Methodological Aspects of Computational Semiotics

Alexander Mehler
Linguistische Datenverarbeitung/Computerlinguistik, FB II
Universität Trier
Universitätsring 15
D-54286 Trier
mehler@uni-trier.de

© This paper is not for reproduction without permission of the author(s).

ABSTRACT

In the following, elementary constituents of models in computational semiotics are outlined. This is done by referring to computer simulations as a framework which neither aims to *describe* artificial sign systems (as done in computer semiotics), nor to *realize* semiotic functions in "artificial worlds" (as proposed in "artificial semiosis"). Rather, the framework referred to focuses on preconditions of computer-based simulations of semiotic processes. Following this approach, the paper focuses on methodological aspects of computational semiotics.

1 INTRODUCTION

The field of *Computational Semiotics* (CS) has emerged in order to account for the intersection between semiosis and machine computation (Clarke 2001). Any definition of computational semiotics hinges on a notion of sign and sign processes used as its starting point. In this paper we will not try to (re-)construct this fundament. Thus, this paper does *not* define computational semiotics, but outlines only some of its methodological implications and replications. Since the author's object of interest is given by natural language discourse systems, it is a trios of schools which needs to be taken into account for this task:

1. Peirce (1993) proposes a dynamic, relational semantics, which describes meaning as a result of processes of ongoing sign interpretation resulting in the constitution/modification of behavioural dispositions restricting the usage of signs in exactly these processes. Without following the differentiation of Peirce's triadic notion of sign, it is this kind of "circularity" which we believe is crucial for a semiotic grounding of CS, i.e. the fact that signs do not only participate in sign processes on the basis of dispositions (usage regularities), but may also – as a result of their participation – *change* these dispositions. The implications of this notion for the linguistic concept of *rule*, or more generally: *regularity*, are manifold, since

Peirce's dynamic perspective does not allow to conceive them as categorial, static entities.[1]

2. Whereas Peirce does not pick the distinction of syntagmatic and paradigmatic regularities – especially with respect to textual units above the level of argumentative structures – as a central topic of his sign philosophy, this distinction, which parallels the distinction of text system and language system, is fundamental for Hjelmslev's glossematics (Hjelmslev 1969). As a consequence, Hjelmslev can describe linguistics as a formal, deductive approach which – starting from the text as an un-analysed whole – tries to reconstruct the system of choices defining the language system and the realisations of these choices defining the system's textual instances. Regardless of the deductive approach of glossematics, it is the dichotomy of syntagmatics and paradigmatics which is seen to be essential for a semiotic grounding of CS.

3. In contrast to Hjelmslev's static notion of language system, it is the dynamics of mutual constitution of text and language system which is stressed in Halliday's *systemic functional linguistics* (SFL) (Halliday 1977), thereby pointing to the context-sensitivity of linguistic processes. In this sense, a text being produced/received as a discourse unit by participants of a speech community always has at least two (types of) contexts: the system of semantic/linguistic choices underlying it, and the (type of) social context, which SFL differentiates according to situational factors (whose recurrent combinations are described as *registers*) and the staging of social interactions (described as *genres*) (Halliday 1977; Martin 1992).[2] As a consequence of this context-sensitivity, linguistics cannot be conducted as a purely deductive discipline, but necessarily builds on a qualitative as well as quantitative analysis of synchronic and diachronic aspects of the dynamics of linguistic structures. Nevertheless, syntagmatics and paradigmatics remain constitutive terms in SFL, but are now – so to say in accordance with Peirce – analysed from a *dynamic* perspective. It is the dynamic co-evolvement of text and system as well as its context-sensitivity – especially its embedding into *social systems* – which is seen to be essential for a semiotic grounding of CS.

The unification of these different notions aims at a unification of structural and procedural aspects of linguistic modelling:

- The resulting models are *structural* in the sense that they model signs with respect to their informationally uncertain syntagmatic and paradigmatic *regularities*.
- They are *procedural* in the sense that they represent these regularities not only as a precondition, but also as an output of sign processes. As being dynamic entities, the processes involved are necessarily modelled by means of *procedures*.

The synthesis of structural and dynamic aspects prevents two fundamental difficulties of purely structural, symbolic approaches: First, the prerequisite of "atomic meaning con-

[1] In this sense, Peirce's sign philosophy may be seen as a possible basis of corpus linguistics (Stubbs 1996) and quantitative text linguistics (Mehler 2002a,b).

