event, process, and state (Davidson 1967; Bach 1986). However, distinguishing situation types on the conceptual level proves difficult. Consider the first phase of the scene. As the PLANE is moving steadily, the situation is of the type process. Yet, firstly, the observer is capable of inferring that the movement has a starting point and an endpoint, which he did not (yet) observe but which would classify the situation as an event. Secondly, after phase 2 the movement is an event, because it has an endpoint. Nevertheless, if the movement itself is described after phase 2 (The plane was moving towards the gate.) it is a process again. Thus, when describing the movement different aspects of the situation can be highlighted, depending on whether the starting or endpoint is taken into account. Instead of representing both, the simpler and more flexible solution is to use an integrated representation on the conceptual level that captures what is common to all views on the situation. The situation type is only determined when it is needed. SITUATION concepts are, therefore, characterised by further attributes, most notably a distinction of instantaneous and extended situations and the fact of whether the situation is already completed. I will use event and situation synonymously most of the time, because most of the situations I investigate would indeed be classified as events.
Considering the situation structure of the docking scene depicted in figure 2 in more detail now, one can see that in phases 1 and 3 an object changes its position in the world, in the other two a moving object ceases to move. Using the concepts introduced above, the underlying conceptual representation of the verbalisations (1a–d) can be expressed in propositions like:

\[ s_1: \text{MOVE}(\text{PLANE}, \text{TOWARDS}(\text{GATE})) \]
\[ s_2: \text{STOP}(\text{PLANE}, \text{AT}(\text{GATE})) \]
\[ s_3: \text{MOVE}(\text{WALKWAY}, \text{TOWARDS}(\text{PLANE})) \]
\[ s_4: \text{STOP}(\text{WALKWAY}, \text{AT}(\text{PLANE})) \]

The terms $s_1$–$s_4$ are assigned to these predicates that stand for the situations representing the four phases, which makes it possible, for example, to refer back to the situation later on. This is a central point of the conception of referential nets, cf section 6.1. The first argument is an object, which is the bearer of motion, ie the moving object; the second argument is a proposition that describes the corresponding path in case of a MOVE and the location in case of a STOP predicate.

A disadvantage of representations given this way is that they do not explicitly state that PLANE, GATE, and WALKWAY are always referring to the same entity. One way to mark this is to give them subscripts, eg \text{PLANE}_1. Furthermore, not for each phase of a scene a new representation should be created for which the relations to the previous one are then explicitly computed. So, the formalism must provide means to represent changes, eg the fact that the PLANE is at a different locations. Having such dynamic representations is a vital requirement for incremental processing.

On such a dynamic, conceptual representation of the scene the four tasks introduced above execute the following operations. Construction builds up the hierarchical structure and determines that the sequence of the four phases constitutes a DOCKING situation. Selection chooses which of the situations are described verbally, eg it may decide to verbalise phases 1 and 2 and the overall DOCKING situation. Linearisation can, for example, invert the verbalisation of phases 1 and 2, which can result in an utterance like \textit{The plane stopped after it moved towards the gate}. PVM-generation, finally, generates a preverbal message for each selected situation. The preverbal messages look similar to the above equations.

Observe that the first two motion events are performed by the \text{PLANE}, the last two by the \text{WALKWAY}. This allows to group the movements with respect to the bearer of motion. The result are more complex and abstract motion concepts, simply called \text{MOVE} in figure 5. The intermediary level in these representations proves useful in conceptual representations (Guhe 2003a).
Figure 5. Example scene: more elaborated representation of the hierarchical event structure

Figure 6. The architecture of INC

4 INC – an overview

The overall architecture of INC is shown in figure 6. It consists of four main processes that form the cascade of this incremental model, one auxiliary process (the concept matcher), and two storages (memories) for representations. The four main processes that make up the cascade are construction, selection, linearisation, and PVM-generation (preverbal message generation). In short, construction reads the output of the perceptual pre-processing unit (PPU), called perceived entities, and builds up the current conceptual representation (CCR) from them with the help of the concept matcher. Each time construction modifies the CCR it informs selection of the change, which decides whether the changed element will be verbalised. If selection changed the plans for verbalisation it tells linearisation, which checks whether the sequence of the events selected for verbalisation needs to be reordered. PVM-generation, finally, takes a selected event and generates an incremental preverbal messages.
bal message for it. Observe that in the non-cognitive model by André et al. (1988) the overall subdivision of tasks is the same. The model by Siskind (2001) is also similar, but it focusses on visual perception and grounding of lexical semantics for event verbs. However, it does not account for the complexity of conceptualisation, eg processes like selection or linearisation are not considered.

In this way the CCR is traversed, and a data structure called the traverse is generated. The traverse contains all the elements of the CCR that are verbalised, ie sent to the formulator. Three processes, selection, linearisation, and PVM-generation, coordinate themselves via the traverse buffer, which contains pointers to event concepts in the CCR, cf figure 7 for a schematic depiction. There are two major reasons for using this buffer. Firstly, the linearisation process needs more than one event in order to perform its functions at all. Secondly, due to Extended Wundt’s Principle, which basically states that INC uses an eager model of processing, cf page 13, the selection process produces output rapidly. During the latency that the elements stay in the traverse buffer it can change its decisions. These CCR events are the events that are selected for verbalisation but which are not yet verbalised. Selection appends pointers to event concepts it decides to verbalise. As long as these are available in the traverse buffer, selection can replace elements of this buffer with other event concepts. Linearisation can reorder the elements in the traverse buffer according to principles of discourse structuring. However, this process has not been developed yet, because it is the one process conceptualisation can work without, in particular in the domain of motion events, where the events are strictly sequential. (See Guhe et al. (2004) for an extension for concurrent events.) After the latency expired PVM-generation takes (and removes) elements from the traverse buffer and generates incremental preverbal messages that describe the corresponding events, in particular the head of traverse buffer, ie the element in the headmost position. (For the purposes of this article I will assume that PVM-generation only takes the head of traverse buffer, but see Guhe et al. (2004) for a version where it takes more than one element.) INC is a parametrised model. Two of the parameters are the latency (parameter LT) and the length of the traverse buffer (parameter LOTB). See Guhe and Habel (2001) and Guhe et al. (2003b) for a description of how different parameter values change the behaviour of INC and make it possible to match it onto empirically collected verbalisation data.

PVM-generation generates the preverbal messages incrementally. Thus, the traditio-
nal view on preverbal messages as static semantic representations is an inadmissible simplification in the context of incremental processing. Calling them incremental preverbal messages emphasises that they are sequences of increments (Guhe et al., 2000; Guhe, 2003b). This has two eminent consequences. Firstly, for each increment of an incremental preverbal message it must be ascertained that it fits to the increments already generated. This is the consistency condition. Secondly, all increments of an incremental preverbal message taken together must form a complete semantic representation, e.g. like the ones given in Jackendoff (1990). This is the completeness condition. In INC this is not done by explicit checks, but these conditions must be fulfilled by the way the algorithms generating the preverbal messages works. This is one effect of Extended Wundt’s Principle.

According to Levelt (1989), each utterance is initiated by a communicative intention. As intentions are one of the most entangled topics in philosophy they cannot be a concern of this article. For the purpose at hand, however, it is useful to think of it the following way. The overall, fixed communicative intention for INC is describe what you see! The situations that are selected for verbalisations are sub-intentions that – in the sense of sub-goaling – serve to fulfil the overall goal of describing the observed.

