BAMBERGER BEITRÄGE

ZUR WIRTSCHAFTSINFORMATIK UND ANGEWANDTEN INFORMATIK

ISSN 0937-3349

Nr. 82

Proceedings of the KI 2009 Workshop on Complex Cognition

Ute Schmid, Marco Ragni, and Markus Knauff (Eds.)

September 15, 2009
Paderborn, Germany

FAKULTÄT WIRTSCHAFTSINFORMATIK UND ANGEWANDTE INFORMATIK

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

Preface

Ute Schmid[1] and Marco Ragni[2] and Markus Knauff[3]

[1] Cognitive System Group, Faculty Information Systems and Applied Computer Science, University of Bamberg
ute.schmid@uni-bamberg.de
[2] Center for Cognitive Science, Institute of Computer Science and Social Research, University of Freiburg
ragni@cognition.uni-freiburg.de
[3] Experimental Psychology and Cognitive Science, University of Gießen
markus.knauff@psychol.uni-giessen.de

The KI'09 workshop *Complex Cognition* was a joint venture of the Cognition group of the Special Interest Group Artificial Intelligence of the German Computer Science Society (Gesellschaft für Informatik) and the German Cognitive Science Association.

Dealing with complexity has become one of the great challenges for modern information societies. To reason and decide, plan and act in complex domains is no longer limited to highly specialized professionals in restricted areas such as medical diagnosis, controlling technical processes, or serious game playing. Complexity has reached everyday life and affects people in such mundane activities as buying a train ticket, investing money, or connecting a home desktop to the internet.

Research in cognitive AI can contribute to support people navigating through the jungle of everyday reasoning, decision making, planning and acting by providing intelligent support technology. Lessons learned from expert system research of the nineteen-eighties are that the aim should not be to provide for fully automated systems which can solve specialized tasks autonomously but instead to develop interactive assistant systems where user and system work together by taking advantages of the respective strenghts of human and machine.

To accomplish a smooth collaboration between humans and intelligent systems, basic research in cognition is a necessary precondition. Insights in cognitive structures and processes underlying successful human reasoning and planning can provide suggestions for algorithm design. Even more important, insights in restrictions and typical errors and misconceptions of the cognitive systems provide information about that parts of a complex task from which the human should be relieved. For successful human-computer interaction in complex domains furthermore it has to be decided which information should be presented when in what way to the user.

We strongly believe that symbolic approaches of AI and psychological research of higher cognition are at the core of success for the endeavor to create intelligent assistant system for complex domains. While insight in the neurological processes of the brain and in the realization of basic processes of perception, attention and sensu-motoric coordination are important for the basic understanding of the basis of human intelligence, these processes have a much too fine granularity for the design and realization of intercative systems which must communicate with the user on knowledge level. If human system users should not be incapacitated by a system, system decisions must be transparent for the user and the system must be able to provide explanations for the reasons of its proposals and recommendations. Therefore, even when some of the underlying algorithms are based on statistical or neuronal approaches, the top-level of such systems must be symbolical and rule-based.

The papers presented at this workshop on complex cognition give an inspiring and promising overview of current work in the field which can provide first building stones for our endeavor to create knowledge level intelligent assistant systems for complex domains. The topics cover modeling basic cognitive processes, interfacing subsymbolic and symbolic representations, dealing with continuous time, Bayesian identification of problem solving strategies, linguistic inspired methods for assesing complex cognitive processes and complex domains such as recognition of sketches, predicting changes in stocks, spatial information processing, and coping with critical situations.

Ute Schmid
Marco Ragni
Markus Knauff

Board of Reviewers:

Programme of the KI'09 Workshop Complex Cognition

9:00-9:15	Opening
9:15-9:45	Martin V. Butz, University of Würzburg, Department of Psychology III *Sensorimotor Self-Motivated Cognition*
9:45-10:15	Frieder Stolzenburg and Florian Ruh Hochschule Harz (FH), University of Applied Sciences, Department of Automation and Computer Sciences *Neural Networks and Continuous Time*
10:30-11:00	Felix Steffenhagen, Marco Ragni, Ivo Chichkov, Andreas Klein, Center for Cognitive Science, Freiburg *Predicting Changes: A Cognitive Model for Dynamic Stocks and Flows*
11:00-11:30	Oliver Kramer, Technische Universität Dortmund, Department of Computer Science, Algorithm Engineering / Computational Intelligence *On Optimization of the Interface between Subsymbolic and Symbolic Representations and the Symbol Grounding Perspective*
11:30-12:00	Claus Möbus and Jan Charles Lenk, Learning and Cognitive Systems, Department of Computing Science Carl von Ossietzky Universität Oldenburg *Bayesian Identification of Problem-Solving Strategies for Checking the ACT-R/Brain-Mapping Hypothesis*
13:30-14:00	Thora Tenbrink and Linn Gralla, FB 10 Faculty of Linguistics and Literary Sciences, Universität Bremen *Accessing complex cognitive processes via linguistic protocol analysis*
14:00-14:30	Angela Schwering, Ulf Krumnack, Helmar Gust and Kai-Uwe Kühnberger, University of Münster, Institute for Geoinformatics, University of Osnabrück, Institute of Cognitive Science *The Recognition of Sketches as a Test Case for Complex Computational Cognition (Position Paper)*
14:30-15:00	Regis Newo, Klaus-Dieter Althoff, and Werner Greve University of Hildesheim, Institute of Computer Sciences, Institute of Psychology *Conflict Resolution while Coping with Critical Situations*
15:30-16:00	Thomas Barkowsky SFB/TR 8 Spatial Cognition, Bremen *CASIMIR. A Computational Architecture for Modeling Human Spatial Information Processing (Abstract)*
16:00-16:30	Wrap up Discussion and Workshop Summary
17:00-18:00	Meeting of the special interest group Cognition (Thomas Barkowski and Ute Schmid)

Table of Contents

Accessing complex cognitive processes via linguistic protocol analysis

Thora Tenbrink and Linn Gralla

FB 10 Faculty of Linguistics and Literary Sciences, Universität Bremen, Germany
{tenbrink | gralla@uni-bremen.de}

Abstract. Complex cognitive processes are often investigated via elicitation of natural language data. While traditional psychological research typically focuses on the analysis and interpretation of content that is directly expressed in verbal reports, linguistic discourse analytic methods can contribute deeper insights into the processes involved, via highlighting linguistic structures and patterns that the speakers themselves may not be consciously aware of. In this paper, we first present the general method of "Cognitive Discourse Analysis", outlining its main features and analysis procedures in the light of requirements from cognitive science and artificial intelligence. In the second part we turn to a more detailed, exemplary presentation of a study of thought processes involved in object assembly. A process model developed on the basis of the verbal data represents the main steps of the generalized abstract problem solving procedure. Furthermore, the linguistic data reflect a complex interplay of structural and functional object conceptualizations and mapping processes between them.

Introduction

A great variety of everyday tasks involve complex cognitive processes: these include route planning and event scheduling, decision making, using household appliances for specific purposes, and many more. What kinds of thought processes are involved in dealing with such tasks? Much research in the area of cognitive science, in particular cognitive psychology and – increasingly – artificial intelligence has been devoted in the past decades to accessing cognitive processes across various types of task, often for purposes of modelling human ways of thinking, and reproducing them in artificial agents. Quite often, such research involves the elicitation of natural language, either as external representations of current internal processes while solving a particular task in so-called think-aloud protocols, or as retrospective reports which are suitable reflections of the earlier thought processes (Ericsson & Simon, 1984).

Currently, this particular type of linguistic data interpretation remains largely uninformed by linguistic expertise. Usually, cognitive scientists elicit and analyse language for the purposes demanded by the task at hand, without consideration of the particular features of the discourse type they are dealing with. The aim of this paper is to show the extent to which linguistic tools for discourse analysis are suitable for capturing and highlighting aspects of language in use that may be of crucial interest to cognitive scientists, both for purposes of investigating psychological procedures involved in problem solving and other complex cognitive processes, and for purposes of modelling such procedures formally and computationally. As one potential outcome, the design of artificial agents that share particular aspects of human thought may profit greatly from a structured, in-depth understanding of the language used to externalize complex cognitive processes. Natural language is unquestionably the most common medium required and used to convey information between agents; employed in an informed way, it can serve as a fruitful mediator and representation method bridging the gap between computational issues and human thought.

We will start out by presenting the main features of a newly developed method called Cognitive Discourse Analysis (CODA), discussing elicitation as well as analysis procedures that have been successfully adopted so far. The second part of this paper concerns a more detailed, exemplary presentation of our current study concerned with object assembly.

CODA – Cognitive Discourse Analysis

Ericsson and Simon (1984) provide a broad and exhaustive account of previous literature in the area of language data collection along with cognitively complex tasks. Along with this, they discuss the question of the validity of verbal data as such, i.e., the extent to which – and the circumstances under which – participants' accounts of thought processes can be trusted. Their work contains a detailed account of the recommended data collection, annotation, and analysis procedures for verbal data, particularly think-aloud protocols and verbal reports. This approach has since been established as a kind of paradigm which is regularly re-used and adopted for a great variety of purposes.

In this tradition, linguistic features are only seldom accounted for in any way. The analysis of verbal protocols generally focuses on extracting aspects that the speakers are themselves aware of, i.e., the conceptual strategies and processes that they report explicitly. However, linguistic representations may reflect conceptual aspects that the participants take for granted, being unaware of the significance of particular ways of framing a verbal representation. This is the basic motivation for adopting discourse analytic tools in addition to the content-based interpretation of verbal protocols.

The approach of CODA targets a systematic analysis of linguistic patterns by addressing the way *how* some content is expressed or structured in addition to *what* is said. As such, this idea envelops a wide range of achievements and ideas from the field of linguistic expertise; indeed the CODA methodology is flexible enough to allow for, and unite, various different perspectives. However, certain procedures of elicitation as well as analysis may be more suitable for particular purposes than others – both in terms of identifying cognitive processes in general, and in terms of addressing specific research questions in a particular study. Crucially, *text-type related* and *task related* aspects need to be differentiated carefully. On the one hand, some types of linguistic patterns are systematically related to the usage of a particular text type (e.g., Biber, 1989), yielding standard and less standard ways of representing information. On the other hand, a range of systematic aspects in language involve cognitively relevant phenomena such as presuppositional aspects, semantic under-specification, and conceptual categorizations, building a bridge between the available linguistic system and the current topic represented during a problem solving task. Such insights support the interpretation of those aspects of the language data that are in fact peculiar to the task at hand, i.e., that reflect cognitive processes related to the participants' behaviour.

In the following, we briefly sketch a range of elicitation issues as well as analysis procedures that have been usefully adopted in CODA-based studies, along with examples. Concerning elicitation, a main focus will be on the significance of linguistic data types. Analysis procedures, on the other hand, center around systematic patterns in language that may be cognitively relevant for a particular task. Following this overview we will turn to a more detailed discussion of a set of think-aloud data collected during a problem solving task: assembling a dollhouse with limited prior information about the functions of the available parts.

Elicitation in CODA: Significance of linguistic data types

Ericsson & Simon's (1984) framework provides a good basis for identifying the cognitive significance of particular text types. For instance, information verbalized *during* the task and retrospective probing is likely to reflect cognitive processes within short-term memory, while generalized questions *after* the task require intermediate processing influenced by long-term memory. Therefore, *think-aloud protocols* and *retrospective reports* are best suited to elicit unbiased verbalizations of cognitive processes. While this insight motivates a focus on these particular text types, other types of verbalizations have different effects which may also be welcome under certain circumstances. For some purposes, slightly enhanced discourse goals – if well understood and systematically accounted for in the interpretation of the elicited language data – may lead to further useful insights. If the instruction given to elicit verbal protocols along with complex cognitive tasks is formulated in a less neutral way, inducing some kind of bias, this will influence not only the elicited language as such but may also affect the way the participant perceives the task, and thus have an impact on behavior. Under certain circumstances, the requirement to verbalize may promote a better understanding of the task itself – or it may lead to an impairment (Schooler, Ohlsson, & Brooks, 1993). In the following, we will briefly discuss three further widely used text types, which may be suitable for different purposes.

Instructions for other people may trigger intermediate processes of verbalization, such as explanations. Such data may provide insights into how cognitive processes can be conveyed from an expert (in solving a complex task) to a novice. Clark & Krych (2004) present a relevant analysis of dialogues concerned with a joint problem solving task (building a LEGO model), showing how experts adjust their instructions according to their partners' reactions. One important field of investigation within spatial cognition research concerns the analysis of route directions. Here, participants are typically not asked to describe what they were thinking when finding their way, but use a verbal representation to enable another person to find their way (e.g., Denis, 1997). This opens up further possibilities for eliciting language under consideration of different perspectives. Apart from the text type itself, the precise nature of the (perceived) discourse goal (i.e., why language is produced) plays a decisive role, with systematic influences on the level of granularity or detail expressed in language as well as the trains of thought that are triggered by the way the current linguistic aims are understood. A recent study by Wiener, Tenbrink, Henschel, and Hölscher (2008), which involved three different types of linguistic data (think-aloud protocols and written route descriptions "for themselves" and "for a stranger"), revealed that the way a route information is conveyed depends on the perceived relevance of the question for the route receiver, based on previous knowledge, presumed preferences (nice routes vs. shortest option), and so on. Such issues have consequences not only for the way a route instruction is formulated but also on the information itself, i.e., the choice of a route. Moreover, the think-aloud protocols highlighted the incremental cognitive processes involved in the actual wayfinding process, drawing on visual information. Thus, variation in the elicitation of language data led to enhanced insights about a range of crucial cognitive aspects.

A recent linguistic in-depth comparison (Tenbrink, 2008a) of three different text types produced by a single study participant (a think-aloud protocol with a subsequently produced retrospective report plus an instruction "for a friend") in relation to a variant of the *Traveling Salesperson Problem* addressed the distinct perspectives of each data type on the conceptualizations of the problem solving task at hand. The linguistic features of the think-aloud data reflected cognitive chunking and a gradual shift of attention focus with respect to perception and action. The retrospective reports coherently represented those cognitive processes that after a number of trials turned out to be most decisive for this particular person. The instructions formulated for an addressee additionally revealed potentially useful ideas that were not necessarily decisive for the participant's own actions.

Interview questions. Ericsson & Simon (1984) pointed out that questions posed by the experimenter, if not formulated in a very general way, lead to filtering processes and may address aspects that the subjects never actually attended to themselves during the problem solving process (such as reasons and motivations). However, this may not necessarily be a disadvantage. In the analysis of strategies used in particular problem solving tasks, intermediate thought processes may lead to the mention of strategies that could have been used but were not; due to conscious reflection, participants may realize that better performance on the current task could have been achieved. Such a recognition of further possible strategies would in most cases also be reflected linguistically, highlighting the need for detailed linguistic analysis. However, after the task, the motivation for improving performance may be reduced, as is the perceptual input; thus, it becomes even more difficult to imagine good ways of solving the problem. Thus, the main danger consists in participants wrongly believing that they solved the task in a particular way; therefore, a particular kind of verbal data always needs to be controlled against other ways of verbalization as well as against behavioral data. Generally, relying on think-aloud data alone may often not be sufficient since verbalizations during the task may influence behavior under certain circumstances, and they may be incomplete in systematic respects (Ericsson & Simon, 1984). Similarly, Someren et al. (1994) point out that retrospective reports may sometimes omit false leads, i.e., fruitless thought processes that the problem solver discarded after a while.

Dialogue. Apart from the possibility of eliciting dialogues between experts and novices as already mentioned, further variations are possible. Boren and Ramey (2000) suggest extending Ericsson & Simon's approach to a *communication-based* one: they argue for allowing the experimenter to communicate in a fairly natural way with the participant in order to elicit more information and to support the user in exploring the ideas and issues at stake. Krahmer and Ummelen (2004) compare this approach directly with Ericsson and Simon's and find that dialogic interaction during performance appears to have an influence on task success but not necessarily on the contents of the comments being produced (thinking aloud vs. dialogue).

Clearly, when engaging participants in dialogue, or when using questionnaires, one should avoid questions that are theory-driven to such a high degree that they bias participants to the kinds of answers that the researcher is looking for. In CODA, various different verbalizations are triggered, not in the first place by specific questions, but by suggesting *different discourse tasks* to the participant. Thus, participants may be asked to produce verbal representations not only for the purpose of revealing thought processes, but primarily for a different purpose in which these thought processes are again put to use, this time not for behavioral purposes but in order to create a linguistic product. This includes monologic and dialogic discourse, as well as spoken and written language. Spoken language differs from written language, for example, with respect to the usage of certain markers of hesitation (see below), repetitions and self-corrections, lexical choices, typical syntactic patterns, and so on. With the presence of an (active) addressee, dialogue patterns such as alignment, clarification, and adaptation to the interaction partner come into play that influence the amount and representation mode of the information to be conveyed, and therefore highlight different aspects as compared to other discourse types. By systematically eliciting and comparing several such accounts, it is possible to approach the thought processes underlying verbalizations from different perspectives.