[2] On the level of instances, this contextualisation is complemented by the actual situative context of producing/receiving as well as by the cognitive context of the producer/receiver.

stituents" can be abandoned. Instead of proclaiming a further, empirically non-observable level of semantic atoms, this approach refers to an analysis of the potential of signs for mutual interpretation. Second, language systems are neither analysed in separation from discourse systems, nor from social contexts: Any sign regularities explored are indispensably tied to representations of context types which are explored on their own as semiotic entities.

Without substantiating the unification of these notions of sign systems and sign processes, methodological implications of computational semiotics are focused in the following.

2 SIMULATION, REALISATION, AND EMULATION

The semiotic approach to situated cognition (Varela 1993; Strohner 1995; Rieger 2001) implies that systems of semiotic cognitive information processing constitute their *endo-view* by means of sign processes. These signs are necessarily part of a language system which is constituted by a social system. *Modelling* such semiotic systems necessarily rules out any identification of the model constructor's exo-view with the system's endo-view. It is this problematic identification of endo- and exo-view, which is characteristic for the cognitive and associative (Varela 1993), but not for the semiotic approach (Rieger 2001). Rather, the semiotic approach aims at modelling those processes, as a result of which "semiotic systems" constitute their own environment. In order to throw some light on the term "computational semiotics", we refer to Pattee's (1988) distinction of *simulation* and *realisation* as a basis for distinguishing different approaches to computational semiotics. This is done by complementing the term *emulation*.[3]

1. Following the terms of classical model theory, *simulations* are necessarily homomorphic, formal descriptions of their original – they *symbolically* stand for the entities they model. As a *differentium specificum* counts their function of reconstruction and prediction, respectively. Simulations are dynamic, procedural models of selected functions of their original. Thus, they necessarily comprise *time* as a constitutive parameter. But although simulations homomorphically map any notion of (physical, biological or semiotic) time, they do not *realise* these functions. The simulated functions and structures they operate on remain once and for all ontologically apart from their formal counterparts inside the simulative model. Nevertheless, as being procedural models, simulations introduce a further level of falsification: They are falsifiable with respect to their procedural organisation, and the reconstructions/predictions they produce.[4]

2. *Realisations* are material models *implementing* certain functions of their originals, which exist in the same ontic area as these originals. As simulations, realisations have a procedural organisation, but now, their instances serve to realise the func-

[3] The following outline reflects the author's point of view and does not claim to reproduce already established readings of these terms, nor to determine their definitions.

[4] In this sense, computer-based simulations of cognitive processes are not intelligent; they do not realise intelligent behaviour.

tions in question. Whether this realisation is possible only in the “real” (physical) world, or also in artificial worlds, is a central question posed in artificial life studies. In any case, realisations are evaluated on the basis of the function, they realise, and not with respect to the goodness of the measurements (mappings) they perform. If for example “flying as dragonflies” is the function to be realised, a helicopter may be a good realisation of this function, but a bad model of dragonflies.

3. Let us now add a third term: *emulation*. In this paper, emulations are seen – from the perspective of process – as mappings between two simulating or realising models. Emulations aim at transforming the system being emulated into another, but functionally/simulatively equally effective medium (i.e. a representational format in case of simulations). If, for example, an operating system is emulated by another operating system, it is irrelevant, whether the emulating system really performs the functions of the emulated system in the way the latter does – it is just the equal effect, which counts when the corresponding functions are served in the emulating medium.

In formal terms we can distinguish simulations, realisations and emulations as follows: Let S be a set of systems being modelled, Φ a set of system functions (“flying”, “speaking”, etc.) of systems in S, and M a set of artificial systems or scientific models which serve to model/measure systems in S. Using these simplistic terms, a simulation X can be described as an association of a system and a model, i.e. $X \in S \times M$, where the system is mapped onto the model by means of a homomorphism. In contrast to this, a realisation $R(F)$ is an association of a subset of functions $F \subseteq \Phi$, an artificial system $s \in S \times M$ serving the functions in F – formally: $F(s)$ –, and a subset of systems serving the same functions, i.e.

$$R(F) \in \{ s \in \mathrm{S} \cap \mathrm{M} \mid F(s) \} \times \{ \{r \in \mathrm{S} \setminus \mathrm{M} \mid F(r)\} \}$$

Finally, an emulation E is an association between elements in M, i.e. $E \in M \times M$, where – analogously to simulations – a homomorphism exists between both systems associated by the emulation.[5]

If we conceive computational semiotics as a discipline which builds procedural models of sign-based processes, the decision, whether these models are simulations, realisations, or something in between is more delicate than may be first expected. This can be explained as follows: In a narrow sense there is a “negative” interpretation of simulations as being models of the interaction of system and environment, where the model designer presupposes those results of measuring operations which constitute the environment under consideration. “To presuppose” means that the designer categorically enumerates the universe of all possible contexts of the system by referring to a compositional semantics operating on a finite set of atomic units for context representation (e.g. relations, locations, time variables, etc.) and thereby reducing the system’s meaning relation to a classical *meaning function* (e.g. for system s, the meaning of sign α in context X_{i1} is m_{j1}, the