INC’s two memories are the CCR (current conceptual representation) and the CS (concept storage). The CCR is the internal conceptual representation of the observed external state of affairs. Since this representation changes over time as the events unfold, it is called current. An abstract representation of the state of the CCR at the end of the example scene is given in figure 5; the actual representation is given in figure 9 and will be discussed in subsequent sections. The CS contains the knowledge about how to construct more complex concepts from simpler ones. For example, it contains a rule on how to construct a complex MOVE concept from a simpler MOVE and a STOP concept.

INC does not possess a detailed memory model, e.g. issues like learning and forgetting are not considered. Furthermore, INC makes no claims about access and storing of information, e.g. the memories are linearly searched without accounting for an organisation of memory, say, with regard to the domain. INC’s memory is subdivided along the well-known distinction between working memory (CCR) and long term memory (CS, Baddeley, 1986; Eysenck and Keane, 1995), which is also used in unified cognitive architectures (Newell, 1990). Put differently, the CS contains encyclopedic knowledge, the CCR situation and discourse knowledge. From a more technical perspective, the distinction in INC is the one between previously available knowledge that is used for constructing a conceptual representation (CS) and knowledge constructed during conceptualisation (CCR). Note that this memory structure is more differentiated than Levelt’s.
5 Incrementality

Piecemeal processing captures the most important aspect of incrementality, ie an information stream is processed increment by increment (piece by piece). This means, not all information that is required for the correct and complete computation of an output is available when an input increment is read in and processed. Using processes to structure a model makes it possible to give another characterisation of incremental processing: incrementality is the parallel processing of a sequential information stream. This means three things. Firstly, there are multiple processes running in parallel, eg the processes in the conceptualiser. Secondly, the processes are arranged in a fixed sequence so that the output of one process is the input to its successor, eg the output of the construction process is sent to the selection process in INC. Thirdly, this characterisation conforms to Levelt’s no-feedback condition. Although in his model component-internal feedback is allowed, INC operates without direct feedback. However, it uses indirect feedback, feedback that is not realised as direct transmission of information. Instead, the component giving feedback alters a representation that the component receiving feedback is using as well. If the modification of the representation by the one component affects the operations of the other component an indirect feedback has been given. In INC these representations are the CCR and the traverse buffer.

Apart from the restriction to indirect feedback, INC is an incremental model that is hungry or eager. I term the underlying processing principle Extended Wundt’s Principle. It is based on Wundt’s Principle (after Wilhelm Wundt, who first proposed such a mode of processing) as it is introduced by Levelt (1989, p 26): ‘Each processing component will be triggered into activity by a minimal amount of its characteristic input.’ There are two important points about this principle. Firstly, the characteristic input must be defined individually for each processing component as this is the input the component can process. Secondly, it must be identified what constitutes a minimal amount for a processing component. It can be said, however, that a minimal amount is an increment, which must also be defined for each component individually. Note that saying the minimal amounts of input are increments does not mean that all increments constitute a minimal amount: increments can be of different sizes and can contain different kinds of information.

The stronger, extended version of Wundt’s Principle says that the components not only start processing as soon as possible but also produce output as quickly as possible: Each processing component will be triggered into activity by a minimal amount of its characteristic input and produces characteristic output as soon as a minimal amount of output is available. Using Extended Wundt’s Principle means that the processes do not evaluate their output, ie they make no decision whether the overall output would benefit if the process waited before giving the output to the subsequent process until, say, the quality of the output can be judged more reliably. Support for this view comes, for example, from Bock et al. (2003) and De Smedt.
who cites work by Hoenkamp stating that ‘what can be uttered must be uttered immediately’. It is also nourished by results of studies on language processing, eg Altman and Kamide (1999) and Crocker and Brants (2000) – although it is not clear whether language production and comprehension work according to the same principles. Extended Wundt’s Principle stands in opposition to a reasoning about the ‘when-to-say’ proposed by Kilger and Finkler (1995), which says to generate output as soon as necessary, not as soon as possible so as to optimise quality and reliability of the output.

6 Representations

6.1 Referential nets

INC uses referential nets (refNets) to represent knowledge. RefNets consist of interrelated referential objects (refOs) representing entities (Habel, 1982, 1986; Eschenbach, 1988). There are several reasons for choosing refNets. Firstly, since refOs are the major means of structuring knowledge, it could be called an object-oriented way to represent predicate logic. Consequently, the representations focus on the knowledge about individual concepts. The conception of refNets was developed in order to reflect the structure and organisation of memory and the way it supports the processes of storing and remembering (Habel, 1986, p 111f). Secondly, refNets are especially suited to connect the different levels of representation used here. That is, it is possible to represent semantic and conceptual knowledge as well as to connect a refO to the perceptual information provided by the PPU. Additionally, refNets can integrate multimodal knowledge, eg pictorial (Habel, 1987) or gestural knowledge.

The following shows part of a refNet, which might be used in the conceptual representation of the introductory example:

```
plane ——— r1
     'CK-314'
     owner('LUFTHansa')
     ηx plane(x)
```

In this graphical notation, r1 is the refO term. The lines leading towards r1 connect it to the expressions on either side. The ones to the left are attributes, the ones to the right designations.

Expressions are sorted (Habel, 1986, p 66): SORT is a set of names of sorts. Each expression has a sort-frame, which defines the sorts of an expression’s arguments. In order to be sort-correct the arguments must be of these sorts, eg the expression ηx plane(x) must have an argument of sort plane. (In the above example this is given, because x is substituted by r1, which has the sort plane.)
Apart from refOs especially two kinds of terms are important: names and descriptions. Examples of names are ‘DAVID’, the proper name of a person, or ‘CK-314’, the flight number of the plane in the above example. Descriptions are either functional expressions, or they are constructed with a description operator. An example of the former is owner(‘LUFTHANSA’), which states that the represented entity is owned by Lufthansa. Two points are important here. Firstly, the owner relation expressed by the functional expression can refer to another refO, eg owner(r2). Then, the owner of the entity is represented by r2, which has the advantage that additional knowledge about the owner can be represented. Secondly, since functional expressions are functions, they must uniquely refer to an entity, ie they must constitute a many-to-one or a one-to-one relation. In the case of ownership one can assume an entity only has one owner. (If there is a group of owners this can be represented by a refO standing for the group. The functional expression then refers to the group refO.) What is to be represented as functional expressions must be determined for each representation individually. Descriptions constructed with a description operator are of the form: op var formula with the operators op ∈ {ι, η, all_t, some_t}, the variables var ∈ {x, y, z, . . .}, and a formula of predicate-logic. The operators reflect the cardinality of the refO and the definiteness of the designation (Habel, 1986, p 137), cf the following table. For example, ηx plane(x) stands for a plane and all_t x plane(x) for all planes.

<table>
<thead>
<tr>
<th>cardinality</th>
<th>definite</th>
<th>indefinite</th>
</tr>
</thead>
<tbody>
<tr>
<td>cardinality = 1</td>
<td>ι</td>
<td>η</td>
</tr>
<tr>
<td>cardinality &gt; 1</td>
<td>some_t</td>
<td>all_t</td>
</tr>
</tbody>
</table>

Names and descriptions form the two sets NAM and DESCR respectively. The latter contains the subset of closed descriptions DESCR.cl. As usual, closed means that the terms contain no free variables. The union of NAM and DESCR.cl constitutes the set of designations (DESIGN = NAM ∪ DESCR.cl), which are the expressions written to the right of the refO term.