Analysis procedures in CODA

Structure and information presentation. The way in which texts (of any type) are structured can be expected to relate systematically to the way the underlying cognitive processes are structured. This concerns both the text as a whole, revealing for instance temporal and causal relationships developing gradually, and smaller portions of the text, for example information packaging within single clauses. Insights from linguistic theory such as Functional Grammar (Halliday, 1994) support the identification of parts of the text that are represented as Given or New, based not only on linear order but also on a range of grammatical features such as presenting vs. presuming reference types. Information presented as Given is linguistically taken for granted, which (if not supported by the previous text) may serve rhetorical purposes or reflect the underlying trains of thought. Information presented as New is apparently "newsworthy" for the speaker. Such effects may be supported by the usage of explicit discourse markers (see next paragraph). Related to our study of route planning under diverse circumstances (Wiener et al., 2008), we analyzed the way in which information about landmarks was packaged in think-aloud protocols in various conceptual situations (Tenbrink, 2008b). The analysis revealed a high amount of occurrences of presuppositions and non-anchored spatial references. For example, the utterance "At the concert hall take the Sedan street in the direction of the theatre" presupposes the location of both *the concert hall* and *the theatre* (i.e., their location cannot be derived from this utterance, though it may be derivable from the earlier discourse); in contrast, due to the spatial anchoring of *the Sedan street* within the utterance, its spatial location can be mentally integrated directly. This reveals the underlying spatial representation on the part of the speaker, where the presupposed locations are firmly anchored but not made prominent, leading to necessary inference processes on the part of the hearer.

Discourse markers. In a line of work on an approach called "psychopragmatics" (Caron-Pargue & Caron, 1991), Caron (1996) identified a number of linguistic markers that may reflect cognitive processes. Particularly interesting in this respect is the usage of connectives: On the one hand, connectives (such as *before, because, while*) serve to explicitly structure the represented contents, revealing how the participant construes the concepts and relations involved. On the other hand, certain markers that are particularly prominent in spoken language may reflect hierarchical thought processes; for instance, occurrences of "Okay, now…" may signal the completion of a subprocess together with the start of a new one. In Tenbrink & Seifert (under review), a route planning task involved the mental combination of two domains, *planning* (based on a map) and *travelling* (in the real world); this combination was systematically reflected by modal markers in retrospective reports.

Lexical choices. The way words and concepts (typically, nouns) are used may be revealing about the role of a particular semantic or conceptual field during a problem solving task. In the analysis of a version of the Traveling Salesperson Problem (Tenbrink & Wiener, 2009), we were interested in the impact of *colour* and *shape* on the path planning processes required for this particular problem solving task. While strategies focusing directly on either one of these concepts were rarely formulated explicitly (which is not surprising since attending to colour or shape did not support the problem solving process in any direct way), the lexical analysis revealed that participants actually relied heavily on concepts of colour, but not shape. In Tenbrink & Seifert (under review), on the other hand, a detailed lexical analysis supported the differentiation of planning and travelling domains based on choices and combinations of words for particular thought processes.

Activity sequences. A focus on the verbs used in verbal protocols reveals the types of activities that are prominent for a participant during a complex cognitive task. According to Halliday (1994), verbs can be classified into a limited number of types according to their basic semantic function; the three main types are verbs of *being* representing abstract relations, verbs of *sensing* representing consciousness, and verbs of *doing* representing the physical world. Each of these types (and some further subgroups) have their own grammatical restrictions as well as functions in discourse. Starting from this classification, a close examination of the development of processes (i.e., usage of verbs and possible nominalizations of verbs) can reveal the particular types of activities that the participants attend to during the task. Such analysis always focuses on whole constructions with verbs at their center, rather than attempting to interpret decontextualized usages. In Tenbrink & Wiener (2009), this type of analysis led to the proposal of an accumulated procedure for solving the Traveling Salesman Problem, generalizing over all collected protocols.

Exemplary study: Object assembly

The lasting success of companies like IKEA suggests that people are willing to assemble their furniture on their own. In general they are aided in their effort by a manual that is supplied by the manufacturer; however, some people are reluctant to use these, or the manual may be missing. Moreover, a situation may occur in which object parts are discovered without information about the composite object that may be assembled from the parts. In such situations, object assembly turns into a problem solving task involving an interesting variety of cognitive processes, resembling earlier findings in other domains (Tversky, Heiser, Lee, & Daniel, 2009). A range of studies have addressed the conveyance of information relevant to an assembly process in situated communication (e.g., Rickheit & Wachsmuth, 2006). In our explorative study, we collected think-aloud data and retrospective reports in an object assembly task, so as to learn more about the cognitive processes involved in solving such problems. A number of studies have shown the impact of prior knowledge on recall (Bransford & Johnson, 1972) and comprehension (Dixon, 1987). In order to address the impact of the amount of prior information on the cognitive processes involved and their linguistic reflections, we tested participants in three conditions. The participants in the first condition were told nothing about the nature of the composite object and thus lacked contextual information altogether. Those in the second condition were told that a dollhouse should be assembled and thus provided with domain knowledge (the general context of the assembly). Those in the third condition were given very specific contextual information on the object and the actual goal state by combining verbal and visual information. In the following we sketch the procedure and analysis involved in this project in order to illustrate procedures of the CODA methodology in practice. As this is work in progress, the analysis is not yet complete; however, we report a range of patterns emerging from the procedure of analyzing think-aloud data.

Procedure

52 participants (graduate and under-graduate Bremen University students, 28 female, 24 male) were presented with a box containing 10 object parts, plus a large roof piece and 2 wooden boards, all of which belonged to a wooden two-story dollhouse by the German toy brand “Selecta”. They were randomly assigned to three conditions. In the first condition (*no goal* condition) they were asked to assemble all given parts in a sensible way. Participants in the second condition were asked to use all parts to assemble a two-story dollhouse in a sensible way (*verbal goal* condition). In the third condition participants were shown a picture of the complete two-story dollhouse for 30 seconds and asked to assemble the depicted dollhouse (*verbal & visual goal* condition). All of the participants were trained and instructed to think aloud during

the assembly, based on Ericsson and Simon's (1984) methodology. After they indicated completion of the task, they were asked to give a retrospective report on the assembly procedure (not analyzed here). The participants were video-taped and their speech was recorded and later transcribed.

Analysis

Structure. Tversky et al. (2009) identified a common structure involved in explanations of construction tasks; across modes (gestures, diagrams, and words), a clear beginning, middle, and end could consistently be identified. We were interested in examining whether similar structures would emerge in think-aloud protocols, which differ from explanations by the lack of an explicit addressee. Based on our data we defined three stages as follows:

- The *beginning* was defined as utterances produced after entering the room and before starting the actual assembly process. Two main categories in regard to content were identified: repetition of parts of the instructions and first perceptual remarks. The first category contained reminders of thinking aloud or repetition of object parts mentioned in the instruction (e.g. box, table, parts). The majority of these utterances included the linguistic marker 'okay, well' signaling that the passive part of receiving instructions was finished and the active part started.
- The major *middle* part directly concerns the assembly process. It contains a *local structure* of sub-processes (also referred to as *episodes*).
- The *end* was defined as utterances following the actual assembly process, expressing completion of the task.

All of the 22 protocols analyzed in this respect so far exhibited this structure. Beginning and end parts were analyzed in regard to content as well as linguistic form; this will not be pursued further here. The analysis of the middle part focuses on the sub-processes of the assembly and their linguistic representation. In the following we will briefly present two aspects of this procedure: first, the content-based derivation of a process model representing the cognitive processes involved in object assembly, and second, a lexical analysis highlighting the mental representations of objects and their functions, and mappings between these, as part of the assembly process.

Process model. Given the explorative nature of the analysis a preliminary process model was derived by a context based analysis of the verbalized actions in a pilot protocol (cf. Someren et al., 1994), drawing on Palmer's (1977) account for working definitions. This model was validated and expanded by the analysis of (so far) 10 further protocols from all three conditions. According to Palmer (1977), problem solving consists of *explorative hypotheses, false leads, dead end, backtracking,* and *fresh starts*. For our current purposes these categories were more specifically defined as follows. *Hypotheses* are ideas and assumptions about objects, moves or consequences of actions. Actions that are evaluated as wrong moves are called *false leads*. *Dead end* states are temporary impasses or states of frustration. *Fresh starts* are instances of disassembly of parts or the whole object and their reassembly in a new way. In addition to utterances expressing these states, some participants also comment on the nature of the task (*meta-level*) or verbalize thoughts that are not directly task related (*aside*). All verbalizations in the middle part of the 11 protocols could be classified as representing one of these categories. Possibly related to the fact that this particular discourse did not serve a communicative intention, some states are not explicitly verbalized. For instance, positive evaluations are seldom stated, but they are implicit in a new hypothesis which shows that the assembly process proceeds. The current version of the process model is shown in Figure 1. The processes that were identified in the think-aloud data are similar to the search-control process described by Newell and Simon (1972). The next step in this analysis procedure will be to spell out the particular linguistic representations used for each of the actions and states. This will provide further insights about their nature and about the patterns of verbalization, which may be useful for computational purposes as explained above.

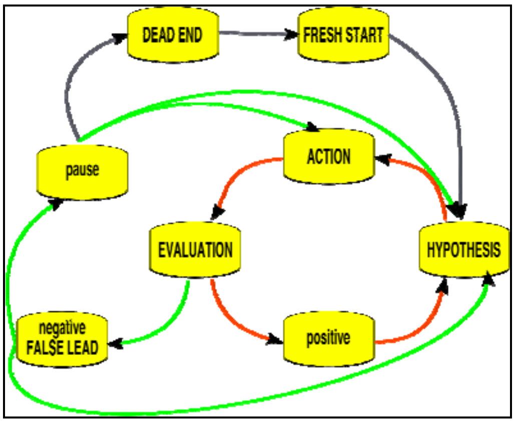


Figure 1. Process model derived from verbal protocol data in object assembly

Lexical analysis. The analysis of the *nouns* used by participants allows for conclusions about the current mental representation of an object part at a particular stage in the assembly process. The nouns can be distinguished in regard to their goal specificity; they can either be *generic*, i.e. not goal object related (e.g. *thing, wood, board*) or *specific* to the goal domain (e.g. *roof, wall, window*). A clear mental representation of the target object should be reflected in a frequent use of goal specific nouns. Participants who were given little or no prior information should therefore use goal specific nouns less regularly, or only later on in the assembly process. Participants who were given a picture of the goal object, in contrast, could draw upon an existing external representation of the object from the start. As an outcome, the distribution of generic and specific nouns should differ systematically between conditions. Our analysis of 22 protocols so far supports this assumption.

While the analysis of the usage of nouns highlights the existence of two levels of representation (generic / specific), the *mapping process* between object parts and functions in the targeted dollhouse is particularly interesting. A *generic* noun (e.g. *thing*) or a *deictic expression* (such as *this (one)*) refers to a particular object in the stock; a *specific* name of a role within the dollhouse (e.g. *wall*) assigns a function to it. These two levels of conceptualization may be connected by *comparison, modal verbs* or *relational verbs*. Altogether, explicit mapping occurs 77 times in the 22 think-aloud protocols analyzed in this respect so far (distributed approximately evenly across individual protocols and conditions, with a slightly higher relative frequency in the *verbal goal* condition as compared to the other conditions). An analysis of the patterns of its occurrence highlights the impact of prior information on mapping processes as follows.

We were particularly interested in the amount of *certainty* concerning the mapping, as this sheds light on the stability of the mental representation of a currently focused object. Linguistic markers expressing high certainty should reflect clearer mental representations on the part of the speaker than linguistic markers expressing neutral or tentative mapping processes and uncertainty. We identified three categories of linguistic representations of mapping processes. First, a high degree of certainty is expressed by the use of relational verbs (present tense of *be*) and a particular set of modals known to signal a high level of certainty (*must*, *will*) (Martin & Rose 2003). Second, another set of modals such as the German equivalents of *may* (*müsste*, *könnte, sollte*) expresses a lower, though still positive level of certainty. The third way in which objects may be assigned functions linguistically is via comparison (such as *(looks) like, (use) as*). Such markers neutrally reflect a tentative assignment of a function to an object. In some cases, hedges such as *a bit* in *this looks a bit like a roof* add an element of uncertainty to the assignment.

According to our analysis of 22 protocols so far, it appears that participants in the *verbal goal* condition assign meaning by using linguistic markers of high certainty most often. These participants were given information about the nature of the target object but not its particular appearance; therefore, they may have had features of typical dollhouses in mind (e.g. open front, walls, roof) and simply matched those to the objects at hand in some suitable way. Mappings via modals expressing a lower degree of certainty were most often used by participants in the *verbal & visual goal* condition. These participants were shown a picture of a correctly assembled dollhouse which they were asked to match. This may have led to a lower degree of certainty if the object parts could not readily be matched to the target picture in memory. Mappings via comparison were most frequent in the *no goal* condition, reflecting the fact that participants were altogether uncertain about the object's functions and tentatively explored mapping options. The analysis of the remaining protocols will shed more light on these issues. However, already at this intermediate stage, a pattern emerges showing that the amount of prior information systematically affects the ways in which object parts are referred to. These results highlight how the cognitive process of assigning functions to previously undefined object parts is linguistically expressed in various ways exhibiting a scale of changing certainty. This systematic variety in linguistic expressions is not necessarily part of the participants' conscious assembly process, but reflects how mental representations change through time, mediated by the amount and nature of prior knowledge.

Conclusion

The linguistically based analysis of verbal protocols enhances the range of insights that can be gained about the cognitive processes involved in complex tasks. In this paper, we have discussed a range of issues concerned with data elicitation, analysis, and interpretation. Two general conclusions can be drawn from this account. On the one hand, diverse types of discourse may be useful for gaining diverse types of insight about thought processes that are externalized in language for diverse purposes. This fact can be made use of for implementation in artificial agents both with respect to computational modelling of thought processes, and in the usage of language for purposes of mediation between different ways of processing (in machines and humans). On the other hand, knowledge about the particular linguistic features involved in texts of any kind may support the analysis of verbal reports effectively, by enabling a focus on those kinds of linguistic items that potentially reflect cognitive processes of interest for scientific progress. While a content-based analysis of language data is suitable for highlighting the conscious processes that study participants verbalize, the structure and linguistic choices involved in these verbalizations contain much more information than one might suspect at first sight. This kind of subtle reflection of cognitive processes becomes informative whenever linguistic evidence exhibits systematic patterns in language use. Particularly if these patterns can be matched to other types of evidence (such as behavioral results, eye movements, and the like), the linguistic data analysis can be trusted as a particularly valuable tool for accessing complex cognitive processes in problem solving tasks.

Acknowledgements

Funding by the Volkswagen Foundation for the Tandem project "Wayfinding Strategies in Behavior and Language", granted to Jan Wiener and Thora Tenbrink, is gratefully acknowledged; also, we thank Universität Bremen for funding the doctoral project "Psychopragmatic analysis of problem solving processes". We thank Katja Abeln for her support in carrying out the assembly study and transcribing the data.

References

Biber, D. 1989. A typology of English texts. *Linguistics* 27:3-43.

Boren, M. and J. Ramey. 2000. Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communication 43*(3), 261–278.

Bransford, John D. and Marcia K. Johnson. 1972. Contextual Prerequisite for Understanding: Some Investigations of Comprehension and Recall. *Jounral of verbal learning and verbal behavior*, 11: 717-726.

Caron, Jean. 1996. Linguistic markers and cognitive operations. In: Caron-Pargue, Josiane and Steven Gillis (eds), 1996. *Verbal Production and Problem Solving*. Antwerp Papers in Linguistics 85, Universiteit Antwerpen, pp. 11-28.

Caron-Pargue, Josiane and Jean Caron. 1991. Psychopragmatics vs. sociopragmatics: the function of pragmatic markers in thinking-aloud protocols. In Jef Verschueren (ed.), *Pragmatics at issue: Selected papers of the International Pragmatics Conference*, Antwerp, August 17-22, 1987, Volume I. Amsterdam: Benjamins, pp. 29-36.

Clark, Herbert H. and Meredyth A. Krych. 2004. Speaking while monitoring addressees for understanding. *Journal of Memory and Language 50*, 62–81.