[5] Clearly, this “formalisation” disregards many constitutive features of simulations, realisations, and emulations. Thus it only serves as a first step toward formalising these terms.

meaning of sign β in context X_{i2} is m_{j2}, etc.). Rieger (2001) critically characterises this view, thereby leading over to the concept of *realisation*, as follows:

> “Trying to understand (conditions of possible) understanding of signs and meanings cannot rely on the *simulative* processing of (symbol) structures whose representational status is declared by drawing on a pre-established semantics (known by the modeler, made accessible to the model, but not at all compulsory for the modeled system). Instead, modeling processes of meaning constitution or understanding will have to realize that very function in an implemented and operational information-processing system which is able to render some structure – in a self-organizing way – representational of something else, and also allows to identify what that structure is to stand for.” (Rieger 2001: 167).

In other words, realisations in the sense of Rieger do not need and – because of the complexity of the environments these *implementational systems* operate on – do not even allow the symbolic enumeration of their possible contexts. Their contextual universe cannot be extensionally enumerated, nor – as the author believes – intensionally specified in purely static terms, but only procedurally defined. Regarding the semiotic approach this means that any static, purely symbolic representation of meaning as a function between enumerable sets, which already presupposes knowledge about all possible referents of all modeled signs in all equally presupposed possible contexts, has to be replaced by a “procedural enhancement” of simulation/realisation models, which allows them to autonomously find out what a relevant context is and what referents of which signs produced/received by the system are adequate in which contexts.

But how to qualify a model in which both, the system and its environment are implemented on a computer? How is a system to be rated in which both, system and environment, are procedural models so that the model necessarily operates in another “physics” than its original.[6] In which sense can such a system be called a realisation? Is it just a simulation? If so, the functions it performs are *not* realised and, consequently, the system *cannot* be seen as *emulating* a real system performing the functions focused upon. Any approach which calls such purely artificial system-environment models “realizations” implies that this model as a whole performs (emulates) the same functions (but not necessarily in the same way) as its original. Clearly, in case of motor-sensory functions this is impossible, but what is about cognitive functions? Are they separable from their bodily realisations and functionally equivalently implementable on a computer? It is well known that computers are universal tools for simulating processes which can be formally represented as recursive functions. Does this commonplace mean that any computer-based model is restricted to the limits of these symbolically representable functions? If so, a computer-based realisation of embodied functions would be impossible, because these models do not leave the symbolic sphere.

The difficulty to find adequate answers for these questions point to the fact that the spectrum spanned by the models discussed so far includes a type of computer-based simulative model, which – equal to simulations – do *not* realise (any cognitive) functions, but simulate their results *as well as* their procedural organisation *without* relying on any pre-established semantics and context models. Comparable to realisations, these models

[6] This is the fundamental question raised in artificial life studies (Pattee 1988).

autonomously perform operations of measurements in their artificial environments at the end of which they semiotically produce contextual representations (according to their own structuredness, needs, etc.). Nevertheless, such a system, when simulating cognitive functions, never becomes intelligent (or cognitive) – it does not realise the functions it simulates.[7] Therefore, it is disputable to speak of emulations in case of these models, since the functions they are supposed to emulate are not realised by their model-theoretic counterpart. Take once more the example of flying: A flight simulator does – in terms of the author – not emulate flying, since there is nobody flying when using the simulator. Moreover, if all, airplane, environment, and pilot are part of a computer-based model, their is less reason to state that the model would emulate flying in a functionally equivalent manner.

Let us now review some approaches in the area of computational semiotics and related fields on the background of the terms outlined so far. More concretely, let us ask what kind of modelling they realise, whether only simulations, emulations or even realisations of semiotic systems/processes:

1. *Computer Semiotics:* Referring to Hjelmslevian structuralism, Andersen (1990) describes semiotically founded principles of the analysis of computer systems understood as *semiotic systems*. He aims at analysing semiotic aspects of software systems as well as of processes of system development and human-computer interface design (and thus distinguishes a product and process perspective). Following the title of his seminal book and separating it from both approaches described below, we call Andersen's approach *computer semiotics*. It deals with computers and related systems as its primary object of interest, but does not imply any decision regarding the question of simulation and realisation.
2. *Artificial Semiosis:* Comparable to the *artificial life* approach (Pattee 1988) and related to *strong AI* (Searle 1980) there has been the proposal to *realise* or rather *emulate* intelligent behaviour by means of artificial, computer-based systems, which – equal to animate systems – experience, for example, their environment, autonomously/intelligently interact with it (Gudwin 1999; Döben-Henisch 2002), or even learn languages. In order to stress the relationship of this approach with strong AI we propose to call it *artificial semiosis*.
3. *The COSINE Approach:* Besides computer semiotics and artificial semiosis a further branch of computational semiotics can be distinguished, which – instead of trying to realise artificial semiotic systems – follows the simulative approach outlined above. The approach of Andersen (2000), who describes genres as self-organising systems, can be seen as an example of it. Comparable to computer-based weather forecast systems used in computational geography, which do not realise weather systems (i.e. they neither produce temperature, rainfall, nor wind), but simulate climate change without pre-establishing all system states and state changes, implementations in the COSINE framework do not realise semiotic systems, but simulate their procedural organisation. As a consequence, the final in-

[7] This is exactly the kind of "autonomous" simulative model which will be meant in the following when using the term *simulation*, leaving out the term's broader definition given above.

terpretation and evaluation of the structures generated by these systems remain on the side of the model constructor/user. In contrast to artificial semiosis, where artificial systems are supposed to operate in an environment equivalent to the one in which their animate originals operate, interpretation cannot be delegated to these allegedly autonomous artificial systems. In order to give this approach a name we will call it *COmputational Semiotics In the Narrow sEnse* (*COSINE*).

It is the latter approach which we follow here. Because of this choice, we will use the terms "computational semiotics" (CS) and "COSINE approach" interchangeably.

3 PROCEDURAL MODELS

Following the terminology of Marr (1982), CS can be characterized as focusing on *procedural* models of sign systems. It asks how to model the processes producing the semiotic structures under consideration. This change of focus of interest is paralleled by a methodological preference of computational models as a means for *simulating* sign processes: Whereas formal semiotic analysis (e.g. algebraic linguistics) is primarily based on calculi for the description of structural aspects of signs, CS integrates both, formal analysis of structural *and* dynamic aspects. This is outlined in figure 1, where the stages of 'conceptual model', 'calculus' and 'description' – ordered by decreasing abstractness – is paralleled by procedures, algorithms (as realisations of procedures) and implementations (as realisations of algorithms). Whereas the branch of conceptual modeling primarily focuses on *descriptions* of semiotic structures – in accordance with the criteria of completeness, consistency, and simplicity as declared in logics –, it is the *procedural reconstruction* of the semiotic structures under observation, which is in the focus of computational semiotics. This reconstruction is tied to the specification of algorithms – possibly enlarging the classes of known algorithms – and their implementation as computer programs.

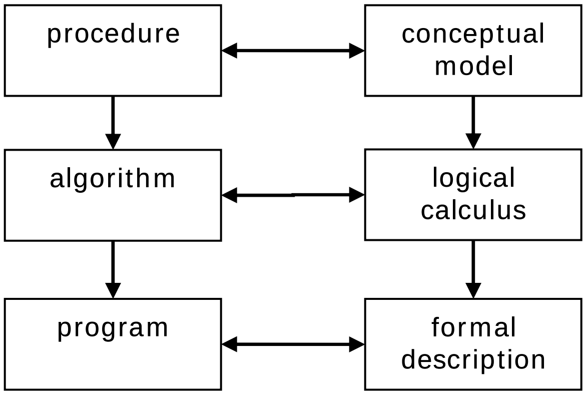

Figure 1: Levels of procedural and conceptual abstractness.

A central question raised in CS is the dynamic interrelation of signs and the situative/cognitive/social contexts of their production/reception: Not only that depending on varying contexts the same sign may be interpreted differently (polysemy), but its ongoing usage has the potential to modify the conditions underlying its actualisation (learning). In spite of the proximity to the area of machine learning, CS departs from approaches which aim at modelling processes of *convergent* learning by referring to prior knowledge. Computational semiotic models of learning are necessarily unsupervised; they refer to procedural knowledge of how to acquire knowledge and allow the learning goal to evolve and thus the learning process to *diverge* temporarily.[8] This departure is reflected by means of the constraint of *procedural interpretability:*

CS demands that the procedural organisation of computational-semiotic models is interpretable in the sense of having semiotic originals.

In spite of this moment of methodological "naturalisation", CS does *not* try to *realise* semiotic machines (comparable to any kind of *hard* AI), but to build information systems which are grounded on sign models reflecting the dynamics of semiotic information processing in order to better fit the demands of their human users.