While designations are mainly used to represent semantic knowledge, attributes mainly represent inferential knowledge. Examples are the information about the sort of a refO, which is obligatory and always stands in first position (plane in the above example). If the sort of an entity is unknown its sort is top, cf figure 8. Temporal relations between situation refOs are also represented by attributes, eg before(r3) as attribute of a refO r2 represents the relation r2 before r3, cf section 6.2. Thus, attributes can have values. Values can be lists, written as [...]. An example of an important list attribute is parts. Assume that r5 represents Oscar Peterson, r6 Ray Brown, and r7 Herb Ellis. Then the refO representing The Oscar Peterson Trio, say r8, has the attribute parts([r5, r6, r7]).

Note that these operators construct terms while the quantifiers like ∃ and ∀ construct formulas, ie they are functions with range {true, false}. 

Draft – Do not cite!
Note that the sort plane and the description $\eta x\ \text{plane}(x)$ encode quite different properties of $r_1$. The former stands for an essential property of the refO, from which, for example, it can be inferred that the entity is capable of flying, the latter says that it can be described as a plane ([Habel] 1986, p 156). Calling them, say, abcd and $\eta x\ efgh(x)$ would mutatis mutandis not change the representation. The difference between attributes and designations is similar to the one between semantic and episodic memory ([Tulving] 1999). This is the reason that mainly designations are used for generating preverbal messages. Properties of the kind used by [Gärdenfors] (2000), which are properties tightly connected to perception, are represented mainly by attributes, properties connected to linguistic knowledge mainly by designations.

Although refNets are mainly used for representing propositional knowledge, they can also represent knowledge in other modalities. This allows a translation into propositions. [Habel] (1987, 1988) provides an extension by depictions, which can represent spatial knowledge. Similar extensions for other modalities are possible. The perceived entities from which the conceptual representation is constructed are identified by non-linguistic properties. Such properties as well as knowledge retrieved from long term memory (INC’s concept storage) need not be propositional. Yet, when a preverbal message is generated, only the propositional parts of a refO are used, ie mainly names and descriptions.

Coming back to the issue of why refNets are well-suited for representations in INC additional advantages become clear. Firstly, refNets allow to represent contradictions, which can reflect the sometimes inconsistent representations of humans. For example, one proposition may be due to perceived information while another one is inferred from previous knowledge. Secondly, it is possible to represent default knowledge, eg the famous case that penguins are birds, birds can fly, but penguins cannot. Default knowledge is overwritten. Thirdly, refNets are suited for incremental processing, because all knowledge about a concept is localised, ie stored in one spot. This makes them highly efficient with regard to access time, because once a refO is accessed all explicitly represented knowledge about the entity is instantly available. Fourthly, refNets can be changed, ie refOs, attributes, and designations can be inserted, deleted, or modified. For example, the name ‘CK-314’ standing for the flight number can be replaced by another name for the return flight. Or, if the members of The Oscar Peterson Trio, represented by $r_8$, change so that Niels-Henning Ørsted Pedersen ($r_9$) and Terry Clark ($r_{10}$) instead of Ray Brown and Herb Ellis now belong to the trio then $r_8$ has the attribute $\text{parts}([r_5, r_9, r_{10}])$.

6.2 Event representations

Concepts are often implicitly taken to be concepts of objects, not events. For the conceptualisation of events, however, event concepts form the core of the represen-
tation, cf section 3. Just like object concepts, they are building blocks of conceptual representations. The main difference is that events are immaterial while objects are concrete, ie they are ‘out there in the world’ (provided you subscribe to this philosophical standpoint). The third kind of concepts used here are spatial entities. The rationale for assuming spatial entities like locations or paths is analogous to the one for event concepts. Firstly, spatial entities are neither objects nor events, but they can be identified, eg a location near a concrete entity or a path leading towards it. Secondly, they function in the same fashion as the other concepts. See, for example, Landau and Jackendoff (1993) for a discussion that what is located in space (objects) is differently represented than where something is (spatial entities).

Each concept is represented by a refO. The three different kinds of concepts are distinguished by their sort, ie sorts subdivide the conceptual representation into layers, in particular hierarchical layers. The sorts form a subsumption hierarchy. The motion event domain uses the sort hierarchy depicted in figure 8.

A crucial factor for situations concepts are their temporal interrelations. The relations INC uses are based on Allen’s calculus (Allen 1983, 1991 but see also van Benthem 1990). The list of possible representations in the CCR is given in table 1. Allen’s calculus defines temporal relations between time intervals. However, since situations cannot only represent time intervals but also time points, eg a STOP event, the temporal relations in the CCR can also be relations between time intervals and time points as well as relations between two time points, cf Vilain (1982). In the former case a relation rel is written as ·rel or rel·, in the latter case as ·rel·. The after relation, for example, becomes ·after for the relation between a time point and a time interval, after· for the inverse case, and ·after· for two time points. Keeping these kinds of temporal relations apart is important as can be seen from the following example. When a punctual event occurs between two time intervals, these time intervals cannot be in the meets relation, because the time point would be in a gap between the two intervals. However, when the time point is the endpoint of one interval, ie part of this interval, it must also be part of the other interval (its starting point), which means that the intervals are not in the meets but in the overlap relation. Using different sets of temporal relations prevents such interferences, and the
temporal relations involving punctual events allow to perform temporal reasoning in an Allen-like style (Vilain 1982). For a detailed discussion of the problem of temporal relations in the context of incremental conceptualisation see Guhe et al. (2004).

7 Pre-processing unit (PPU) and perceived entities (PEs)

There is no sharp boundary between perceptual and conceptual processing, cf, for example, Barsalou (1999, p 588). However, this idealisation is usually made in order to keep models manageable. INC is a model of conceptual processing only; all perceptual pre-processing is done in a separate component, the pre-processing unit (PPU). The output of the PPU is the input to INC (more precisely: to construction). For each domain INC operates upon there is a different PPU. (INC can also generate verbal descriptions of the drawing of sketch maps, cf Guhe and Habel 2001) The increments generated by the PPU are called perceived entities (PEs). This section gives a sketch of how PEs for motion events are computed, but the exact way this is done is beyond the scope of this article. The approach is similar to the one proposed by Gärdenfors (2000), who describes a method to ground symbols in (mainly physical) dimensions so as to escape the symbol grounding problem (Harnad 1990).

PEs are computed from spatio-temporal coordinates. For motion events the input to the PPU consists of sequences of quintuples of the following form:

\[(\text{frameID}, \text{objectID}, \text{x-coordinate}, \text{y-coordinate}, \text{direction})\] (2)
For example, a sequence of
\[(1, 1, 50, 100, 90), (2, 1, 54, 112, 89), (3, 1, 52, 127, 91)\] (3)
of positions of an object, say, a plane, with the identifier 1 is translated into a PE representing a straight movement in the time interval \([1, 3]\) from coordinates \((50, 100)\) to \((52, 127)\). Each time frame is 50 ms long, so the sequence lasts 150 ms. The current direction of the object is the most recent value, here: 91. As can be seen in this example, the PPU must cope with noise, because the object is not moving along a completely straight path. Its values on the x-axis are 50–54–52. Nevertheless, on the conceptual level it is desirable to consider this movement as straight. For this example a triple of PEs is generated that is translated to the following referential net by construction:

```
object       r1       \eta x \text{plane}(x)
  \eta x \text{chpos}(r2, x, r3)
  pe(0)

situation    r2       \eta x \text{chpos}(x, r1, r3)
situation    pe(1)
at_time(1, 3)  

path         r3       \eta x \text{chpos}(r2, r1, x)
  \eta x \text{straight}(x)
  pe(2)
```

(The direction information is currently not used.) The at_time attribute serves to synchronise INC to the points in time when the events occurred. Note that the number identifying the object in the quintuples above is different from the one in the pe attribute. The former is generated by the program displaying the objects on the screen and enables the PPU to identify the objects, the latter is used on the interface between PPU and INC for referring to PEs. This is, for example, necessary when the PPU informs INC about an update of a PE, eg if the object continues the straight movement in the next time frame.