Dixon, Peter. 1987. The Structure of Mental Plans for Following Directions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(1):18-26.

Denis, Michel. 1997. The description of routes: A cognitive approach to the production of spatial discourse. *Cahiers de Psychologie Cognitive, 16*(4):409-458.

Ericsson, K. Anders, and Simon, Herbert A. 1984. *Protocol analysis –Verbal reports as data*. Cambridge, MA: Bradford books/MIT Press.

Halliday, M.A.K. 1994 (2nd ed). *An introduction to functional grammar*. London: Edward Arnold.

Krahmer, Emiel and Nicole Ummelen. 2004. Thinking About Thinking Aloud: A Comparison of Two Verbal Protocols for Usability Testing. *IEEE Transactions on Professional Communication, Vol. 47, No. 2*, 105-117.

Martin, J.R., & David Rose, 2003. *Working with Discourse: Meaning Beyond the Clause*. Continuum International Publishing Group Ltd.

Newell, Allen and Herbert E. Simon. 1972. *Human Problem Solving*. NJ: Prentice Hall.

Palmer, S.E. 1977. Hierarchical structure in perceptual representation. *Cognitive Psychology 9*(4), 441-474.

Rickheit, Gert and Ipke Wachsmuth (eds.). 2006. *Situated Communication.* Mouton de Gruyter.

Schooler, Jonathan W., Stellan Ohlsson, and Kevin Brooks. 1993. Thoughts Beyond Words: When Language Overshadows Insight. *Journal of Experimental Psychology: General*, 122:2, 166-183.

Someren, W. M., Barnard, F. Y., & A. C. J. Sandberg. 1994. The think aloud method. A practical guide to modeling cognitive process. London: Harcourt Brace & Company.

Tenbrink, Thora. 2008a. The verbalization of cognitive processes: Thinking-aloud data and retrospective reports. In Wiebke Ramm & Cathrine Fabricius-Hansen (eds.), *Linearisation and Segmentation in Discourse. Multidisciplinary Approaches to Discourse 2008 (MAD 08)*, Feb 20-23 2008, Lysebu, Oslo. Oslo: Dept. of Literature, Area Studies and Europ. Languages, Univ. of Oslo, pp. 125-135.

Tenbrink, Thora. 2008b. Wayfinding in an urban environment: A discourse-analytic approach to thinking-aloud data. *Space, Interaction, and Discourse*, Aalborg, 12.-14.11.2008.

Tenbrink, Thora and Inessa Seifert (under review). Conceptual Levels and Strategies in Tour Planning.

Tenbrink, Thora and Jan Wiener. 2009. The verbalization of multiple strategies in a variant of the traveling salesperson problem. *Cognitive Processing 10*:2, 143-161.

Tversky, Barbara, Julie Heiser, Paul Lee, and Marie-Paule Daniel. 2009. Explanations in gesture, diagram, and word. In Kenny Coventry, Thora Tenbrink, and John Bateman (eds), *Spatial Language and Dialogue.* Oxford: Oxford University Press, pp. 119-131.

Wiener, Jan, Thora Tenbrink, Jakob Henschel, and Christoph Hölscher. 2008. Situated and Prospective Path Planning: Route Choice in an Urban Environment. In B. C. Love, K. McRae, & V. M. Sloutsky (eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 851-856). Austin, TX: Cognitive Science Society.

Learning and Recognition of Sketches for Complex Computational Cognition – Position Paper –

Angela Schwering[1] and Ulf Krumnack[2] and Helmar Gust[2] and Kai-Uwe Kühnberger[2]

[1] University of Münster, Institute for Geoinformatics
angela.schwering@uni-muenster.de
[2] University of Osnabrück, Institute of Cognitive Science
{krumnack,gust,kkuehnbe}@uos.de

Abstract. In order to enable machines to operate intelligently in their environment, it is important that they do not only collect sensory input about their environment, but also recognize and understand objects. Analogical reasoning is considered fundamental for many complex cognitive processes. In this paper, we present an experiment which gives empirical support of our hypothesis that object recognition and concept formation rely fundamentally on analogical similarities. Similar object sketches with the same structure are recognized faster and more frequently than similar object sketches with different structure. Afterwards, we introduce our analogy-making framework Heuristic-Driven Theory Projection (HDTP) and explain how HDTP can be used for object recognition.

1 Introduction

In order to enable machines to operate intelligently in our world, it is important that they do not only collect sensory input and observe the environment, but also recognize and understand it. The correct classification of perceived objects allows a machine to use its background knowledge about the world to reason on it. Sketches, i.e. freehand schematized drawings, are an intuitive medium for people to communicate about objects in the world. In this paper, we focus on learning and recognition of sketched objects. We present empirical evidence for our hypothesis that structural similarities are important in the human recognition process. We propose a computational model how machines recognize new sketches by detecting common structures to known sketches and classify the objects according to their ontological knowledge. We examine how concepts change over time and develop an analogy-based approach for learning and revising conceptual knowledge and for explaining the creation of new and abstract knowledge.

Realizing learning and recognition of sketched objects on a machine requires an appropriate language for describing spatial objects in their environments. It must be possible to capture the geometry of all elements in a scene and the spatial relations between them. Furthermore, the representational formalism must be adaptable to change representations of the same scene according to the different perceptions in varying contexts. Recognition requires the ability of comparing new stimuli to already known stimuli in the memory. The structural composition of the object parts is very important, in particular for sketches of spatial objects. Analogical mapping is used to compare two stimuli–a new stimulus and a well-known stimulus–for structural similarities. In a recognition task, the well-known stimulus can be a typical instance of a concept or the specification of a concept from memory.

The model of computational cognition proposed in this paper uses knowledge gained through recognition tasks to learn new and revise old concepts. The two main mechanisms for learning constitute learning via transfer and learning by abstraction [10]. Once a new stimulus is successfully classified, either additional knowledge about the concept can be transferred to the newly classified stimulus, or features observed about the new stimulus can be transferred and integrated in the existing concept description. This additional knowledge leads to a richer and more precise concept description. Moreover, the comparison process aligns analogous elements in both stimuli, i.e. reveals the commonalities of both stimuli at an abstract level. These analogous commonalities describe the essential characteristics defining a concept.

This paper is structured as follows: in Section 2, psychological evidence is provided that structural changes of a visual stimulus do influence object categorization of humans stronger than non-structural changes. Section 3 proposes some ideas for a model of object recognition based on the analogy engine Heuristic-Driven Theory Projection. Section 4 provides a vision how adaptations of representations for analogy-based stimulus recognition can be used for learning new concepts. Section 5 concludes the paper.

2 Object Categorization and Structural Alignment

2.1 The Experiment

A lot of common everyday objects are made up of several, distinct components. The same is true for the kitchen stove depicted by the line drawing in Figure 1. Some components typical for the outward appearance of such a stove have been highlighted in grey color. Obviously, these core elements are spatially related to each other. It is possible to describe these relationships in a qualitative manner. Commonly used spatial relations are topological, directional, or metric relations [1] and may involve other qualities such as symmetry and repetition of elements.

When applying this general idea to the stove in Figure 1, its highlighted components might be regarded as separate regions with certain underlying topological relations. The four hotplates on top could be regarded as four disjoint regions all of which are in turn situated inside Area 1. Underneath, Area 2 contains six disjoint temperature regulators. Similar relationships can be found as to the front handle and the spy window both of which are disjoint and situated

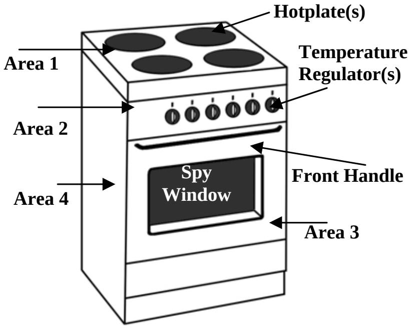


Fig. 1. Line drawing of a typical kitchen stove.

within another area (Area 3) on the stove's foreside. Furthermore, the lateral Area 4 directly meets Area 2, and so forth.

To investigate the role of structured representation in human object recognition, an experiment was set up, in which subjects had to recognize line drawings of different objects.[3] 132 line drawings were selected for the experiment. Of these, 72 functioned as filler items, whereas the remaining 60 drawings acted as the so-called "basic" experimental stimuli. The latter served as a basis for the development of four additional variations, namely two versions of non-structural modifications and two versions of structural modifications (cf. Figure 2). Generally speaking, each experimental condition was conceptualized as a pair of two experimental stimuli, henceforward referred to as item pairs.

Basically, a single experimental trial was composed of a source image stimulus and a subsequent target image stimulus. First, the source stimulus was shown and all subjects were expected to name the object that they thought to have identified in the black and white line drawing by an oral answer. Then, subjects had to press the keyboard's down-arrow key to call up the target image. In preparation for the imminent stimulus, a fixation cross with a duration of 250 ms was shown in the middle of the monitor prior to the occurrence of the target image. Finally, the target image stimulus appeared for maximally 650 ms. This time, the subjects' task consisted in deciding as quickly as possible by pressing the "yes" or "no" button whether the object they were just seeing was an instance of the same concept as the object they had named in the step before.

Due to the five experimental conditions, we created equally many stimulus lists that counterbalanced item pairs and conditions. Each subject saw 36 filler item pairs, 12 MAT items pairs, 12 NS1 item pairs, 12 NS2 item pairs, 12 S1 item pairs, and 12 S2 item pairs yielding 96 experimental trials in total. Figure 2 specifies the modified versions of the original stimulus.[4]

[3] The interested reader is referred to [20] for a complete presentation of the experiments.

[4] 75 native German subjects, 50 females and 25 males, volunteered for the experiment and confirmed normal or corrected normal vision. The vast majority of participants

MAT: The match condition was conceptualized as an item pair with identical source and target images. Solely the 60 basic experimental stimuli served as basis to set up this condition. Furthermore, this condition served as a baseline with respect to the reaction time measurements and required a clear "yes" response from the subjects.

NS1: This condition entailed the movement of significant picture elements. These manipulations were not taken for a structural change since it was made sure that the topological relationships between the manipulated and unaffected picture elements remained untouched. It was anticipated that the subjects would show a high tendency to give a "yes" response.

NS2: This condition entailed the resize of picture elements without moving them to another position. Simple resize was not taken for a structural change as long as the topological relations between the manipulated and other picture elements remained constant. It was anticipated that the subjects would show a high tendency to give a "yes" response.

S1: As for the first structural condition, it exclusively implicated the removal and/or addition of selected picture elements. Adding to or removing significant elements from the overall scene was regarded as a clear structural change. It was decided to accept both a "yes" and a "no" response as "potentially correct".

S2: The second structural condition likewise implied the movement of significant picture elements as with condition NS1. However, this time a structural change was deliberately caused by moving selected elements into another area. Alternatively, this condition involved the resize of desired picture elements as with condition NS2. Both "yes" and "no" were accepted as potentially correct answers.

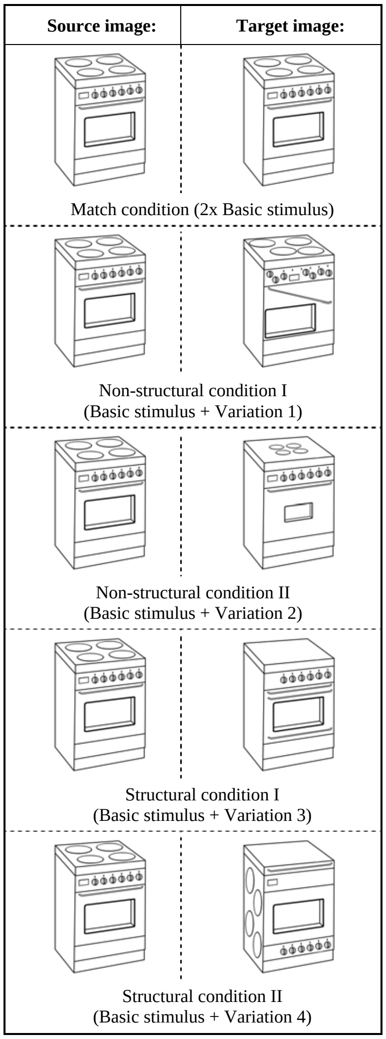

Source image:
Target image:
Match condition (2x Basic stimulus)
Non-structural condition I
(Basic stimulus + Variation 1)
Non-structural condition II
(Basic stimulus + Variation 2)
Structural condition I
(Basic stimulus + Variation 3)
Structural condition II
(Basic stimulus + Variation 4)

Fig. 2. The types of stimuli used in the experiment: Match condition, non-structural condition I (NS1), non-structural condition II (NS2), structural condition I (S1), and structural condition II (S2).

2.2 Results

For the goals of this paper, it suffices to find evidence for the assumption that humans would need more time to recognize structurally manipulated objects compared to non-structurally manipulated objects. As a consequence, it was decided to combine both non-structural (NS1 & NS2) as well as the two structural

consisted of undergraduate students who were enrolled in Psychology or Cognitive Science at the University of Osnabrück. The mean age was 23.2 years, ranging from age 18 to age 58. The experiment was conceptualized and generated with the aid of the software suite E-Prime 2.0 by Psychology Software Tools Inc.

Condition	RT in ms (Std. Dev)	ACC in %	Yes / No Ratio in %
MAT	618 (147)	95.6	—
NSCOM	708 (182)	—	82.1 / 17.9
SCOM	752 (200)	—	61.3 / 38.7

Table 1. Descriptive statistics results - analyses by subjects (”Yes” and ”No” responses).

conditions (S1 & S2), essentially because of their strong relatedness.[5] The relevant reaction times per subject were summed up and averaged afterwards. The same holds for the “yes”/“no” response ratios yielding the numbers shown in Table 1.

On that basis, a 1 (source image) × 3 (target image type: MAT, NSCOM, SCOM) factorial analysis of variance (ANOVA) including repeated measures was conducted on the response latencies by subjects and by items. Only data points that were maximally two standard deviations away from their corresponding mean were taken into account to reduce the quantity of outliers in the first place. A confidence interval of 95% was consistently used.

As a result, the main effect for target image type was highly significant in the analysis by subjects (F1) and by items (F2) with $F1(1.61, 112.56) = 87.51$, $p < .001$ (Huynh-Feldt corrected); $F2(2, 110) = 69.15$, $p < .001$. Concerning the main effect for list, it was only significant in the analysis by items, $F1(4, 70) = .52$, $p > .72$; $F2(4, 55) = 7.50$, $p < .001$. By contrast, the two-way interaction between list and target image type was not significant at all with $F1(8, 138) = 1.21$, $p > .30$; $F2(8, 108) = 2.00$, $p > .05$.

Several pairwise comparisons (MAT vs. NSCOM; MAT vs. SCOM; NSCOM vs. SCOM) were carried out. In all pairwise comparisons, the main effect for target image types was highly significant in the analysis by subjects and by items. As an example the results for NSCOM vs. SCOM are mentioned[6]. The main effect for target image type was highly significant by subjects and by items with $F1(1, 70) = 34.82$, $p < .001$; $F2(1, 55) = 15.90$, $p < .001$. The main effect for list was only significant in the analysis by items, $F1(4, 70) = .41$, $p > .80$; $F2(4, 55) = 3.40$, $p < .05$. The two-way interaction between list and target image type was not significant ($F1(4, 70) = 1.52$, $p > .21$; $F2(4, 55) = 1.16$, $p > .34$).

2.3 Discussion

The experiment provides two results that are relevant for the discussion in this paper. First, the relation of “yes”/“no” responses shows that the degree of recognition is significantly higher if the structure of the visual stimulus is not changed (NSCOM), compared to the cases where it is changed (SCOM). This indicates that subjects are more willing to accept an object as belonging to a category, if its relational structure stays intact. Second, the reaction time is shorter in these

[5] A detailed presentation of the results with separate treatment of all conditions can be found in [20].

[6] The complete results can be found in [20].

cases, indicating that the task is cognitively less complex if a structural match of stimuli can be found.

Both results back the claim, that object recognition seems to be based, at least partly, on matching structural representations of the provided stimuli. A cognitive plausible model of object recognition should therefore incorporate such representations and matching mechanisms. In the rest of the paper, we sketch a model for recognizing visual stimuli that is driven by analogical mapping and that furthermore allows to introduce a learning mechanism based on recognition.

3 Analogy-Based Recognition of Visual Stimuli

The model we propose is based on Heuristic-Driven Theory Projection (HDTP), a formal framework to compute analogies. This section gives a brief introduction to analogies and HDTP focussing on those aspects relevant to the intended application. A more comprehensive description of HDTP can be found in [19].