Let us summarise the conception of CS outlined so far: A computational semiotic model is a *simulative, procedural* model which comprises representational formats

(i) for the representation of system, context and process,

(ii) formats which allow to model the dynamics of system and context change as well as various aspects of informational uncertainty,

(iii) are not only algorithmisable, but also implementable,

(iv) and have a procedural organisation which is semiotically interpretable: they do not simply reproduce certain effects by realising/emulating certain functions, but simulate the *way* in which the original system is performing its function.

(v) Computational semiotic models serve to implement a kind of machine learning mechanism, where the learning system evolves its learning capability as well as its learning targets.

Thus, computational semiotic models necessarily realise a sort of *explorative* information processing, where system and learning environment mutually evolve and are not pre-established by the model-constructor. Computational semiotic models are formalised in algorithmic terms. Structures produced by their implementations are bound to the contextual information underlying their processing. Computational semiotic models are falsifiable regarding their theoretical background, their procedural claims, and the structures they produce (reconstruct/predict).

[8] This conception necessarily departs from the paradigm of *categorisation*, where the learning goal is fixed by pre-established sets of static, unstructured categories whose quantitative characteristics have to be learnt in pre-categorised test sequences. To separate computational semiotic models from this categorisation setting one may say that the former *explore* their respective categories (or meaning representations) as *structured* as well as *dynamic* entities which change in the course of sign processing.

REFERENCES

Andersen, Peter B. 1990. *A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems*. Cambridge: Cambridge University Press.

Andersen, Peter B. 2000. Genres as Self-Organising Systems, in Andersen, P. B. et al., eds., . *Downward Causation: Minds, Bodies and Matter*. Aarhus, 214-260.

Clarke, Rodney 2001. Studies in Organisational Semiotics: an Introduction. *Australian Journal of Information Systems* 8(2), 75-82.

Döben-Henisch, Gerd 2002. Alan Matthew Turing, the Turing Machine, and the Concept of Sign, in Schmitz, W. & Sebeok, T. A., eds., *Das Europäische Erbe der Semiotik, The European Heritage of Semiotics*. Dresden: w.e.b. thelem.

Gudwin, Ricardo R. 1999. From Semiotics to Computational Semiotics, in *Proceedings of the 9th International Congress of the German Society for Semiotic Studies, 7th International Congress of the International Association for Semiotic Studies (IASS/AIS)*. Dresden.

Halliday, Michael A. K. 1977. Text as Semiotic Choice in Social Context, in van Dijk, T. A. and Petöfi, J. S., eds., *Grammars and Descriptions*. Berlin: de Gruyter, 176-225.

Hjelmslev, Louis 1969. *Prolegomena to a Theory of Language*. Madison: University of Wisconsin Press.

Marr, David 1982. *Vision: a computational investigation into the human representation and processing of visual information*. New York: Freeman.

Martin, James R. 1992. *English Text. System and Structure*. Philadelphia: John Benjamins.

Mehler, Alexander 2002a. Textbedeutungsrekonstruktion. Grundzüge einer Architektur zur Modellierung der Bedeutungen von Texten, in: Pohl, I., ed., *Prozesse der Bedeutungskonstruktion*. Frankfurt a. M.: Peter Lang, 445-486.

Mehler, Alexander 2002b. Quantitative Methoden, appears in: Lobin, H. and Lemnitzer, L. (ed.). Texttechnologie. Perspektiven und Anwendungen. Tübingen: Stauffenburg-Verlag.

Pattee, Howard H. 1988. Simulations, Realizations, and Theories of Life, in Langton, C. G., ed., *Artificial Life. SFI Studies in the Sciences of Complexity*. Redwood: Addison-Wesley, 63-77.

Peirce, Charles S. 1993. *Semiotische Schriften 1906-1913*, Vol. 3. Frankfurt a. M: Suhrkamp.

Rieger, Burghard B. 2001. Computing Granular Word Meanings. A Fuzzy Linguistic Approach in Computational Semiotics, in Wang, P., ed., *Computing with Words*. New York: Wiley, 147-208.

Searle, John R. 1980. Minds, Brains, and Programs. *The Behavioral and Brain Sciences* 3, 417-457.

Strohner, H. 1995. *Kognitive Systeme. Eine Einführung in die Kognitionswissenschaft*. Opladen: Westdeutscher Verlag.

Stubbs, Michael 1996. *Text and Corpus Analysis. Computer-Assisted Studies of Language and Culture*. Cambridge, Massachusetts: Blackwell.

Varela, Francisco 1993. *Kognitionswissenschaft, Kognitionstechnik: eine Skizze aktueller Perspektiven. Mit einem Vorwort von Siegfried J. Schmidt*. Frankfurt a. M.: Suhrkamp.