The PPU segments movements according to two criteria:

1. an object changes between moving and not moving, and
2. the shape of the path changes.

An example of the first case is given in the introductory scene. When the plane stops the situation in which it moved ends, and a new situation starts in which it is not moving. An example of the second case is that a plane that was moving on a straight path begins to follow a curved path. Then, the point where the plane commences the curved movement is the segmentation point between the two situations.
This segmentation criterion may mean that corrections of already produced PEs may be required. Consider the case that the fourth quintuple is (4, 1, 60, 142, 90), making the sequence of x-coordinates 52–54–52–58. Assuming that the third value deviates due to noise the evidence for a curved movement is now quite strong, while the justification for assuming a straight movement got weaker. If the PPU decides to revise its decision all PEs starting from frame 1 must be changed.

Apart from the sharp distinction between perception and conceptualisation I assume another idealisation: the unidirectionality of the information flow, ie no information is given back from INC to the PPU. Cast in the INC terminology this means that the PPU is not sensitive to results of computations of INC, eg elements of the conceptual representation that are expected could facilitate processing of the next input to the PPU. However, the unidirectionality is a valid idealisation, because INC focuses on the data-driven aspects of conceptualisation, and the bottom-up (data-driven) information in most cases dominates over top-down information [Barsalou 1999 p 588f]. That is, information from perception to cognition exerts more influence than information flowing in the opposite direction.

A final remark on the term perceived entity. These entities are actually perceived and do not belong to a fixed level of granularity. Thus, they are no basic entities and in particular no basic level entities. For example, a motion event can be perceived as consisting of sub-events or as a whole. As far as INC goes this might even be true for the rather complex DOCKING concept (which may then be analysed into its sub-events). INC is designed in a way that it can deal with both, ie it is able to segment one motion event into sub-events on the basis of conceptual knowledge, and it is able to group motion events to more complex events. Currently, however, only the second possibility is implemented.

8 Construction – building up the CCR

Conceptualisation starts with the construction of a conceptual representation. For INC this means that a stream of PEs is read in, from which a hierarchical representation of the external states of affairs is built up: in the introductory example a DOCKING event consists of sub-events. Levelt does not mention the important question of where input comes from and that the structure of this knowledge depends on perceptual factors. He simply says that the conceptualiser has access to different kinds of knowledge. The building up of hierarchical conceptual representations, however, is a complex task.

The main part of constructing a conceptual representation is the matching of perceived entities onto the conceptual knowledge that is available in long term memory. In most cases this means that more complex elements are inserted into the con-
ceptual representation that group together PEs. If a concept available in long term memory is matched only partially and if there is a possibility that the missing parts will arrive as the next PEs, an expectation can be generated. An example of such an expectation in the example scenario is that the WALKWAY stops moving when it reaches the PLANE. After the STOP has occurred it becomes a regular event. (Note that a human will immediately and invariably say that it stops moving in order not to damage the PLANE. INC makes no such sophisticated inferences. The expectation is generated, simply because the STOP is part of the overall DOCKING event.)

In the remainder of this section I will show how the conceptual representation for the docking example is built up incrementally and point out particulars of the representation, cf also Guhe et al. (2003a, 2004) on this issue. The final state of the CCR is shown in figure 9. Since the refNet is rather large, I use a non-standard numbering of refOs. It groups refOs belonging to one movement and refOs that have the same sort. \( r_{6P} \) is the path of the plane movement, \( r_{6W} \) the path of the walkway movement, \( r_{7S} \) is the location of the start of the walkway movement, \( r_{7F} \) the location of the final point of both movements. The subscripts of the situation refOs indicate the order in which the segments of the respective overall situation occur. However, this is just a naming convention; using different names would not change the referential net.

To take up the motivating setting from section 3 again, when you look out of the window while waiting to board you will first see a configuration of objects:\[11\]

\[
\text{gate} \quad r_1 \quad \eta \times \text{gate}(x) \\
parts([r_2]) \\
\quad \quad \quad \quad 'B21' \\
\]

\[
\text{walkway} \quad r_2 \quad \eta \times \text{walkway}(x) \\
part_of([r_1]) \\
\]

Thus, the initial state of the conceptual representation contains the two refOs that stand for the GATE and the WALKWAY. The fact that the WALKWAY is part of the GATE is represented by the attributes parts and part_of that establish the part-of hierarchy between refOs. Both are list attributes and are inverse attributes, i.e., using one would suffice, but using both has the advantage of having faster access to the information. The designations represent that \( r_1 \) can be described as a gate, as B21, or as gate B21 if two designations are used in the verbalisation, and \( r_2 \) can be described as a walkway.

When the PLANE enters the scene in phase 1 (see figure 2.1) three refOs are simul-
Figure 9. Referential net for the example of the docking plane
The situation refO (r4\textsubscript{i}) and the three chpos descriptions that come with it connect the triple. The descriptions represent the fact that in the situation (r4\textsubscript{i}) an object (r3) moves along a path (r6\textsubscript{p}). This means, the sort frame of chpos is (situation, object, location). Since r4\textsubscript{i} represents the first situation of the scene, it has no temporal relation. –complete says that the situation is extended, i.e., takes place in a time interval, and that it is not yet completed. Situations having the –complete attribute are ongoing situations in a scene. The set of ongoing situations is important, because for these situations the temporal relations are established, which are explicitly represented. The ongoing situations are also important, because the temporal relations between them cannot be uniquely determined. The reason is that the situations are not yet fully known [Guhe et al., 2004]. Temporal relations between other situations are computed only if they are required. The description η\textsubscript{x} straight(x) of r6\textsubscript{p} represents the shape of the path of the movement. Note that paths are no real-world objects but abstract, ‘invisible’ spatial entities [Eschenbach et al., 1999, 2000]. The description η\textsubscript{x} to(x, r1) (sort frame (path, object)) says that the path is leading towards r1, which is the gate.

This representation already allows some inferences. The description η\textsubscript{x} to(x, r1) says that the path is leading towards the gate. Together with the fact that in the situation a change of position is occurring (because of the chpos descriptions) this leads to the inference that there is an overall situation (r4) in which the goal of the movement is the gate. The sort frame of goal is (situation, object).

The next inference is already indicated in r4 by the parts attribute: the complex situation brings with it additional simple ones. Although the plane reaches the gate only in phase 2 of the scene, this is anticipated in phase 1 by adding two refO\textsubscript{s} to the representation.
The fact that these two refOs were not actually perceived but are only anticipated is represented by the status attribute, which can have the values expected, regular, or discarded. RefOs of which all parts are regular are regular. Expectations that were not fulfilled are discarded. For reasons of brevity I will only specify the status attribute when it is relevant; the refOs in figure 9 are all regular. For the moment I simply assume that INC introduces these expectations and also that it does not introduce others. What is more, I assume that the overall situation (r8) is not expected at this stage. This reflects a ‘medium willingness’ of the speaker (INC) to generate expectations. This willingness can be varied in INC by setting the parameter DOAT (degree of agreement threshold). A low value causes that no expectations are generated, with a high value INC would also create r8. Expected events are considered for verbalisation by selection just like regular ones.