3.1 Syntactic Basis of HDTP

Classically, an analogy is established between two domains of knowledge, called *source* and *target* domain. By discovering corresponding structures in both domains, an analogical relation can be constructed. Such a relation can be used to identify commonalites and differences between the domains. Furthermore, gaps discovered in one domain can be filled by transferring knowledge from the other domain, based on the analogical relation. Such analogical inferences, though possibly incorrect from a logical point of view, can be a basis to explain certain aspects of cognitive phenomena like creativity and learning.

HDTP provides a formal framework to compute analogical relations and inferences, for domains represented in first-order logic. Both, source and target domain, are given by axiomatizations, i.e. finite sets of first-order formulae. The basic idea is to associate pairs of formulae from the domains in a systematic way. HDTP uses anti-unification to identify common patterns in formulae. In anti-unification, two formulae are compared and the most specific generalization subsuming both formulae is identified. As a result, besides the generalized formula a pair of substitutions is computed, that expresses the analogical relation between the two formulae.

This process of generalization by anti-unification can be iteratively applied to formulae of the two axiomatizations. However, it might be the case that for some axiom no good corresponding axiom exists on the other side. Nevertheless, there might still exist a good formula in the theory spanned by the axiomatization, i.e. among the formulae that can be derived from the axioms. In this case, HDTP will try to prove such a formula. This process can be considered as a kind of re-representation [11], since the originally given axiomatization is adapted to match the needs of the analogy considered. As a consequence HDTP does not compute an analogy between two specific axiomatizations, but between the theories spanned by these axiomatizations.

$line(a, (1, 1), (1, 4))$
$line(b, (1, 1), (4, 1))$
$line(c, (1, 4), (4, 4)) \ldots$
$round(m, (2, 5), (3, 5), 0.5)$
$round(n, (4, 5), (5, 5), 0.5) \ldots$

$polygon(p1, [a, b, c, d])$
$polygon(p2, [d, e, f, g])$
$above(p2, p1)$
$group(g2, [m, n, o, p])$
$inside(g1, p2)...$

Fig. 3. Representation of a stove with its primitive elements in an unstructured way (left) and in a structured way (right).

HDTP distinguishes between domain knowledge (facts and laws holding for the source or the target domain) and background knowledge, which is true across domains. The background knowledge is of special importance in the context of re-representation, as it may be used to derive further formulae in the two domains, which then can be used again for generalization.

Uncovered parts of the source and the target domain, i.e. formulae that are not part of the analogical relation and therefore cannot be derived from the generalized formulae, are candidates for analogical transfer. The established analogical relation is used to translate these formulae. If the result does not lead to a contradiction in the other domain, it can be considered as an analogical inference, i.e. new knowledge that might be added to the axiomatization of that domain.

3.2 A Formal Language to Represent Spatial Objects

We now apply the ideas of HDTP to the processing and recognition of visual stimuli. In this setting, source and target are both from the same domain, i.e. sketch drawings. We distinguish between flat and structured representations of visual stimuli. A flat representation covers all features of a stimulus without any relational structure between them (e.g. the left side of Figure 3 listing the primitive visual elements of the stove). A structured representation captures regularities of a stimulus, like symmetry, iterations, Gestalt groupings etc. It furthermore comprises geometrical and topological relations. The structured representation on the right side of Figure 3 replaces the lines by a description of closed shapes such as polygons. Although the flat and the structured representation contain the same information, the structured representation is closer to the way humans perceive the visual stimuli. Our computational model of cognition shall take a flat representation as input and automatically compute a structured representation of the sketch reflecting human perception. A structured representation can be build from a flat representation according to a certain set of rules.

The application of HDTP as a framework for object recognition requires the development of a suitable language to represent spatial objects, the ability to adapt these representations such that analogous structures between the source and the target object become visible, and finally a mechanism for analogy-based learning of concepts. As a consequence the language has to meet two major requirements: it must describe all elements in a spatial scene with respect to the aspects relevant in human perception, but it must describe as well the spatial relationships which are important to compare and recognize objects. To reflect human perception, the language must comprise significant perceptual features, but also vocabulary to specify visual structures. When the human visual sensory system observes a spatial object, it transforms the unstructured information into a structured representation of coherent shapes and patterns. Human perception tends to follow a set of Gestalt principles: stimuli are experienced as a possibly good Gestalt, i.e. as regular, simplistic, ordered, and symmetrical as possible. Therefore the language focuses on basic Gestalt principles of perception, i.e. it allows for groupings according to the principle of similarity, the principle of proximity, closure, and good continuation.

The second requirement refers to spatial features: the geometry of elements in a scene and their spatial relations have to be represented in a way that allows for cognitively plausible reasoning. Common calculi for qualitative spatial reasoning such as RCC 8 for topological relations [14] and TPCC calculus [12] or neighborhood-based approaches [6, 15] for directional relations are integrated in the formal language.

In [17], we developed first steps towards a language for representing simple figures in geometric proportional analogies. Figure 3 shows exemplary a formal language representing a stove. On the left is an unstructured representation of the stove listing its primitive elements (lines and round elements). On the right is a structured representation of a stove: The four connected lines are represented as closed polygon. The four hotplates are grouped together according to the Gestalt principle of similarity and proximity. The topological relation inside and the directional relation above are captured as well. The groups of hotplates are inside the polygon $p2$ and polygon $p2$ is above polygon $p1$. In the following section, we explain how HDTP automatically adapts the unstructured representation to form a structured one.

3.3 Adaptation of the Representation for Analogy-Based Stimulus Recognition

The cognition of spatial objects involves the construction of a consistent and meaningful overall picture of the environment. Gestalt Psychology argues that human perception is holistic: instead of collecting every single element of a spatial object and afterwards composing all parts to one integrated picture, we experience things as an integral, meaningful whole. The whole contains an internal structure described by relationships between the individual elements.

In HDTP, a visual stimulus is described via a set of axioms specifying the features of all elements at a basic level (Figure 4). A set of perception rules and rules for spatial reasoning form the background knowledge of the system. The set of all formulae that can be inferred from the axioms comprises all possible re-representations of the same visual stimulus, but at different structural levels.

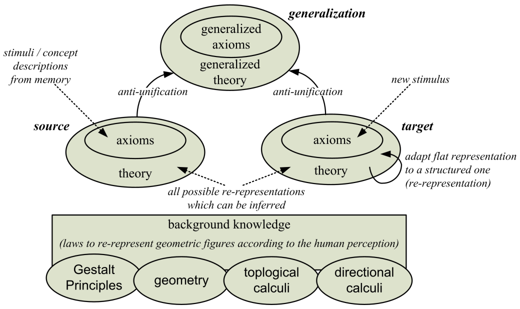

generalization
generalized axioms
generalized theory
stimuli / concept descriptions from memory
anti-unification
anti-unification
new stimulus
source
axioms
theory
target
axioms
theory
adapt flat representation to a structured one (re-representation)
all possible re-representations which can be inferred
background knowledge
(laws to re-represent geometric figures according to the human perception)
Gestalt Principles
geometry
toplogical calculi
directional calculi

Fig. 4. Analogical comparison in the logical framework HDTP.

The initially flat representation can be transformed into a structured one by means of logical inference.

In the recognition task, a new stimulus (target) is compared to a known stimulus (source). The source stimulus is described via a structured representation recalled from the memory or knowledge base. The structural commonality between the flat representation of the target and the structured representation of the source is initially not obvious. To successfully classify a new stimulus, a mapping between the target stimulus and the source stimulus must be established, i.e. an analogous structure has to be established on the target stimulus. During the analogy-based mapping process the target must be re-represented such that common structures become visible. The re-representation process building a structure on the target side can be driven by heuristics motivated by human perception, like Gestalt principles.

Figure 5 shows adaptation rules as they can be found in the background knowledge: The first rule is applied to detect closed shapes such as a polygon and the second one is applied to compute topological relations such as inside. The re-representation process is driven by heuristics based on properties of human perception and by building a structure on the target side analogously to the structured stimulus on the source side. Experimental data shall give the necessary insight for creating appropriate heuristics reflecting human strategies in spatial object recognition. The heuristics have a great influence on the efficiency of the whole computational approach.

4 Analogy-based Learning, Concept Formation, and Creativity

Similarity judgment is one of the most central constructs in cognitive processes. Organization of conceptual knowledge in memory, recognition of new stimuli, and learning hinge crucially on similarity comparisons [8]. In particular, the role of structural similarity in relational categories has been considered as important [7]. We argue that structural similarity as detected in analogies is particularly important for learning spatial concepts. Our approach for computational cognition

Closed Shape (adapted from Gestalt principle)

```
lineConnection(A, B) :- line(A, (_, _), (X, Y)), line(B, (X, Y), (_, _)).
lineConnection(A, B) :- line(A, (X, Y), (_, _)), line(B, (X, Y), (_, _)).
polygon(P, [. . .]) :- . . .
```

Topological Relation proper part (adapted from RCC8)

```
regionConnection(X, Y) :- region(X), region(Y), not(disjoint(X, Y)).
part(X, Y) :- not(regionConnection(Z, X), not(regionConnection(Z, Y))).
properPart(X, Y) :- part(X, Y), not(part(Y, X)).
overlap(X, Y) :- part(Z, X), part(Z, Y).
```

Fig. 5. Adaptation rules are stored in the background knowledge and define how unstructured descriptions can be re-represented to structured ones.

shall learn to classify spatial objects, i.e. the system shall be able to revise and refine its ontological knowledge during a training phase. Although researchers agree that analogy-making is central for human learning, there does not exist a comprehensive theory for analogical learning. Our own first ideas for a learning model based on HDTP were outlined in [18].

HDTP supports learning at two levels: analogical transfer and abstraction. Learning via analogical transfer means gaining new knowledge by applying additional knowledge about the source to the target. The system transfers knowledge about the concept (e.g. knowledge about the functionality) and applies this to the new stimulus. This enables the system to draw new inferences on the target side. Transfer also happens from the target to the source: the system observes characteristics about the new stimulus which leads to a revised concept definition. Learning via abstraction refers to the generalization process that is essential to derive abstract concept definitions. Existing approaches apply classical inductive learning which requires large set of data samples to create general laws. However, humans can generalize already over a small set of samples. Applying analogical comparison and describing structural commonalities at a general level is one possible way to make the essence defining a concept apparent. Reflecting this analogical generalization process is one of the strengths of HDTP [16]: during the analogical mapping, anti-unification automatically constructs a generalization for every aligned pair of formulae. This way, HDTP creates an explicit generalized theory over two theories – the source and the target theory. We argue that this generalized theory captures exactly the essential commonalities of the instances of a concept at an abstract level and therefore is an ideal mechanism for extracting the defining elements of a concept.

The following example illustrates how HDPT functions in concept formation and concept learning (cf. Figure 6). HDTP has a structural description of a stove in its knowledge repository. Presenting a new stove in a recognition task, HDTP detects the analogous structure and constructs a generalization containing the commonalities (i.e. common aspects about the geometry and spatial relation such as the temperature regulators being situtated in the front polygon). The generalization represents the concept "stove" at an abstract level. If again a new stove is presented in a second recognition task (e.g. the third one in the above figure), it could be classified as a stove, however the new generalization is not so

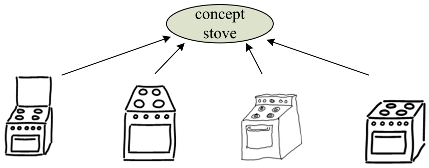


Fig. 6. A structural comparison of these stoves reveal that all stoves sketches have the form of a 3D cube with four hotplates on top and a spy window at the front. Three sketches show stoves with temperature regulators at the front.

specific on the position of the temperature regulators. First steps towards this incremental analogy-based learning have been sketched in [9].

5 Conclusions and Critical Evaluation

Analogies play a major role for cognition. We have shown empirically, that structural commonalities are important in object comparison and recognition: In a recognition task, subjects have recognized sketches of non-structurally varied objects faster and easier than sketches of objects which were structurally varied.

We have suggested an approach using HDTP, a symbolic analogy-making framework, to compute analogies between sketches of objects. HDTP is a promising framework, because it supports adaptation and learning at an abstract level. Many times analogical structures are not visible per se, but result from a comparison and mapping task. HDTP combines the representation of basic elements in a sketch with background knowledge on human perception. Therefore, HDTP can reveal commonalities in different contexts and different perceptions. It re-represents an unstructured flat representation of a sketch and determines a structured representation of the target stimulus which possibly matches the structured representation of the source stimulus. Furthermore, HDTP compares structures of source and target stimuli and computes a generalization of the shared structures. This supports concept learning.

Lately, various approaches have been developed to describe visual stimuli and detect analogous structures. *CogSketch* (comprising GeoRep and nuSketch) [4, 5] is a powerful tool for sketch understanding. A sketch consists of glyphs, which are the primitive elements. The spatial structure of the overall sketch is analyzed by topological, metric and directional relations between glyphs. A glyph is a piece of ink, i.e. a glyph can be a simple point but also a complex drawing. The approach proposed in this paper considers primitive elements as the most basic entity in a sketch, which itself can be re-represented as more complex figures by re-representation rules such as the ones depicted in Figure 5. The *Languages of Perception* [2] developed for Indurkhya's algebraic Interactionist Theory has a similar idea of re-representing simple geometric elements. It incorporates Gestalt-motivated mechanisms for re-representation such as groupings and iterations. The approach presented here goes beyond the Languages of Perception: We also aim at the explicit description of spatial relations and the integration of existing qualitative spatial reasoners. *Galatea* and the Proteus analogy model [3] was developed to describe visualizations in the context of problem solving. It aims at detecting visual similarities and transferring problem solving solutions, but not at the re-representation for perceptual understanding of sketches.

References

1. A. Cohn and J. Renz. Qualitative spatial representation and reasoning. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.
2. M. Dastani and R. Scha. Languages for Gestalts of line patterns. *Journal of Mathematical Psychology*, 47:429–449, 2003.
3. J. Davies and A. K. Goel. Representation issues in visual analogy. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, pages 300–305, 2003.
4. R. W. Ferguson and K. D. Forbus. GeoRep: A flexible tool for spatial representations of line drawings. In *18th National Conference on Artificial Intelligence*, Austin, Texas, 2000. AAAI Press.
5. K. D. Forbus, K. Lockwood, M. Klenk, E. Tomai, and J. Usher. Open-domain sketch understanding: The nuSketch approach. In *AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural*, Washington, DC, 2004.
6. C. Freksa. Using orientation information for qualitative spatial reasoning. In A. U. Frank, I. Campari, and U. Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, volume 639 of *Lecture Notes in Computer Science*, pages 162–178, Pisa, Italy, 1992. Springer.
7. D. Gentner and K. J. Kurtz. Relational categories. In W.-K. Ahn, R. L. Goldstone, B. C. Love, A. B. Markman, and P. W. Wolff, editors, *Categorization inside and outside the laboratory*, pages 151–175. APA, Washington DC, 2005.
8. R. L. Goldstone and A. Kersten. Concepts and categorization. In A. F. Healy and R. W. Proctor, editors, *Comprehensive Handbook of Psychology*, volume 4, pages 599–621. Wiley, New Jersey, 2003.
9. H. Gust, U. Krumnack, K.-U. Kühnberger, and A. Schwering. Integrating analogical and inductive learning at different levels of generalization. In *Workshop on Learning from Non-Vectorial Data (LNVD2007) at the 30th Annual German Conference on Artificial Intelligence (KI07)*, Publications of the Institute of Cognitive Science (PICS), Osnabrück, 2007. University of Osnabrück.
10. H. Gust, U. Krumnack, K.-U. Kühnberger, and A. Schwering. Analogical reasoning: A core of cognition. *KI - Zeitschrift für Künstliche Intelligenz, Sonderausgabe KI und Kognition*, pages 8–12, 2008.
11. U. Krumnack, H. Gust, K.-U. Kühnberger, and A. Schwering. Re-representation in a logic-based model for analogy making. In *AI 2008: Advances in Artificial Intelligence*, volume 5360/2008 of *Lecture Notes in Computer Science*, pages 42–48. Springer, 2008.
12. R. Moratz, B. Nebel, and C. Freksa. Qualitative spatial reasoning about relative position. In *Spatial Cognition III*, volume 2685 of *Lecture Notes of Computer Science*, pages 385–400. Springer, 2003.
13. E.-C. Mullally and D. P. O'Donoghue. Spatial inference with geometric proportional analogies. *Artificial Intelligence Review*, 26(1-2):129–140, 2006.
14. D. A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In B. Nebel, W. Swartout, and C. Rich, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference*, pages 165–176, Los Alloss, CA, USA, 1992. Morgan Kaufmann.
15. R. Rauh, C. Hagen, M. Knauff, T. Kuss, C. Schlieder, and G. Strube. Preferred and alternative mental models in spatial reasoning. *Spatial Cognition and Computation*, 5(2&3):239–269, 2005.
16. A. Schwering, U. Krumnack, H. Gust, and K.-U. Kühnberger. Abstraction in analogy. In *8. Fachtagung der Gesellschaft für Kognitionswissenschaft (KogWis2007)*, Saarbrücken, Germany, 2007.
17. A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust. Using Gestalt principles to compute analogies of geometric figures. In D. S. McNamara and J. G. Trafton, editors, *29th Annual Conference of the Cognitive Science Society (CogSci07)*, pages 1485–1490, Austin: TX, 2007.
18. A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust. Analogy as integrating framework for human-level reasoning. In *1st Conference on Artificial General Intelligence (AGI08)*, pages 419–423, Memphis, TN, 2008. IOS Press.
19. A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust. Syntactic principles of heuristic-driven theory projection. *Journal of Cognitive Systems Research*, 2009. Special Issue on Analogies - Integrating Cognitive Abilities.
20. R. Stollinski, A. Schwering, K.-U. Kühnberger, and U. Krumnack. Structural alignment of visual stimuli influences human object recognition. In *Proc. of Analogy'09*. NBU Press, 2009.