At this stage the first temporal relations are inserted (r41 is updated accordingly). r42 finishes the PLANE’s movement and starts the situation where it stands still (r43). Since r42 represents a punctual SITUATION, it has no temporal extension, r41 and r43 are in a meets relation. Adding these refOs to the CCR causes the insertion of yet another refO (r7F), representing the LOCATION where the STOP event takes place and where subsequently the PLANE is standing. This location is next_to the GATE and the WALKWAY. It is also the final point of the PATH r6p. Again, r1, r2, and r6p are updated with the according descriptions.

In phase 2 of the scene the PLANE stops next to the GATE at the location already introduced as r7F. This has the effect that the two situation refOs r42 and r43,
which were introduced with status(expected) now get status(regular). So, during this phase no new refOs are inserted into the conceptual representation, but the representation is modified. For the movement of the WALKWAY in phases 3 and 4 the conceptual representation changes analogously. During these phases the DOCKING refO (r8) is created as well. A full representation would also include a refO r50 for the SITUATION when the WALKWAY is at LOCATION r75.

Note that there are many other event sequences constituting a DOCKING. Note also that this is a conceptual representation from which language is produced, ie it is a representation for speaking [Slobin1996]. However, it is likely that representations for other purposes, eg of the pilot planning the plane movements, are similar.

Concept matcher. The concept matcher starts processing when it receives a candidate list from construction. The candidate list is computed with a heuristic by construction and contains the refOs that are likely to form a more complex concept. It is matched against concepts stored in the concept storage (CS). For example, the heuristic collects recent movements performed by the same object, eg a chpos and a stop situation, which can likely be grouped to a more complex event. The chronological order of events constitutes an event sequence – a prerequisite for matching them onto a more complex event. Concepts representing complex events are similar to the scripts of Schank and Abelson (1977).

The concept matcher determines the best match of the candidate list and the concepts stored in the CS. The best match is the concept having the highest degree of agreement (DOA) with the candidate list. If the DOA is greater or equal than the degree of agreement threshold (DOAT), one of INC’s parameters, construction inserts the best match into the CCR. If DOA < 1.0 the best match may contain refOs that are not yet perceived, ie that are expected. If a different best match is computed the next time the concept matcher is called, the expected refOs become discarded. If the same best match is produced and if it contains refOs that are now actually perceived, the expected refOs become regular. The exact way the concept matcher works can be found in Guhe (2003a) but is beyond the scope of this article.

The CS can only be accessed by the concept matcher, and the concept matcher is connected only to construction. Thus, CS and concept matcher could be integrated into construction. However, the presented architecture has the advantage that the representation corresponding to long term memory is a separate structure and that the access to it is modelled by a separate process. In this way the concept matcher can be seen as interface to long term memory.
9 Selection – deciding upon events to be verbalised

The next step of conceptualisation towards the generation of preverbal messages is selecting events from the hierarchical conceptual representation for verbalisation. Since INC is specialised on conceptualising and verbalising events, selection only considers events for verbalisation. Verbalisation of other concepts, say objects, works along the same lines, but INC would have to be extended accordingly. Selection corresponds to generating sub-intentions as discussed earlier. Depending on the values of INC’s parameters and on the temporal interplay with the PVM-generation process it decides, for example, whether the sub-events of the DOCKING event are verbalised, resulting in a four-utterance verbalisation of the scene as it is given on page 5, only one utterance like (4), or perhaps a mixture of both like (5).

(4) *Ein Flugzeug dockt an.*
   ‘A plane is docking.’

(5) a. *Ein Flugzeug bewegt sich auf ein Gate zu.*
   ‘A plane is moving towards a gate.’

b. *Es stoppt beim Gate.*
   ‘It stops at the gate.’

c. *Es dockt an.*
   ‘It is docking.’

For the online description of events the selection task can be reduced to looking at each change that the construction task makes in the conceptual representation. This is possible because of the data-driven mode of operation. (If INC worked memory-driven the major difference would be that construction retrieves its input not from a PPU but from long term memory instead. Selection would not need to be changed.) The selection task decides whether the changed event will be verbalised. In a full account of the selection task a lot of factors must be considered in order to determine the appropriateness of the selected content, eg hearer model, discourse model, social appropriateness, estimation of reliability (too speculative information is not verbalised), etc. Due to the neglect of a hearer model and the data-drivenness, two selection strategies proved to be particularly successful. They make use of the fact that pointers to the event refOs that are selected for verbalisation are stored for a latency in the traverse buffer, cf section 4.

The first selection strategy can be phrased as: *always take the most complex concept under consideration.* In this selection strategy each new or modified event is evaluated whether it has already been selected for verbalisation but not yet verbalised. In other words, it is checked whether there is already a pointer to this event in the traverse buffer. If this is the case then there is no need to select it a second time, because this would lead to two utterances describing the same event. The other part of this selection strategy is that the most complex event under consideration is selected. For example, if a pointer to r8, leading to utterance (4), is stored in the
traverse buffer then no sub-events like r41 are selected – they are already described by verbalising the complex event. Conversely, if r41 is in the traverse buffer when r8 is inserted into the CCR, the simpler event is replaced by the more complex one so that only an utterance like (4) is generated. This strategy was developed on the basis of empirical studies, described in Guhe and Habel (2001). It conforms to the Gricean maxims be brief, a sub-maxim of the maxim of manner, and do not make your contribution more informative than is required, a sub-maxim of the maxim of quantity (Grice, 1975).

The second selection strategy is: retain the level of granularity if possible. This strategy selects an event if another event of the same level of granularity is present in the traverse buffer. Only if the traverse buffer is empty, events of another level of granularity are selected. The strategy is motivated by the principle that a verbalisation should ideally consist of utterances that are generated at a constant rate. One rationale is that the (imagined) hearer should be kept up-to-date about what is currently happening. With this strategy sequences of utterances like (1) on page 5 can be generated.

A communicative intention does not necessarily lead to exactly one utterance. Instead, one communicative intention may lead to multiple utterances, or multiple communicative intentions may be aggregated to one utterance. However, since this is of no consequence for the purposes at hand, I assume such a one-to-one correspondence as working hypothesis. Similarly, I will leave the relation of utterances and sentences unspecified – utterances being the linguistic correspondents to communicative intentions. The formulator must match utterances onto sentences, because sentence is a grammatical notion.

10 Linearisation – ordering selected events

Linearisation is the task of bringing the selected events into an order that is appropriate with respect to the goal of the discourse. Although linearisation is perhaps the most investigated task of conceptualisation (see, for example, Levelt 1983, 1989), it plays only a minor role here. Firstly, linearisation is the most dispensable task of the four: in contrast to linearisation, verbalisation cannot work without construction, selection, and the generation of preverbal messages. Secondly, the present task provides a strictly chronological order of the input information, so that a reordering of events is expendable to a certain degree. However, the difference in meaning in the examples given by Levelt (1989, p 138f):

(6) She married and became pregnant.
(7) She became pregnant and married.
makes clear the need for linearisation. The different orderings of the two events (marrying and becoming pregnant) are verbalised in chronological order.

However, the principle of natural order proposed by Levelt (1989, p 138): arrange information for expression according to the natural ordering of its content is too simple. Habel and Tappe (1999) show that the order of utterances is the result of far more complex linearisation processes. In particular, verbalising concurrent events (ie multiple events happen simultaneously) means that for the sequential medium language the events have to be brought into a linear order, see also Guhe et al. (2004).

11 Generation of preverbal messages

After the events to be verbalised have been selected and linearised INC produces a preverbal message expressing the speech act. More technically: it takes a pointer to an event refO from the traverse buffer and generates an incremental preverbal message describing the event. This is done by the process pvm-generation (preverbal-message-generation). Levelt (1989) describes the four major aspects of this task: computing the accessibility status of the referents, topicalisation, propositionalisation, and acknowledging language specific requirements. The accessibility status determines whether a referent is known and by which expression it can be referred to. Accessibility depends on the state of the system: which entity has been used last, or, for the data-driven setting used here, which entity has been perceived last. Hence, questions of accessibility can be treated elegantly with incremental processing. Topicalisation is the assignment of the topic role to one of the referents of a preverbal message. Propositionalisation is the task of generating a proposition for a communicative (sub-)intention (event, speech act). A major aspect of propositionalisation is the assignment of perspective, or perspectivisation (Levelt, 1989, p 152–157). Habel and Tappe (1999) even consider perspectivisation as the only task at this stage of conceptualisation. However, conceptualisation is not finished

The accessibility status of a referent is one of the following (in increasing specificity): inaccessible, accessible, in discourse model, or in focus. Inaccessible means that the referent has not been introduced and is not inferable by the hearer. Accessible means the same but the referent is inferable. For example, in

(i) My favourite football team is Arsenal.
(ii) The goalkeeper is a genius.

the referent my favourite football team is neither introduced nor inferable from what has been said so far. Thus, it is inaccessible. The goalkeeper is introduced neither but accessible, because it is inferable by the fact that each football team has a goalkeeper. After these utterances Arsenal and its goalkeeper are both introduced into the discourse and can be referred to easier from now on. The former is in discourse model, the latter in focus.
with perspectivisation: no preverbal message (proposition) has been generated yet.

In the context of incremental processing topicalisation and perspectivisation are no transformational processes (like, for instance, in Ziesche 1997), but are side-effects of incrementality (De Smedt 1990b). The incremental generation of a preverbal message commences with one concept – the concept that is given as first increment of an incremental preverbal message to the formulator. Since in INC concepts are represented by refOs the first refO that is handed on to the formulator ‘sets the stage’. The following refOs depend on this first one, see below. The first increment of a preverbal message is what the preverbal message ‘is about’. The models by De Smedt (1990b) and by Kempen and Harbusch (2003) show that this method allows to build generators without a separate perspectivisation component. Deciding upon the first increment of a preverbal message is at the same time the first part of perspectivisation. For example, the decision, which of the two following utterances will be produced, is already made.

(8) a. *David loves Sarah.*
    b. *Sarah is loved by David.*

If *DAVID* is generated before *SARAH* as increment of the preverbal message (8a) is realised; if the order is inverse it is (8b). The two utterances describe the same state of affairs (*LOVE*(DAVID, SARAH)) but have different perspectives: in the first it is an statement about DAVID, in the second one about SARAH. A second step in taking perspective – not related to the order of increments – is to decide how to refer to an entity. For instance, SARAH cannot only be referred to with her name (*Sarah*) but with many different descriptions: *Anna’s daughter, the mother of Peter’s children, Peter’s wife*, etc.

Finally, the generated preverbal message must meet the language specific requirements of the formulator in the sense that it must account for requirements of the particular language in which the preverbal message will be expressed. For example, if a language overtly marks tense this information must be present in the preverbal message, while it may be left out in non-tense marking languages. Since the example utterances for which INC generates preverbal messages are compared to utterances that were recorded in German, INC implicitly assumes language specific requirements for German.

Since in the Levelt model there is no feedback from formulator to conceptualiser, it must be guaranteed that a preverbal message can indeed be linguistically encoded by the formulator. In the field of NLG the encodability problem corresponds to the problem of the generation gap (Meteer 1990, 1991). Since humans do make mistakes, this is no absolute requirement; a perfect model should even account for these mistakes. Ensuring encodability is a hard problem when the full productivity and complexity of natural language is to be used. Assuring encodability by restricting the possible representations is no solution, because it means that the system
undergenerates, ie the full richness and diversity of natural language is not used. This limits the quality and cognitive adequacy of the system. Assuring encodability by checking whether a preverbal message is encodable before it is handed on to the formulator is no solution either. This contradicts the idea of incrementality, because it interrupts the flow of information from one stage to the next: in order to perform the check the preverbal message must be available completely at a given point in time. This is hardly cognitively adequate – in particular for long utterances. Therefore, encodability is a requirement a conceptualiser must fulfil without explicit mechanism. Whether INC fulfils the encodability requirement can only be determined when it is connected to a formulator though, see also section 12.

A further problem at this stage of language production is the verbalisation problem [Bierwisch and Schreuder 1992]. There is not only no general, simple one-to-one mapping of communicative intentions onto preverbal messages, there is also no one-to-one mapping of the parts of a preverbal message onto lexical items. This latter problem must be solved by the formulator, not by the conceptualiser, because the reasons are that lexical access is performed by the formulator. Additionally, the preverbal message is a (particular) conceptual representation, not a linguistic representation (it is only a pre-linguistic representation).

The following information is required by the formulator [De Smedt 1990a]:

1. semantic concepts (that refer to entities, events)
2. semantic roles between concepts (what he calls case relations or ‘deep’ case)
3. features (definiteness, number, etc)

INC generates the first two kinds of information. Features required by a formulator can only be added when INC is actually connected to one, because this depends on the input requirements of the formulator and on how language specific the preverbal message must be. For example, it is quite likely that a formulator for a language that not only makes a singular–plural distinction but also knows a dual needs a threefold distinction of number in the preverbal message while in other languages a twofold distinction suffices.

PVM-generation’s algorithm bears some resemblance to the so-called ‘incremental algorithm’ by Dale and Reiter [1995] [Reiter and Dale 1992 2000] [13] The Dale & Reiter algorithm is the first of a series of algorithms for the generation of referring expressions like the red table. PVM-generation’s algorithm is much more general, because generating referring expressions is only part of generating a preverbal message. The Dale & Reiter algorithm possesses some psychological plausibility, because its results are in accordance with empirical findings, eg the ones by Pechmann (1984) and because it implicitly observes the Gricean maxims [Grice 1975]. Some extensions are proposed in [van Deemter 2002], and [Krahmer and Theune 2002].

[13] The Dale & Reiter algorithm is only a single one, not a class of algorithms as defined in Guhe (2003a). All of INC’s algorithms are incremental algorithms.
describe a variant for the generation of reduced expressions in a discourse context. For example, *the tiny, red, cute dog* may be referred to simply as *the dog* when it is mentioned for a second time. A proposal of how this algorithm may be cast into graph-theoretical terms, finally, is given by Krahmer et al. (2003), which allows to use results from graph theory for improving the algorithm. This version would be particularly suited to be integrated into INC, because the graph representations can be translated into refNets. All variants of the Dale & Reiter algorithm are natural extensions of PVM-generation for the subtask of generating referring expressions. However, this has not been done up to now.

PVM-generation does not perform a linearisation of increments. This is in agreement with assumptions of models of incrementally working formulators. Such formulators, e.g. IPF by De Smedt (1990a,b) or the Performance Grammar by Kempen (Kempen, forthcoming; Kempen and Harbusch, 2002, 2003), explicitly acknowledge that input increments do not come in a pre-specified order. The latter, for example, uses a slot–filler model for the positioning of increments (phrases/segments), i.e., the generation of an element is delayed if a grammatically required element is still missing.