Neural Networks and Continuous Time

Frieder Stolzenburg and Florian Ruh

Hochschule Harz (FH), University of Applied Sciences, Department of Automation and Computer Sciences, D-38855 Wernigerode, Germany, {fstolzenburg,fruh}@hs-harz.de

Abstract. The fields of neural computation and artificial neural networks have developed much in the last decades. Most of the works in these fields focus on implementing and/or learning discrete functions or behavior. However, technical, physical, and also cognitive processes evolve continuously in time. This cannot be described directly with standard architectures of artificial neural networks such as multi-layer feed-forward perceptrons. Therefore, in this paper, we will argue that neural networks modeling continuous time explicitly are needed for this purpose, because with them the synthesis and analysis of continuous and possibly periodic processes in time are possible (e. g. for robot behavior) besides computing discrete classification functions (e. g. logical boolean functions). We will relate possible neural network architectures with (hybrid) automata models that allow to express continuous processes.

Key words: neural networks; physical, technical, and cognitive processes; hybrid automata; continuous time modeling.

1 Introduction

During the last decades, the field of (artificial) *neural networks* has drawn more and more attention due to the progress in software engineering with artificial intelligence. Neural networks have been applied successfully e. g. to speech recognition, image analysis, and in order to construct software agents or autonomous robots. A basic model in the field is a multi-layer feed-forward perceptron. It can be automatically trained to solve complex classification and other tasks, e. g. by the well-known backpropagation algorithm (cf. [4, 15]). Implementing and/or learning discrete functions or behavior is in the focus of neural networks research.

Nevertheless, technical, physical, and also cognitive processes evolve in time continuously, especially if several agents are involved. In general, modeling multiagent systems means to cope with constraints that evolve according to the continuous dynamics of the environment. This is often simulated by the use of discrete time steps. In the literature, *hybrid automata* are considered for the description of systems by a mathematical model, where computational processes interact with physical processes. Their behavior consists of discrete state transitions plus continuous evolution [5]. Hybrid automata have been successfully applied especially to technical and embedded systems, e. g. for describing multi-robot behavior [2, 14]. However, a feasible procedure for learning hybrid automata does not seem to be available.

Therefore, we will at first introduce application scenarios that include complex cognitive, technical, or physical processes for the synthesis and analysis of continuous and possibly periodic systems of agent behavior (Sect. 2). After that, we briefly discuss some related works on neural networks and hybrid automata wrt. their applicability to timely continuous systems (Sect. 3). Then, we present an enhanced model of neural networks with continuous time, which we call *continuous-time neural network* (CNN) (Sect. 4), which can simulate the behavior of hybrid automata as a system that interprets periodic, continuous input and the response to that. It can also be used for periodicity detection, e. g. in speech or musical cognition. Finally, we will end up with conclusions (Sect. 5).

2 Scenarios of Agents in a Continuously Evolving Environment

Scenario 1 (deductive reasoning). *Classification tasks like e. g. image recognition or playing board games (see Fig. 1) require deductive reasoning and cognition. In this scenario, the environment is discrete (according to the classification in [15]), because there is only a limited number of distinct percepts and actions. In particular, it is not dynamic, i. e., the environment does not change over time, while the agent is deliberating.*

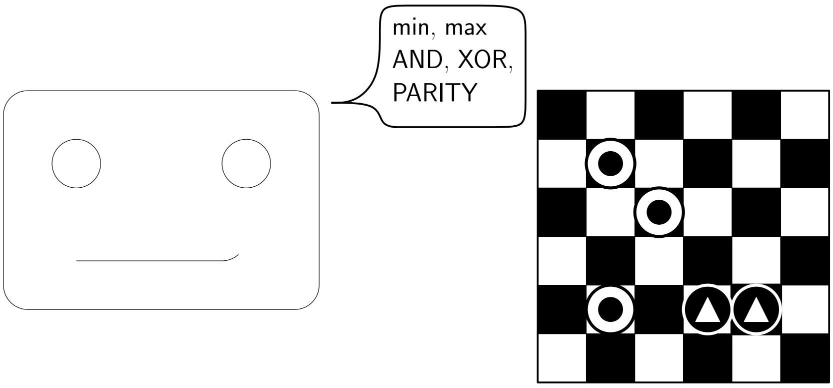


Fig. 1: Agent reasoning deductively.

Ordinary artificial neural networks allow to solve classification tasks and to express logical boolean functions for deductive reasoning directly, i. e. functions of the form $f : X \to Y$, where $X = (x_1, \dots, x_n)$ represents the input values and $Y = (y_1, \dots, y_m)$ the output values. Therefore, deductive reasoning can be adequately implemented by using them. Neural networks in general consist of an interconnected group of nodes, called units, which are programming constructs mimicking the properties of biological neurons. Standard neural networks such as multi-layer feed-forward perceptrons have a restricted architecture. There, we have only three or more layers of units: input, hidden, and output units, which are connected only in this order [4, 15]. It is well-known [4] that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer, if we have sigmoidal activation functions, i. e. bounded, nonlinear, and monotonously increasing functions, e. g. the logistic function or the hyperbolic tangent (tanh). Multi-layer networks use a variety of learning techniques, the most popular being backpropagation. In general, any declarative logical operation can be learned by such a network. However, many real cognitive or physical processes depend on time, as in the following scenario.

Scenario 2 (robot at a conveyor belt). *Let us consider a robot that has to perform a specific routine again and again, e. g. grabbing a brick from a conveyor belt (see Fig. 2). Fig. 3 shows the height h of the robot arm depending on the time t. For the ease of presentation, we abstract from releasing the box, moving the arm down and grabbing the next one here. In addition, we assume, that the agent knows the duration T of each episode.*

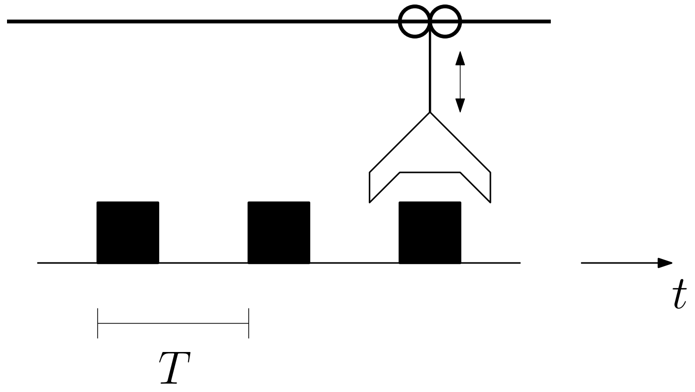


Fig. 2: An example robot arm, picking boxes on a conveyor belt.

This scenario requires the solution of several tasks. In particular, continuous behavior of the robot agent must be producible for grabbing the bricks continuously and periodically. Clearly, for synthesis and also for analysis of processes or behavior, modeling the time t explicitly is necessary, because we have to model mappings of the form $X(t) \mapsto Y(t)$. For Scenario 2, we assume that the robot has to move its arm up and down within a fixed time interval T. This leads to a sawtooth function, if we consider the dependency from time (see Fig. 3). Such behavior can be expressed easily by an automaton model, especially hybrid automata [5] (see Sect. 3). However, the procedure with hybrid automata mainly is a knowledge-based approach. They cannot be learned easily by examples as e. g. neural networks.

While clearly Scenario 1 can be specified directly with ordinary neural networks, Scenario 2 requires to model the time t somehow. This can be achieved by discretizing time, i. e. by considering input values at different discrete time points, $t, t-1, \ldots, t-T$ for some time horizon T. Then, we may use $x_i(t), x_i(t-1), \ldots, x_i(t-T)$ with $1 \leq i \leq n$

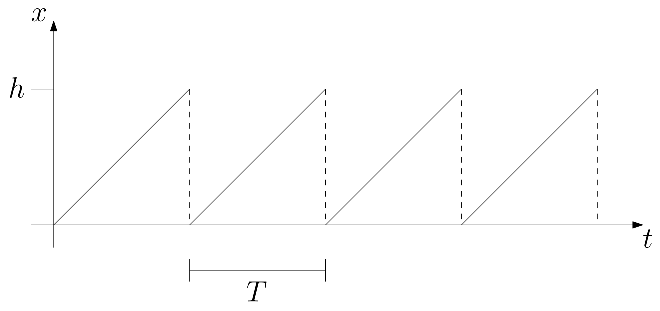


Fig. 3: The sawtooth function for the height of the robot arm, assuming that it can lower the arm in zero time.

as input values. But this procedure has several disadvantages: It increases the number of input units significantly, namely from only n to $(k+1) \cdot n$. In addition, it is not clear in this case, what granularity and past horizon of discrete time should be used.

Therefore, a presentation by (enhanced) neural networks seems to be a good idea, that makes use of the (continuous) time t as additional parameter, at least implicitly. Furthermore, oscillating periodic behavior must be producible, even if the input X remains static, i. e. constant. For instance, once switching on a robot, i. e. change one input unit from 0 to 1, the periodic behavior should hold on, until the input unit is switched off again. Therefore, we will introduce units, whose input may be a fixed value, but whose output yields a sinusoid (see Sect. 4, Def. 2). By this, we can express periodic behavior in time by neural networks. Furthermore, we should be able to analyze behavior and to detect period lengths, which we formulate now:

Scenario 3 (behavior and periodicity analysis). *Before a robot is able to behave adequately in a dynamic environment, it has to analyze its environment, e. g. to find out the duration of an episode of the robot at the conveyor belt (Scenario 2, Fig. 2), i. e. the period length in time. This task also appears in speech and musical harmony recognition, as illustrated in Fig. 4.*

Since cognitive science may be defined as the study of the nature of intelligence and thus of intelligent behavior, drawing on multiple disciplines, including psychology, computer science, linguistics, and biology, we consider behavior and periodicity analysis here, because it is obviously an important aspect of intelligence. In particular, this holds for scenarios with several agents and/or agents in dynamically changing environments, because it is the basis for coordination and synchronization of (periodic) behavior of agents. For instance, finding the way through a dynamic environment with many obstacles and crossing traffic of a specific frequency, requires synchronization among agents, including periodicity analysis.

One possible key for determining overall period lengths is auto-correlation, i. e. the cross-correlation of a signal with itself. It can be mathematically defined by convolution (cf. [1], see also Sect. 3.3). However, we choose another formalization here: We simply assume that a unit of a CNN (cf. Def. 2) can delay its incoming signals for a specific

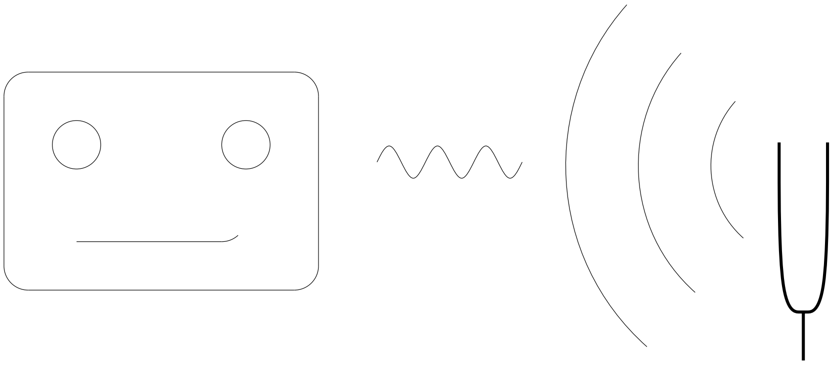


Fig. 4: Agent analyzing periodic episodes in the environment.

time delay δ. Then, a comparison of the original signal with the delayed one yields the appropriate result. Eventually, biological neural networks, e. g. the hearing system in the brain, seem to be able to delay signals [8]. Before we present the CNN model in more detail (Sect. 4), let us first discuss related works that are more or less suitable for modeling the scenarios introduced here.

3 Neural Networks, Hybrid Automata, and Continuous Time

The underlying idea that the original model of artificial neural networks tries to capture is that the response function of a neuron is a weighted sum of its inputs, filtered through a nonlinear, in most cases sigmoidal function

$$y = h(\sum_{i=1}^{n} w_i x_i)$$

where h is the activation function, e. g. the logistic function $(\frac{1}{1-\exp(-x)})$. Fig. 5 shows the general scheme of a unit of a neural network with the inputs $x_1, \ldots, x_n$ and one output y. Each incoming and also the outgoing edge is annotated with a weight w_i.

3.1 Fourier Neural Networks

An obvious paradigm to combine neural networks with periodic input are so-called *Fourier neural networks* [11, 16]. They allow a more realistic representation of the environment by considering input oscillation for implementing and/or learning discrete functions or behavior. From a neurophysiological point of view, they appear to be closer to reality, because they model the signals exchanged between neurons as oscillations, making the model to better agree with discoveries made in neurobiology. In [16], the output function of a neuron is defined as $f(X) = \int_D c(X)\,\varphi(X,Y)\,dY$, where $\varphi(X,Y)$ is some characteristics of the input X, weighted by the coefficients $c(X)$, i. e., we get a weighted integral (replacing the sum from above) of the inputs and their characteristics. However for the computation, a discretized model given by the equation $f^d(x) = h\left(\sum_i c_i \Pi_{j=1}^n \cos(\omega_{ij}x_j + \varphi_{ij})\right)$ is used with the sigmoidal logistic function h from above in order to obtain output in the interval $[0;1]$.

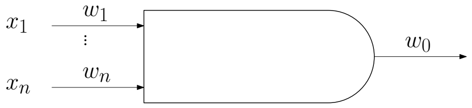


Fig. 5: A unit of a neural network (scheme).

In [11], Fourier neural networks with sinusoidal activation function $h(x) = c\sin(ax+b)$ are considered. Additional non-linear (sigmoidal) activation functions are not needed to express arbitrary functions in this case. In fact, the sine function has the characteristics of a sigmoid function in the interval $[-\pi;\pi]$. All logical operators with two inputs (Scenario 1) can be implemented in this framework (see Fig. 6) by only *one* single unit with sinusoidal activation function, in contrast to the standard neural networks with other, monotonously increasing activation functions. However, learning these neural networks is a difficult task, because sinusoidal activation functions are non-monotonous. In addition, continuous time is not modeled explicitly in this approach.

function	# inputs	a	b	c	meaning
AND	2	$\frac{\pi}{4}$	$-\frac{\pi}{4}$	$\sqrt{2}$	logical conjunction
XOR	2	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	1	exclusive or
ODD	n	$\frac{\pi}{2}$	$(n-1)\frac{\pi}{2}$	1	odd parity, returns 1 iff an odd number of inputs is 1

Fig. 6: Implementing logical functions for one Fourier neural network unit with activation function $c\sin(ax+b)$. The Boolean values *true* and *false* are represented by $+1$ and -1, respectively.

3.2 Continuous Neural Networks

[9] introduces neural networks with an uncountable number of hidden units. While such a network has the same number of parameters as an ordinary neural network, its internal structure suggests that it can represent some smooth functions more compactly. [9] presents another approach for neural networks with an uncountable number of units, where the weighted summation of input values is replaced by integration. Because of this, they are called continuous neural networks. However, continuous time and hence temporal processing is not modeled explicitly there, which is the primary goal of this paper.