When reading the following description of PVM-generation’s algorithm you should observe that deciding upon a verbalisation always involves two steps in INC. First INC decides that a refO will be verbalised and only then how to verbalise it. This shows in particular in two places in the model. The first one is the division of labour between selection (which decides that an event concept will be verbalised) and PVM-generation (which decides how it is verbalised). The second place is PVM-generation’s algorithm. RefOs can contain lots of designations, which makes constraints for choosing them indispensable, because not all information provided by the designations is needed in a verbalisation. For example, when talking about SARAH one may need her name but not her favourite colour. However, some designations refer to other refOs. This is again tantamount deciding on verbalising them (step 1). In step 2 it is decided how these refOs (with further designations with further refOs and so on) are verbalised.

The PVM-generation process consists of a main loop and two main functions. The loop monitors the traverse buffer and calls the function *Verbalise* that generates the incremental preverbal message. *Verbalise* calls *SelectDesignations* that selects designations for the refOs that are verbalised. More detailed descriptions of the algorithms can be found in Guhe (2003a) and Guhe et al. (2003a, 2004).

The outer loop watches the head of traverse buffer and waits until the defined latency (parameter LT) expires. (See Guhe et al. (2004) for a more elaborated version of this part of PVM-generation.) It fetches the head of traverse buffer and starts a new current preverbal message. The current preverbal message keeps track of what information was already sent to the formulator as parts of the incremental preverbal message that is currently generated. This enables PVM-generation to perform
certain checks when selecting further information, see below. Then Verbalise is called, which generates the verbalisation for the head of traverse buffer. The loop runs until INC is ready to terminate and no further refOs are in the traverse buffer.

This way of operation implies that an incremental preverbal message must be finished before the next head of traverse buffer can be taken. This has two consequences. Firstly, it is not ascertained that the head of traverse buffer is verbalised immediately after the latency expired. What is more, this means that the head of traverse buffer can be removed from the traverse buffer by an operation of the selection process without being verbalised even after its latency expired. Secondly, the generation of an incremental preverbal message cannot be interrupted, ie selection cannot decide not to verbalise a situation after the generation of the corresponding preverbal message commenced.

Verbalise is called initially with the head of traverse buffer. Since the verbalisation of a head of traverse buffer involves the verbalisation of further refOs, Verbalise calls itself recursively. The first operation is to append the pointer the refO it is currently verbalising to the current preverbal message\(^{14}\) (For reasons of brevity I will not distinguish the pointers in the traverse buffer from the actual CCR refOs the pointers refer to in the following.) Then, a new verbalisation refO is generated. A verbalisation refO is a copy of the CCR refO that is currently verbalised. This is necessary, because, firstly, the CCR refO may change before it is verbalised, secondly, not all information of the CCR refO is used in the verbalisation, and thirdly, different verbalisation of the same CCR refO will usually use different information of the CCR refO. Yet, in order be able to refer to a verbalisation later on, a record of this particular verbalisation must be kept. After the creation of the verbalisation refO it only contains the attributes of the original refO and a link to the CCR refO (attribute verb_of). Of the designations only those actually used in the verbalisation will be added to the verbalisation refO in the next steps. A pointer to the CCR refO is appended to the current preverbal message, and SelectDesignations is called for the new verbalisation refO.

SelectDesignations decides which designations are used for verbalising a refO. Three kinds of constraints are used in the selection of designations, structural, activation, and conceptual. The structural constraint is most important; it forms the ‘backbone’ of the algorithm. It says that only grounded designations are verbalised. If designations refer to no other refO they are directly grounded, eg names or the description \(\eta_x \text{ plane}(x)\) of refO \(r_3\) in figure 9. Designations pointing to other refOs, eg \(\eta_x \text{ chpos}(x, r_3, r_6P)\), are not directly grounded, thus it must be tested whether they are groundable. In the chain of grounding, all refOs that the designation contains are checked whether they have a grounded designation. In the example this

\(^{14}\)The current preverbal message does not contain verbalisation refOs (see below) but pointers to refOs in the CCR. This is the prerequisite for detecting conceptual changes, cf Guhe and Schilder 2002.
must be done for r3 (the PLANE) and r6p (the PATH). If a designation is groundable by a chain of grounding the designation is indirectly grounded. A designation evaluated in the chain of grounding is cyclic if it has already been used in this chain, and the grounding attempt fails. Assume, for example, that in a verbalisation of r4 the description ηx chpos(x, r3, r6p) has been chosen and is now part of the current preverbal message. Now a grounded designation for r3 must be found. One possibility would be to select ηx chpos(r4, x, r6p). However, choosing this description would add no new information to the current preverbal message, because both designations can be reduced to chpos(r4, r3, r6p). For this reason it is a cyclic designation that cannot be used to verbalise r3. This also holds for r6p and ηx chpos(r4, r3, x).

The activation constraint evaluates the activation values assigned to the designations in question. Designations obtain their activation when they are inserted into the CCR, depending on their perceptual–conceptual prominence. For example, the property of the path refO r6p of being a straight path is valued as 0.6, its partaking in a chpos as 0.5, and its being a finalpoint as 0.1. Currently these values are theoretical stipulations, but they do not contradict the verbalisations recoded so far (Guhe 2003a; Guhe et al. 2003b). The activation constraint checks whether the activation of a designation is above the activation threshold (parameter AT). However, there are exceptions where a designation is chosen when the activation is below the threshold (Guhe 2003a; Guhe et al. 2003a,b).

Conceptual constraints evaluate whether a designation is conceptually consistent with earlier chosen designations. Currently, there is only one conceptual constraint, the homogeneous-part-of constraint. If a refO is tested in the chain of grounding that is part of another refO in the current preverbal message and the refOs in question are of a homogeneous sort, no designation from the refO will be used in the verbalisation. An entity is of a homogeneous sort if its parts are of the same sort as the entity. An example is path, because paths are extended entities. This means that points lying on a path are not parts of the path but coinciding with it (Eschenbach et al. 1999, 2000). Thus, parts of paths are paths and a path can have multiple sub-paths, but a point, eg a location, cannot be part of a path.

After the designations are added to the verbalisation refO it is complete. It is sent to the formulator as next increment of the incremental preverbal message and appended to the traverse. If there are further refOs that need to be evaluated Verbalise calls itself for the refOs that the verbalised designations refer to.

This completes the generation of one incremental preverbal message. For a discourse a sequence of more than one preverbal message must be generated, eg for generating the verbalisation given in example (1). For the purpose of this article this would lead to far, but see Guhe et al. (2004) on how it is done.
The underlying incremental preverbal message for an utterance like (1a) that is produced by PVM-generation looks as follows:

\[
\begin{align*}
\text{situation} & \quad \nu_1 \quad \eta x \ chpos(x, r_3, r_6) \\
\text{---complete} & \quad \text{verb}_\text{of}(r_4) \\
\text{plane} & \quad \nu_2 \quad \eta x \ plane(x) \\
& \quad \text{verb}_\text{of}(r_3) \\
\text{path} & \quad \nu_3 \quad \eta x \ to(x, r_1) \\
& \quad \eta x \ chpos(r_4, r_3, x) \\
\text{gate} & \quad \nu_4 \quad \eta x \ gate(x) \\
& \quad \text{verb}_\text{of}(r_1)
\end{align*}
\]

The refO terms have the prefix \( \nu \) instead of \( r \) to indicate that they are verbalisation refOs and not refOs of the CCR. The attribute \text{verb}_\text{of} \) specifies of which CCR refO they are a verbalisation. The sequence in which they are given here reflect the order in which they are output by PVM-generation. In this section I will demonstrate how these verbalisation refOs correspond to the formalisation proposed by Jackendoff \((1990)\). For this, two points must be kept in mind. Firstly, Jackendoff’s conceptual structures fulfil the function of preverbal messages \((Levelt 1989\ p\ 71ff)\). Thus they are different from the CCR. Secondly, he implicitly uses a comprehension perspective. In his examples he accesses a lexical conceptual structure (LCS) by lexical items like \textit{into} or \textit{run}. Assuming one lexicon for production and understanding an LCS can be accessed by its structural properties under the production perspective.

A formalisation of the example (1a) according to Jackendoff \((1990: 45ff)\) is:

\[
[\text{Event}\ \text{GO}([\text{Thing}\ \text{PLANE}], [\text{Path}\ \text{TO}([\text{Thing}\ \text{GATE}])])]
\]

\((9)\)

The construction of this structure starts with the first increment of the preverbal message, the situation refO \( \nu_1 \), which can be translated to:

\[
[\text{Event}\ _ ([\text{Thing}\ _] _)]
\]

\((10)\)

where ‘\_’ are placeholders for items that are not known yet or are left unspecified. From the chpos description the formulator knows the basic predicate argument structure:

\[
[\text{Event}\ \text{GO}([\text{Thing}\ _], [\text{Path}\ _])]
\]

\((11)\)
Since a positional change takes place, the predicate must be GO and the second argument must be a Path. The manner of motion is not specified here but could be inferrable from further refO attributes that specify it. This would narrow down the number of possible adequate lexemes. In fact, it may be the case that lexical access can already take place, because the necessary knowledge is available to the formulator. In Jackendoff’s approach this kind of knowledge is given as features, which are currently not implemented in INC, cf section 11. The −complete attribute specifies the structure further by the fact that the motion event has not ended yet, so we can expand the Path to:

$$[\text{Event GO}([\text{Thing }], [\text{Path TOWARD}([\_])])]$$ (12)

In case the preverbal message contained a +complete attribute, ie the movement is finished, the structure is:

$$[\text{Event GO}([\text{Thing }],[\text{Path TO}([\_])])]$$ (13)

Now the two still open argument positions must be filled up. The next increment is v2, the verbalisation refO for the plane. Via the chpos description it is identified as the first of the two missing arguments, so that with an application of Argument Substitution [Jackendoff 1990 p 51] we get:

$$[\text{Event GO}([\text{Thing PLANE}],[\text{Path TOWARD}([\_])])]$$ (14)

Finally, as soon as the formulator obtains v3 and v4 the path and the object where it is leading to are further specified by the to description:

$$[\text{Event GO}([\text{Thing PLANE}],[\text{Path TOWARD}([\text{Thing GATE}])])]$$ (15)

This is identical to (9), the structure given above.

## 13 Conclusions

**Summary.** The first stage of language production in Levelt’s (1989) model is conceptualisation, the corresponding component the conceptualiser. It is the bridge between the world (more precisely: perception) and language. Due to the complex, open-ended nature of conceptualisation developing computational models of the conceptualiser proves difficult. With the incremental conceptualiser (INC) I presented a first computational model within Levelt’s framework. INC reads perceived entities originating from observing events and generates preverbal messages, which can be encoded linguistically by a subsequent formulator. The preverbal messages contain a description of the observed events. The four stages of conceptualisation are the construction of a current conceptual representation (CCR), selection of situations to be verbalised, linearisation of the selected situations, and generation of incremental preverbal messages.
INC emphasises the incremental mode language production. This means that a sequential information stream is simultaneously processed on multiple stages in a piecemeal fashion. Slobin’s (1996) notion of thinking for speaking becomes a thinking while speaking here. INC adheres to two processing principles. Firstly, Extended Wundt’s Principle says that not only is the characteristic input of a component processed as soon as it is available (this is stated by Wundt’s Principle) but that the component also produces output as soon as possible. Secondly, no direct feedback is given but only indirect feedback. This kind of feedback does not consist of the direct transmission of information but of modifying a memory or buffer shared between two or more components.

Discussion. Although Levelt’s model specifies some constraints for conceptualisation, additional problems needed to addressed. Therefore, I added enough detail to the conceptualiser so that it can be implemented as a working model: INC. The most important additions to Levelt’s treatment of the conceptualiser are that the origin of represented knowledge is important and that the temporal aspects of processing tell a lot about conceptualisation. In the ‘age of embedded cognition’ probably only little convincing is needed that work on conceptualisation benefits from using a data-driven approach. This helps to determine the content and organisation of knowledge representations, and it helps to overcome the open-endedness of the task. Based on the data-drivenness the temporal aspects of conceptualisation I elaborated on the temporal interleaving of different stages of conceptualisation, for example the interplay of selection and PVM-generation. Similarly, the interplay of different memory systems like long term and working memory (reflected by the differentiation of previously available knowledge in the CS and perceived and conceptualised knowledge in the CCR) sheds light on how a conceptual representation is built up. This was supported by generating event descriptions instead of object descriptions (which is mostly done in conceptualisation research), as this focusses on the permanently changing state of affairs. (For this reason INC’s conceptual representation is called current conceptual representation.)

Further down in the conceptualisation processing criteria are needed in addition to Levelt’s model for selecting events. Furthermore, an algorithm is required to generate preverbal messages incrementally, an aspect usually neglected in language generation, but vital for a cognitively plausible model. There are two reasons for this. Firstly, the conceptual representation can change while the preverbal message is generated (see Guhe and Schilder (2002) on this issue), secondly, despite their limited resources humans are (in principle) capable of producing infinite utterances. This can only be done by an incremental model of generation. INC uses parameters to reflect how cognitive resources influence different verbalisations of the same scene (Guhe and Habel (2001), Guhe et al. (2003b)).
**Outlook.** The risk of doing interdisciplinary research is to satisfy no one. ‘While writing the text, I felt like a centaur, standing on four legs and waving two hands. The four legs are supported by four disciplines: philosophy, computer science, psychology, and linguistics . . . ’ (Gärdenfors, 2000, p ix). Covering a large area always means that many problems cannot be treated as detailed as they should; instead, the value of the work lies in integrating different approaches. Consequently, INC is not a very ‘deep’ model in a number of areas, but it demonstrates how different stages of conceptualisation interact with each other.

INC needs to be equipped with more sophisticated models of categorisation (concept matcher) and memory (CCR, CS). It is not clear yet how general the presented selection strategies are. However, they do fit the verbalisation data collected so far, and it seems likely that they are applicable to data-driven settings in general. The linearisation process must be fully developed. INC can already produce multi-utterance output (Guhe et al., 2004), which is the prerequisite to investigate the different effects of linearisation. For PVM-generation the next important step is to integrate a version of the Dale & Reiter algorithm for generating referring expressions. Furthermore, more detail must be added to INC’s output, ie the preverbal messages must be enriched by features in Jackendoff’s and De Smedt’s sense so that INC can be connected to a formulator. Finally, although INC reproduces recorded verbalisations to a certain extent more simulations and validation with empirical data are required in order to improve INC’s cognitive adequacy.

**References**


Kempen, G., forthcoming. Grammatical performance in human sentence production and comprehension, Experimental Psychology Unit, Department of Psycho-