In [10], neural networks are used in a nonlinear system identification algorithm for a class of nonlinear systems. The algorithm consists of two stages, namely preprocessing the system input and output and neural network parameter estimation. However, first and foremost, it is only applicable to the analysis of control systems with a special structure.

3.3 Finite Impulse Response Perceptrons

Temporal processing in neural networks means to deal with dynamic effects and to introduce time delays in the network structure [4]. Therefore, in the *finite-duration impulse response (FIR) model*, temporal processing is realized by a linear, time-invariant filter for the synapse i of a neuron j. Its impulse response $h_{ji}(t)$ depends on a unit impulse at time $t = 0$. Typically, each synapse in the FIR model is causal and has a finite memory, i. e. $h_{ji}(t) = 0$ for $t < 0$ or $t > \tau$, with the memory span τ for all synapses. The response of a synapse can be defined as the convolution (auto-correlation) of its impulse response with the input $x_i(t)$. Thus, we can express the output as $h_{ij}(t) * x_i(t) = \int_{-\infty}^{t} h_{ji}(u)x_i(t-u)du$. The net activation potential over all p synapses, with threshold θ_j, is given by $v_j(t) = \sum_{i=1}^{p} \int_0^{\tau} h_{ji}(u)\,x_i(t-u)\,du - \theta_j$, where the overall output is the sigmoidal nonlinear logistic activation function (see above). With this, an artificial neuron can represent temporal behavior. The FIR multi-layer perceptron, with its hidden and output neurons based on this FIR model, has been applied for adaptive control, dynamic system identification, and noise cancellation. Once trained, all synaptic weights are fixed. Then, the network can operate in real time.

Instead of the FIR model, where time is simulated by additional copies of a neuron for different times (cf. Sect. 2, Scenario 2), *real-time recurrent networks* (cf. [4]) are designed by using a common neural model, where the temporal processing is realized by the feedback of the network.

3.4 Hybrid Automata

Another model that allows to model discrete and dynamic changes of its environment and hence continuous time are *hybrid automata*, a combination of Moore and Mealy automata [5]. A hybrid automaton is a mathematical model for describing systems, where computational processes interact with physical processes. In contrast to simple finite state automata, well-known in computer science [3, 12], their behavior is stated not only by discrete state transitions, but also by continuous evolution. Hybrid automata consist of a finite set of states and transitions between them. Thus, continuous flows within states and discrete steps at the transitions are possible. If the state invariants do not hold any longer, a discrete state change takes place, where a jump condition indicates which transition shall be used. Then, a discrete step can be done, before the next state is reached. States are annotated with invariants and flow conditions, which may be differential equations. There, the continuous flow is applied to the variables within the state invariants. Thus, the behavior of the robot in Scenario 2 can be described as shown in Fig. 7. Hybrid automata, however, are not well-suited for mapping continuous input with periodic behavior. In addition, (hybrid) automata cannot be learned easily by examples as e. g. neural networks.

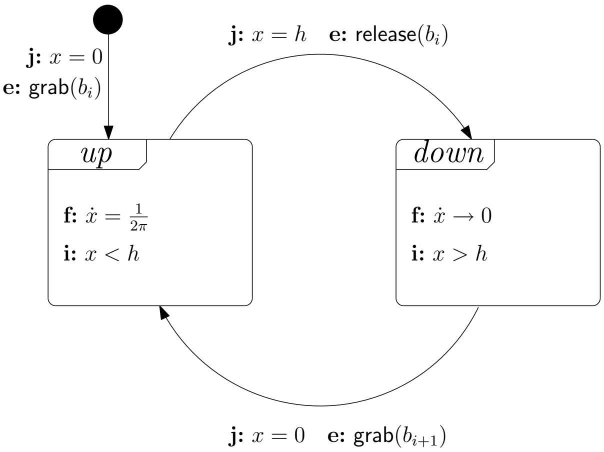

j: $x = 0$
e: grab(b_i)
j: $x = h$ e: release(b_i)
up
f: $\dot{x} = \frac{1}{2\pi}$
i: $x < h$
$down$
f: $\dot{x} \to 0$
i: $x > h$
j: $x = 0$ e: grab(b_{i+1})

Fig. 7: Hybrid automaton for the robot arm (Scenario 2).

3.5 Central Pattern Generators

For Scenario 2, oscillating, periodic patterns must be generable. This can be achieved, if a single unit is able to oscillate spontaneously, as we will assume here (cf. Def. 2). Alternatively, recurrently connected units can trigger each other, inducing periodic patterns. Such a system is called *central pattern generator* (CPG). They can be defined as neural networks that can endogenously (i.e. without rhythmic sensory or central input) produce rhythmic patterned outputs [6] or as neural circuits that generate periodic motor commands for rhythmic movements such as locomotion [7]. CPGs have been shown to produce rhythmic outputs resembling normal rhythmic motor pattern production even in isolation from motor and sensory feedback from limbs and other muscle targets. To be classified as a rhythmic generator, a CPG requires: two or more processes that interact such that each process sequentially increases and decreases, and that, as a result of this interaction, the system repeatedly returns to its starting condition.

4 Towards Continuous-Time Neural Networks

We will now define *continuous-time neural networks* (CNN). With them, we are capable of modeling the three general scenarios, introduced in Sect. 2. At first glance, they are very similar to standard neural networks, because they also consist of an interconnected group of units. In fact, a CNN degenerates to an ordinary neural network, if the extended functionality is not used. We distinguish several types of units (see Def. 1 and 2).

Definition 1 (input and output units, on-neurons). *In a CNN, there may be one or more* input and output units*. Input units do not have any incoming edges, while output units do not have any outgoing edges. In the following, we restrict our attention to networks with only one output unit. The values of the input units* $x_1(t),\ldots,x_n(t)$ *and of the output unit* $y(t)$ *depend on the time* t*. There may also be so-called* on-neurons*, i. e.*

units without incoming edges, yielding a constant output c, independent from the actual time t.

In our model, as in standard neural networks, we assume that the input value of a unit j is a weighted sum of the incoming values, and we have a nonlinear activation function. But in addition, we have two further optional components in each unit (for integration over time and for enabling oscillation) that may be switched on or off. Furthermore, inputs may be delayed or not. This is summarized in the following definition, leading to a unit with up to four stages, called *sub-units* in the sequel:

Definition 2 (continuous neural network unit). *In general, a CNN unit computes its output value $y(t)$ from its input values $x_1(t),\ldots,x_n(t)$, which may be the overall input values of the network or the output values of immediate predecessor units, in four steps. Each step yields the value $y_k(t)$ with $1 \leq k \leq 4$, where $y(t) = y_4(t)$. For $k \geq 2$, the respective sub-unit may be switched off, which means that $y_k(t) = y_{k-1}(t)$.*

1. ***summation:*** *The input value of the unit is the sum of the incoming values $x_i(t)$ with $1 \leq i \leq n$, each weighted with a factor w_i and possibly delayed by a time amount δ_i, which is 0 by default:*

$$y_1(t) = \sum_{i=1}^{n} w_i \cdot x_i(t - \delta_i)$$

2. ***integration:*** *In certain cases, the integrated activity, i. e. the average signal power, is useful. Therefore, we introduce an optional integration process, which is switched off by default.*

$$y_2(t) = \sqrt{\frac{1}{\tau} \int_{t-\tau}^{t} y_1(u)^2 \, du}$$

Note that, for $\tau \to 0$, we have $y_2(t) = |y_1(t)|$, i. e., the unit is switched off for positive values. If it is switched on, we take $\tau \to \infty$ by default. Alternatively, the statistical variance of $y_1(t)$ could be used here.

3. ***activation:*** *In order to be able to express general, non-linear functions, we need a non-linear activation function (cf. [4]). Instead of the often used logistic function (cf. Sect. 3), we use the hyperbolic tangent here, because $\tanh(x) \approx x$ for small x and the range of the hyperbolic tangent is $[-1;+1]$, which corresponds to the range of sinusoidal periodic functions. It holds:*

$$y_3(t) = \frac{\tanh(\alpha \cdot y_2(t))}{\alpha}$$

We make use of a factor α that retains these properties here. By default, $\alpha = 1$. For $\alpha \to 0$, the sub-unit is switched off.

4. ***oscillation:*** *The unit can start to oscillate with a fixed (angular) frequency ω:*

$$y_4(t) = y_3(t) \cdot \cos(\omega t)$$

This corresponds to amplitude modulation of the input signal. In principle, other types of modulation, e. g. frequency or phase modulation, would be possible, but this is not considered here. For $\omega = 0$, this sub-unit is switched off.

With this type of units, all scenarios, introduced in Sect. 2, can be implemented. If the integration and the oscillation sub-unit is switched off, the functionality of the unit is identical with that of standard neural network units (cf. Sect. 3 and [4, 15]). Hence, all logical boolean functions (Scenario 1) can be expressed easily, of course, in contrast to Fourier neural networks, generally with hidden units. Everything that can be expressed by an ordinary neural network can be expressed by a CNN, because the former one is a special case of a CNN.

Scenario 2 can be implemented with several oscillating units, i. e. $\omega_k \neq 0$, because it is known from the study of Fourier series, that arbitrary periodic functions can be written as the sum of simple waves represented by sines and cosines. For the sawtooth-like graph (Fig. 3), we have $f(x) = \frac{h}{2} - \frac{h}{\pi}\sum_{k=1}^{\infty}\frac{1}{k}\cdot\sin(\frac{2\pi}{T}kx)$. The latter sum may be approximated by the first n summands, which can be expressed by n oscillating CNN units (see Fig. 8).

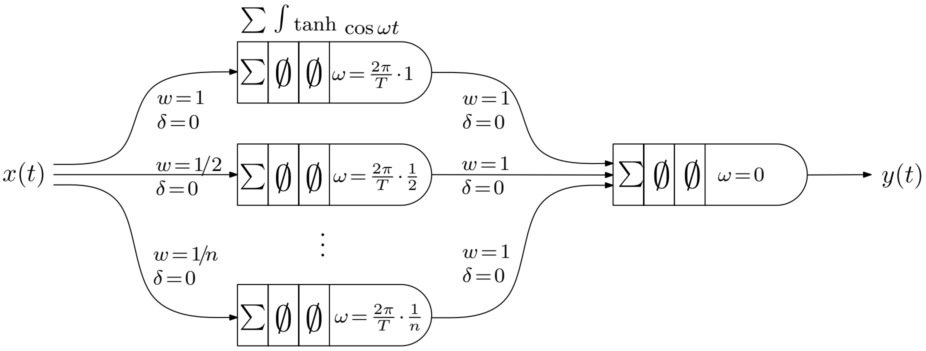

$\sum$ $\int$ tanh $\cos\omega t$
$\sum$ $\emptyset$ $\emptyset$ $\omega=\frac{2\pi}{T}\cdot 1$
$w=1$
$\delta=0$
$x(t)$
$w=1/2$
$\delta=0$
$\sum$ $\emptyset$ $\emptyset$ $\omega=\frac{2\pi}{T}\cdot\frac{1}{2}$
$w=1/n$
$\delta=0$
$\sum$ $\emptyset$ $\emptyset$ $\omega=\frac{2\pi}{T}\cdot\frac{1}{n}$
$w=1$
$\delta=0$
$w=1$
$\delta=0$
$w=1$
$\delta=0$
$\sum$ $\emptyset$ $\emptyset$ $\omega=0$
$y(t)$

Fig. 8: Network with several oscillating units for Scenario 2. Sub-units, that are switched off, are marked with $\emptyset$.

In Scenario 3, we have to find out the period length T of a task automatically from a complex signal, e. g. the duration of an episode of the robot at the conveyor belt (Scenario 2, Fig. 2). For this, consider the function $x(t) = \cos(\omega_1 t) + \cos(\omega_2 t)$, whose overall period length depends on the ratio ω_2/ω_1. Let $\omega_1 = 2\pi$ and $\omega_2 = \sqrt{2}\omega_1$. The corresponding graph for $x(t)$ is shown in Fig. 9. In order to determine the overall period length, we must be able to find out the so-called missing fundamental frequency, i. e., we have to find a time duration T such that $x(t) - x(t-T)$ becomes zero. Applying the least squares method, this could be turned in finding the minima (almost zeros) of $1/T\int_0^T (x(u) - x(u-T))^2\,du$, i. e., we overlap the original signal ($\delta = 0$, $w = 1$) with a phase-shifted and inverted copy of itself ($\delta = T$, $w = -1$), which yields an effect of *comb filtering* (cf. [8]).

Fig. 10 shows the graph for the square root of the latter integral in dependency from T, which can be achieved by switching on the integral sub-unit. It has minima near 5 and 12 (and also near 7 and 10) which alternatively can be derived by continued fraction development of the ratio ω_2/ω_1 [13, 17]. Thus, the corresponding CNN unit yields approximately constant output wrt. t, namely the values shown in the graph in Fig. 10, where small values near 0 indicate periodicity. This procedure allows us to express analysis of periodic behavior as desired.

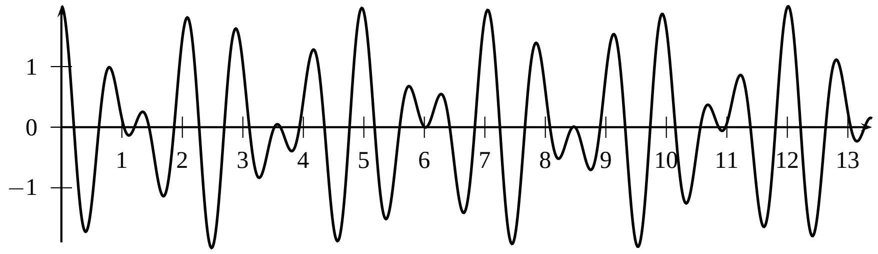


Fig. 9: Complex periodic signal $x(t) = \cos(\omega_1 t) + \cos(\omega_2 t)$ with $\omega_2/\omega_1 = \sqrt{2}$.

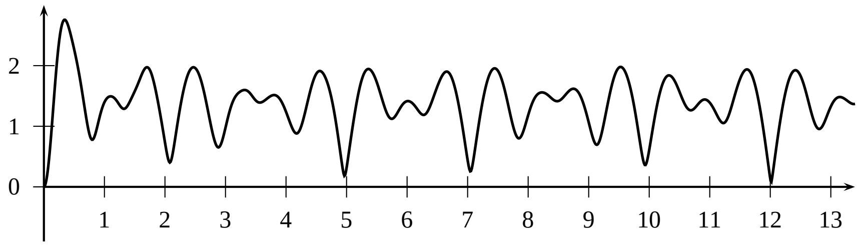


Fig. 10: Periodicity analysis for complex signal $x(t)$. The graph shows the output of the comb-filtered signal in dependency from the delay in time T, checking the period length of the overall signal, with main minima at $T = 5$ and $T = 12$. It is constant wrt. the current time t.

5 Conclusions

In this paper, we sketched ongoing work on neural networks with continuous time. These networks can support the modeling of behavior synthesis and analysis in robotics and for cognitive systems. For arbitrary continuous, periodic input, the robot or the agent in general has to react continuously and within a certain time interval. Hence, complex, physical and/or cognitive processes can be modeled adequately by a CNN. A CNN without recurrence and constant values for the angular frequencies ω_k in the oscillation sub-units and switched-off integration sub-units correspond to standard neural network units in principle. Thus, the classical backpropagation method can be employed for learning a CNN from examples, where a set of input and output values must be given for different time points t. Therefore, future work will implement this theory. We intend to do this on a concrete autonomous robot platform, namely a quadrocopter, i. e. flying robots with four propellers. Analysis of the network is also an important part and will be investigated in further detail, too.

References

[1] M. Ebeling. Neuronal periodicity detection as a basis for the perception of consonance: A mathematical model of tonal fusion. *Journal of the Acoustical Society of America*, 124(4):2320–2329, 2008.

[2] U. Furbach, J. Murray, F. Schmidsberger, and F. Stolzenburg. Hybrid multiagent systems with timed synchronization – specification and model checking. In M. Dastani, A. El Fallah Seghrouchni, A. Ricci, and M. Winikoff, editors, *Post-Proceedings of 5th International Workshop on Programming Multi-Agent Systems at 6th International Joint Conference on Autonomous Agents & Multi-Agent Systems*, LNAI 4908, pages 205–220, Honolulu, 2008. Springer, Berlin, Heidelberg, New York.

[3] A. Gill. *Introduction to the Theory of Finite-State Machines*. McGraw-Hill, New York, 1962.

[4] S. Haykin. *Neural Networks: A Comprehensive Fondation*. Macmillan College Publishing, New York, 2nd edition, 1994.

[5] T. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292, New Brunswick, NJ, 1996. IEEE Computer Society Press.

[6] S. L. Hooper. Central pattern generators. *Embryonic Encyclopedia of Life Sciences*, 1999.

[7] A. D. Kuo. The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor Control*, 6:129–145, 2002.

[8] G. Langner. Temporal processing of periodic signals in the auditory system: Neuronal representation of pitch, timbre, and harmonicity. *Zeitschrift für Audiologie*, 46(1):8–21, 2007. In German.

[9] N. Le Roux and Y. Bengio. Continuous neural networks. In *Proceedings of 11th International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007.

[10] Y. Liu and J. J. Zhu. Continuous-time nonlinear system identification using neural network. In *Proceedings of American Control Conference*, pages 613–618, Seattle, USA, 2008.

[11] L. Mingo, L. Aslanyan, J. Castellanos, M. Diaz, and V. Riazanov. Fourier neural networks: An approach with sinusoidal activation functions. *Information Theories and Applications*, 11:52–55, 2004.

[12] Object Management Group, Inc. *UML Version 2.1.2 (Infrastructure and Superstructure)*, November 2007.

[13] C. D. Olds. *Continued Fractions*, volume 9 of *New Mathematical Library*. L.W. Singer Company, 1963.

[14] F. Ruh and F. Stolzenburg. Translating cooperative strategies for robot behavior. In G. J. Nalepa and J. Baumeister, editors, *Proceedings of 4th Workshop on Knowledge Engineering and Software Engineering at 31st German Conference on Artificial Intelligence*, CEUR Workshop Proceedings 425, pages 85–96, Kaiserslautern, 2008.

[15] S. Russell and P. Norvig. *Artificial Intelligence – A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[16] A. Silvescu. Fourier neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN 1999)*, pages 488–491, Washington D. C., 1999.

[17] F. Stolzenburg. A periodicity-based theory for harmony perception and scales. In *Proceedings of 10th International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009. To appear.

Bayesian Identification of Problem-Solving Strategies for Checking the ACT-R/Brain-Mapping Hypothesis

Claus Möbus and Jan Charles Lenk

Learning and Cognitive Systems, Department of Computing Science
Carl von Ossietzky Universität Oldenburg
26111 Oldenburg, Germany
{claus.moebus|jan.lenk}@uni-oldenburg.de,
WWW home page: http://www.lks.uni-oldenburg.de/

Abstract. John R. Anderson proposed a correspondence between ACT-R modules and brain regions (brain mapping hypothesis). To check this conjecture we plan to compare model-generated blood-oxygen-level dependent (BOLD) signal curves with BOLD curves obtained from functional Magnetic Resonance Imaging (fMRI) scans. In contrast to Anderson's studies our subjects were not urged to follow a *single* strategy but construct their *personal* strategy within a constraint-based strategy space. So, the mapping hypothesis has to be checked strategy-specific. The identification of strategies was difficult because subjects were not able to identify their own in a retrospective manner. So we used Response-Time (RT) data in combination with a Bayesian Belief Net to identify *personal* problem solving strategies without using fMRI data for checking the mapping hypothesis.

1 Introduction

Of late, one of the busiest branches of ACT-R related research is focusing on the neurophysiological analogy of the ACT-R theory [1]. The ACT-R architecture provides a set of modules with specific functions. Anderson [2, 3] postulates a mapping between these modules and brain regions. For instance, the procedural module is mapped onto the basal ganglia, while the declarative module is mapped around the inferior frontal sulcus. The ACT-R 6.0 implementation provides a set of tools which directly predict BOLD signals for these brain regions.

Several studies were conducted by Anderson et. al. in order to verify the mapping hypothesis. These included experiments from various domains, as algebraic problem solving [4, 5], associative learning [6], or insight problems [7]. One particular feature in common of all these experiments was the fact that participants had to employ the same problem solving strategy on all tasks.

The empirical validation of the mapping hypothesis is among the research goals of our multidisciplinary research project[1]. While also the effects of affective and informa-

[1] *Cognitive Modeling and Bayesian Identification Analysis* (CoMBIAn), work package within project *Impact of affective and informative feedback on learning in children before and after a reattribution training: An integrated approach using neuroimaging, educational research and modeling*, Möbus, Moschner, Parchmann & Thiel (main applicant), BMBF-Programme for the Promotion of Scientific Collaboration between the Neurosciences and Research on Learning and Instruction, 03/01/2008 − 02/28/2011

tive feedback on learning are being studied [8], an accompanying fMRI study offers us the possibility to compare BOLD signal predictions generated from strategy-specific ACT-R models with BOLD signals obtained from actual fMRI scans. However, the difficulties we encountered during our efforts suggested a refinement of our modelling methods. In contrast to the experiments described by Anderson et. al. [4], the tasks in our experimental setting were far more complex, because in order to solve these tasks, participants were free to choose their *personal* strategies. Because different strategies lead to different brain region activation predictions, we had to model these different strategies and identify the chosen subject-specific strategy *without* using fMRI data (Fig. 1). We would work unduly in favor of the mapping hypothesis if we would assign subjects to strategies according to similarity of their BOLD curves with the strategy-specific ACT-R-BOLD curves.

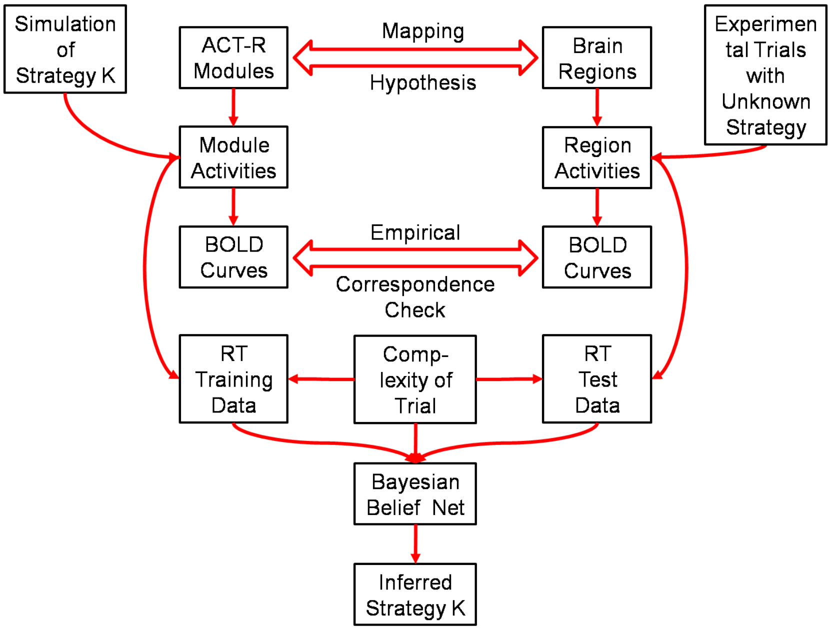


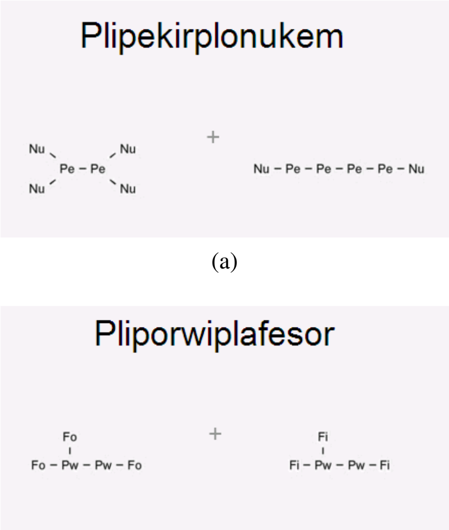
Fig. 1: Process model for checking the mapping hypothesis and classification of subjects according to their *personal* strategies.

Our short-term research goal is now to devise a method of identifying strategies from behavioural data which does not include fMRI-data. In this paper, after having shortly described ACT-R and its mapping hypothesis, we will present the experiment, discuss the modelling of strategies by the means of two example strategies, and present the method and results of the Bayesian Identification of *personal* problem-solving strategies.

2 Experimental design

All participants were children in the age between 11 and 12. The exercises which the children had to solve come from the domain of the chemical formula language [9], which is generally unknown to children of that age. However, instead of real-world chemical elements, pseudo-elements (like *"Pekir"* or *"Nukem"*) were used to ensure that the children only applied the rules of the artificial formula language.

The children were asked to answer 80 trials during fMRI scans. A single trial consists of the auditive and visual presentation of a chemical compound name and the visual presentation of a pair of structural formulae (Figs. 2a and 2b). A selection between two distinct structural formulae, one on the left, the other on the right, is expected from the subject. The participant has to select the correct structural formula to match the compound name. The total presentation lasts for 4.5 seconds. An additional time of 1 second for the answer is granted, so that the maximum response time amounts to 5.5 seconds.



(a)

(b)

1. The abbreviation for an element is defined by two letters
2. The first letters of the abbreviation are also the first letters of the names of the elements
3. Both letters appear in the elements name
4. An element may have a multiplicity from 1 to 4 in the compound. Distinct numerals are used to denote the multiplicity:
   1 -
   2 pli
   3 pla
   4 plo
5. The position of a numeral is always before the element in the compound name
6. The central element of the structural formula is always the first in the compound name

(c)

Fig. 2: Two of 80 trials, trial A (Fig. 2a) and trial B (Fig. 2b). The compound name is at the top, structural formulae left and right below. For both trials the left formula matches. The rules on the right (Fig. 2c) determine the correct formula.

If the response occurred in time, a feedback is given. The feedback is split into two parts: one part informs about the participant's performance; a second, (hopefully) affective part informs about the performance of a fictional peer group.

In order to find the correct structural formula for a compound name, six semantically partly overlapping rules (Fig. 2c), which are part of the instruction given to all participants, have to be applied and checked for violations. These rules define the constraints of a strategy space from which correct *personal* strategies can be constructed by the subjects. There is no explicit order in which the rules should be applied. Either the left or the right formula violates at least one of the rules. The trials are thus classified by the position of the faulty formula (left/right) and by the number of the violating rule.

The rules were well known by the children because they went through an extensive instruction phase in multiple sessions. They familiarised themselves with the rules using age-based material and games especially designed for that purpose. They also passed 20 trials on a computer and another 40 in a fMRI simulator before entering the actual fMRI experiment.

While the chemical formula language as context of the problem seems to be more typical of algorithmic problems, the evidence of a clear goal combined with multiple solution paths suggests the classification as a rule-using problem as described by Jonassen [10], albeit a well-structured one, since rules and all parts of the problem are available to the problem solver. However, the complexity is higher than that of the problem in previous research [4–7] in this area. These may be considered true algorithmic problems in the sense that they have to be solved by a fixed number of steps.

3 Strategy Modelling

Two input channels are available to the problem solver. The visual input channel is mandatory, while the auditory input channel is auxiliar. This fact adds to the complexity of the problem, especially as both channels may be perceived in parallel or consecutively. Either the left or the right formula or both have to be evaluated visually. This results in a variability of conceivable strategies, which differ in efficiency as well as module activation. Two of them are shown in Fig. 3. A set of *basic tasks* is derived from the rules in Fig. 2c. These tasks are shared by all strategies, though not necessarily in the order presented here:

1. Visually and/or auditorially perceive and encode the different parts of the compound name (mandatory for any successful strategy)
2. Count the outer elements of a structural formula and compare them with the first numeral in the compound name
3. Count the inner elements of a structural formula and compare them with the second numeral in the compound name
4. Compare the inner element with the first element of the compound name
5. Compare the outer element with the second element of the compound name
6. Indicate the correct formula

Tasks 2-5 may be applied to both formulae, or, more efficiently, to either the left or the right formula. It should be noted that some concurrency can take place if the compound name is encoded using only auditory input. Tasks 4 and 5 may be split into two different tasks as the abbreviation of an element always consists of two letters. Since the first letter is easier to compare with the name, it may be more appropriate to prioritise the first comparison and leave the second letter for later. A second open question which is not reflected within the above list of tasks is the position of the retrieval for the numerals. It can take place very early when encoding the compound name, but there is also the possibility to retrieve the numeral later on between the counting and comparison stages.

While all the strategies share the same basic set of tasks, they all perform differently on each trial. Some trials may only be solved by counting the elements (Fig. 2a), others by name-element comparisons (Fig. 2b), others by both. A strategy shows higher performance (shorter response time) if it concentrates on a single structural formula to decide whether it matches or not. Each trial class (the violated rule and location of the violating formula) has an impact on the performance of the strategy.

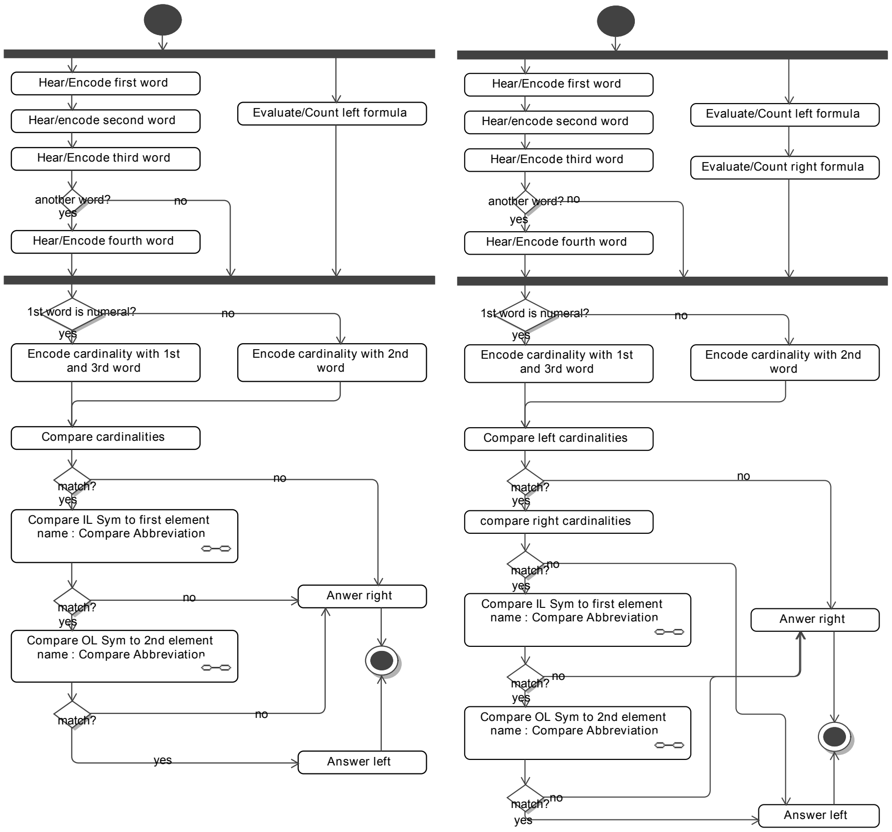
Several, though so far not all possible, strategies were modelled, at first on an abstract layer as UML activity diagrams (Fig. 3), and subsequently within the ACT-R environment as a set of production rules. As only expert participants were modelled, all strategies find the correct answer but with a large variation in performance. For example, model A (Fig. 3a), which already counts when listening to the compound name, performs extremely fast by taking only one formula into consideration. A similar model B (Fig. 3b) which checks both formulae shows slower performance.

In a subsequent interview, nearly all children stated that they counted the elements before comparing the abbreviations with the names, which is reflected in our modelling. Another indication for the correctness of this assumption can be seen in Fig. 4. Trial B may not be solved by counting; abbreviations have to be compared with the compound name instead. It took all children considerably longer to solve trial B. Also a few children said that they looked at a single formula and not at both. So a single-formula strategy is plausible and must be taken into account, even if it may not be the common case. Indeed, a few children show a fast response, resembling that of model A in Fig. 3a.

Most children with a high success rate stated that they benefited from the aural presentation of the compound name. So far, our models use solely the auditive input for encoding the compound name, although alternatives will be implemented later. Both models predict adequate response times for trial A, but fail to do so for trial B, which can be seen in Fig. 4. Apparently, both models are too fast, which may be a hint that our productions which compare the abbreviations are too effective.

Both models perform quite differently on trial A, as can be seen in the module traces in Fig. 5. Model A shows less activity in the declarative and visual modules, since it does not count as much as the second model. However, using the ACT-R 6.0 built-in fMRI tools, we found that this has an impact on the BOLD prediction for some of the regions (figs. 5c and 5d). Note that this is just the case for two models which are very similar in their design. This effect is even more distinctive if the models differ in their conception.

For instance, any realisation of Task 1, percieving and encoding the compound name, would surely engage ACT-R's visual or aural module, if not both, and the imaginal module. Tasks 2 and 3, which encompass encoding and counting the structural formulae, would involve the imaginal, the visual and the retrieval module, as well as the imaginal module. Tasks 4 and 5 would also require at least the imaginal module, but


Hear/Encode first word
Hear/encode second word
Hear/Encode third word
Evaluate/Count left formula
another word?
no
yes
Hear/Encode fourth word
1st word is numeral?
no
yes
Encode cardinality with 1st and 3rd word
Encode cardinality with 2nd word
Compare cardinalities
match?
no
yes
Compare IL Sym to first element name : Compare Abbreviation
match?
no
yes
Anwer right
Compare OL Sym to 2nd element name : Compare Abbreviation
match?
no
yes
Answer left
Hear/Encode first word
Hear/encode second word
Hear/Encode third word
Evaluate/Count left formula
Evaluate/Count right formula
another word? no
yes
Hear/Encode fourth word
1st word is numeral?
no
yes
Encode cardinality with 1st and 3rd word
Encode cardinality with 2nd word
Compare left cardinalities
match?
no
yes
compare right cardinalities
match? no
yes
Compare IL Sym to first element name : Compare Abbreviation
Anwer right
match? no
yes
Compare OL Sym to 2nd element name : Compare Abbreviation
match? no
yes
Answer left

(a)

(b)

Fig. 3: Two example strategies as UML activity diagrams. This kind of diagram was chosen because it provides an abstraction from ACT-R rule models. Both start counting the elements of a formula while still listening. They differ because model A (Fig. 3a) evaluates only the left formula, while model B (Fig. 3b) examines both left and right formulae. Subsequently, they resolve the numerals and compare them with the cardinalities of the left structural formula or both formulae respectively. If there is no discrepancy in the cardinalities, both models check the inner element symbol of the left formula (IL) for consistency with first element's name, and if needed, continue to do so with the outer symbol of the left formula (OL).

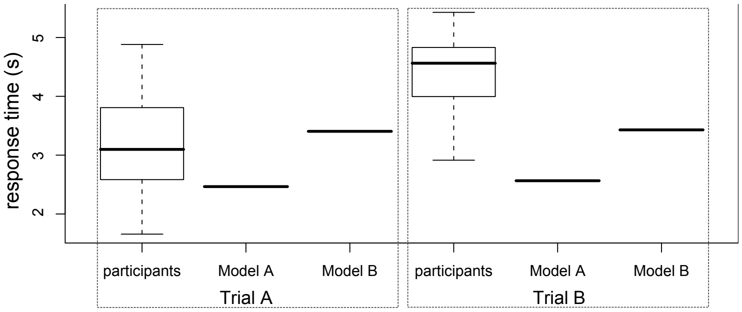


Fig. 4: Box plot of the participants' response times for correct answers for trial A from Fig. 2a (left) and trial B from Fig. 2b (right). Model results from a single model run are shown next to the box plots.

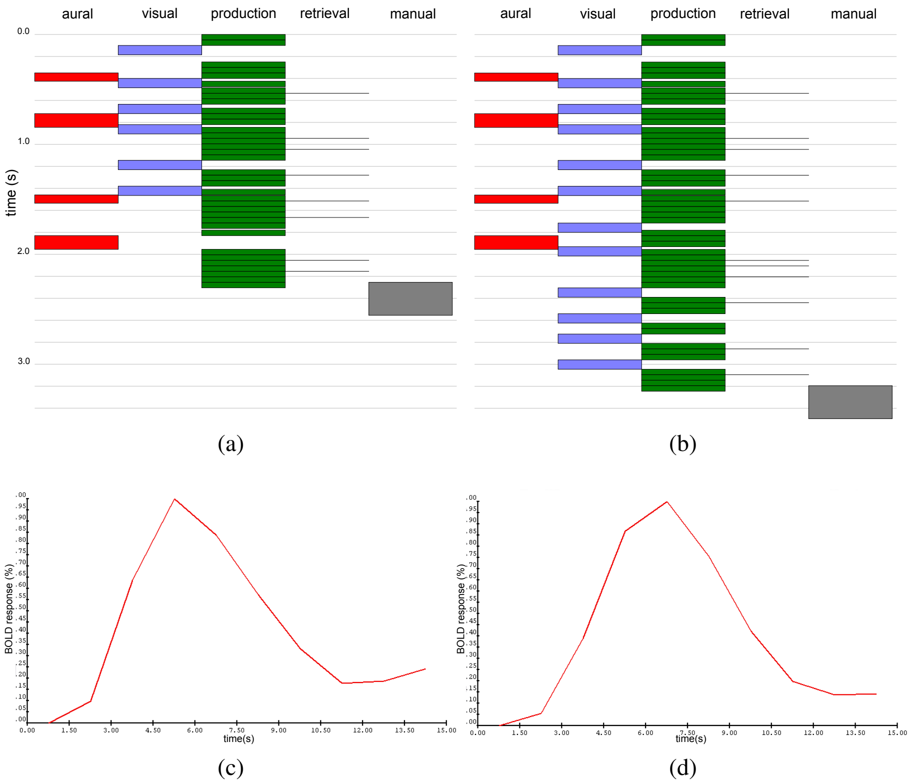
it could involve the visual module if the second letter of the symbol has to be checked for occurance in the compound name. As the Tasks 2-5 can be arranged in any arbitrary order, or even be splitted into subtasks which could run in parallel, quite different patterns of module activation would emerge. This implies that even models which produce similar behaviours may predict distinct BOLD signals, if the productions involved activate different modules.

4 Bayesian analysis of behavioural non-fMRI data

It is doubtful whether the participants are able to remember their problem solving strategy for each trial. It is also possible that they applied different strategies to trials. This switching hypothesis could be investigated by Hidden Markov Models (HMMs). The choice of strategy may be related to the trial configuration. However, we assume that the participants already settled for a *single* strategy after the extensive instruction and training phases. In order to determine which of our models is suitable to explain the performance of the actual strategy used by the participant, we devised a Bayesian Belief Network (BBN) [11] as diagnostic tool for identifying the personal trial-independent strategy of the subject.

The main idea is that all models produce distinct response times for each trial. The class of the trial, i.e. the criteria which need to be checked to solve the trial, is known. We assume that response times for a strategy are dependent on the class of the trial. This may be derived from the task structure: As the order of the rule checks vary from model to model, they produce different response times for each trial. This is reflected in the BBN in Fig. 6. Note that this is a naive Bayesian classifier with two latent variables `Complexity_Left` and `Complexity_Right`; response times and rules applied to the trial are indicators for a strategy.

The probability tables of the BBN are being learned by running all the strategy-specific ACT-R models to generate cases. This results in a data matrix whose columns correspond to the nodes from the BBN and whose rows correspond to trials. The two



(a)

(b)

(c)

(d)

Fig. 5: Buffer traces from model A (Fig. 5a) and model B (Fig. 5b) on trial A. In Fig. 5a, model A shows considerable less activity in the visual system (second column from the left) than model B in Fig. 5b. This manifests itself in the BOLD-predictions for the visual module of model A (Fig. 5c). The BOLD-prediction for model B in Fig. 5d peaks about 2 seconds later and decays slower than the prediction for model A in Fig. 5c.

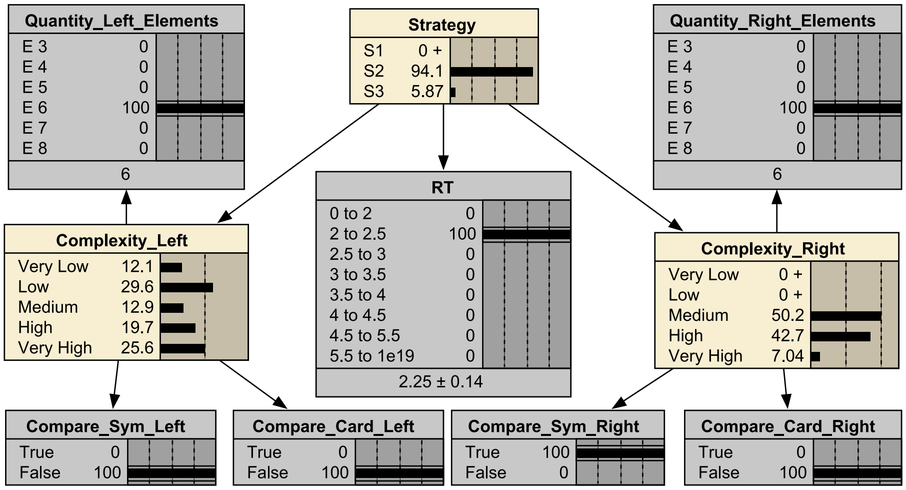
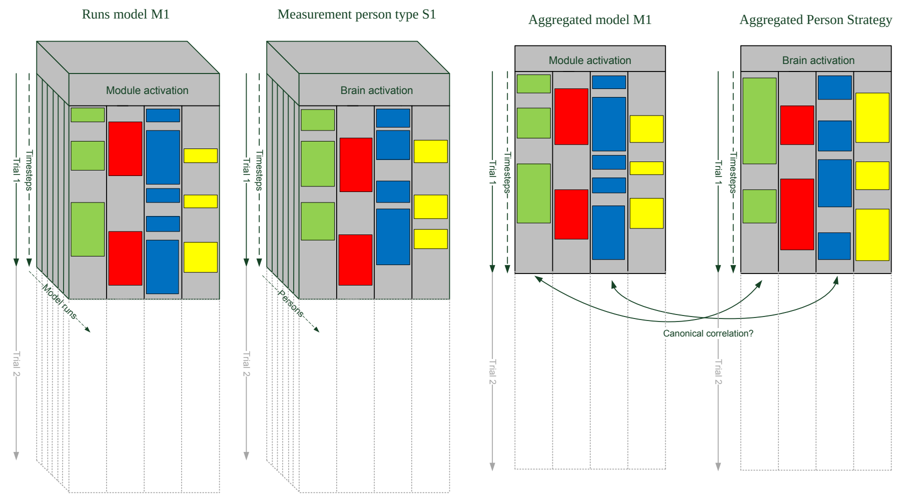


Fig. 6: Bayesian Belief Network for strategy identification. When using the training data as test cases, inference on the strategy node shows an error rate of 35.83%. An Augmented Bayesian Classifier, not shown here, improves the error rate to 23.23%.

latent nodes `Complexity_Left` and `Complexity_Right` in the BBN describe the complexity of each structural formula. The complexity is dependent on the number of elements in the formula and which rules have to be applied to the formula. The ACT-R models provide response times for the `RT` node. The `Strategy` node indicates the strategy of the ACT-R model. Because of the existence of latent variables the expectation-maximization learning method [11] is used to learn the probabilities from the cases. At best a response and the complexities are perfect indicators for a single strategy under a given trial configuration.

After training the BBN, it is used for classifying the subjects' data according to the inferred strategy. The response time of a subject, together with the trial class, is entered into the BBN as evidence. It is then possible to infer the most likely strategy. If, for example, the class of trial A is entered as evidence (6 elements for each formula and the exclusion criteria, being `True` for symbol comparison on the right, otherwise `False`), and a response time between 2.5 and 3 seconds, strategy `S_2` (which corresponds to model A) is indicated with the highest probability in the `Strategy`-Node (Fig. 6).

As can be seen in Fig. 1, we intend to check the BOLD-prediction against BOLD signals from the fMRI data for a chosen strategy. If the mapping hypothesis is correct, the corresponding simulated and real-world BOLD signals should correlate significantly positive. For this reason, we identify all persons who share the same strategy with the BBN (Fig. 7a). Subsequently, we test the correlation between the aggregation of all their BOLD responses in the defined regions with the aggregated BOLD-prediction for the ACT-R modules by the model (Fig. 7b). If the mapping hypothesis is correct, such corresponding correlations should be higher than between any other module-region pairs.



(a) (b)

Fig. 7: Activations and BOLD curves obtained from strategy-specific ACT-R model runs and subject fMRI scans (fig 7a) are aggregated across simulation runs and persons(Fig. 7b). These aggregated data are used for computing strategy-specific module-region correlation matrices.

5 Conclusion

So far, our research work is concentrating on the actual identification and modelling of the possible strategies. According to our new approach it is now possible to validate our RT-predicted strategies with behavioural and fMRI data. This was not done in the brain-mapping related research before. E.g. Anderson simplified problem solving strategies and urged subjects to follow a *single* strategy. With our method it is possible to use more complex and *personal* strategies. This is possible as long as strategies could be analyzed rationally before data analysis of fMRI scans starts.

The analysis of the experimental data and the definition and implementation of strategies is currently in progress. If the mapping hypothesis is correct, there should be a correlation between the aggregated model-brain activations.

References

1. J. Anderson, D. Bothell, M. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An integrated theory of the mind,” *Psychological Review*, vol. 111, pp. 1036–1060, 2004.
2. J. Anderson, *How can the human mind occur in the physical universe?* Oxford University Press, 2007.
3. J. Anderson, J. M. Fincham, Y. Qin, and A. Stocco, “A central circuit of the mind,” *Trends in Cognitive Science*, vol. 12, pp. 136–143, 2008.
4. J. Danker and J. Anderson, “The roles of prefrontal and posterior parietal cortex in algebra problem-solving,” *Neuroimage*, vol. 35, pp. 1365–1377, 2007.
5. A. Stocco and J. Anderson, “Endogenous control and task representation: An fmri study in algebraic problem solving,” *Journal of Cognitive Neuroscience*, vol. 20, no. 7, pp. 1300–1314, 2008.
6. J. Anderson, D. Byrne, J. M. Fincham, and P. Gunn, “Role of prefrontal and parietal cortices in associative learning,” *Cerebral Cortex*, vol. 18, no. 4, pp. 904–914, 2008.
7. J. R. Anderson, J. F. Anderson, J. L. Ferris, J. M. Fincham, and K.-J. Jung, “The lateral inferior prefrontal cortex and anterior cingulate cortex are engaged at different stages in the solution of insight problems,” in *Proceedings of the National Academy Sciences*, 2009.
8. J. Özyurt, M. Rietze, and C. Thiel, “fMRI of feedback processing in children and adults,” in *Frontiers in Human Neuroscience*, 2008.
9. S. Heuer and I. Parchmann, “Son$_2$e oder Fus$_2$bal$_2$ - wie Sechstklässler die chemische Formelsprache interpretieren,” *Naturwissenschaften im Unterricht - Chemie*, vol. 19, 2008.
10. D. Jonassen, “Toward a design theory of problem solving,” *Educational Technology Research and Development*, vol. 48, no. 4, pp. 63–85, 2000.
11. F. V. Jensen and T. Nielsen, *Bayesian Networks and Decision Graphs*. Springer, 2007.

Predicting Changes: A Cognitive Model for Dynamic Stocks and Flows

Felix Steffenhagen, Marco Ragni, Ivo Chichkov, Andreas Klein

University of Freiburg, Center of Cognitive Science
D-79098 Freiburg, Germany

Abstract. We briefly report about experimental investigations we conducted for the so-called dynamic stocks and flows task (DSF) and present a cognitive model to replicate human's behavior. The goal in the DSF task is to maintain a certain level of water in a tank under the influence environmental flows which depend on unknown dynamics. Our findings are complemented by an analysis of recent experimental data from the literature. The results are integrated in a cognitive model with which we are able to reproduce and predict human behavior for this task.

1 Introduction

Assume your task is to regulate the level of a water tank whose stock depends on in- and outflow events. Your aim is to maintain a specific water-level (goal) by letting water flow into the tank to increase the level or pumping water out of the tank to decrease the water-level. Additionally, the water-level is influenced by environmental in- and outflows that can not be controlled. The environmental flows are dynamic and rely on unknown mathematical functions.

What are the appropriate actions to choose? Are there differences for the prediction of the underlying mathematical functions? Such research questions have been investigated in the dynamic stocks and flows challenge (DSF-challenge)[1].

Current research so far has mainly covered the manipulation of one function (in- or outflow function, but not both). Thus, many aspects still remain undiscovered: First of all, what happens if there are different functions for in and outflow? A central result is the correlation heuristic [1]. This heuristic claims that individuals tend to choose to copy the value of the environmental inflow value to the user outflow and vice versa. The question is, do participants use this heuristic if the in- and outflow functions are not constant? Or would they rather use a function which is an appoximation of the difference between in- and outflow? Despite its very simplistic representation of this dynamic open-ended task as a waterlevel scenario the hidden function could be any computable function! In other words, it is possible to precisely test which kind of functions (e.g. linear, quadratic, exponential, logarithmic, and so on) humans are able to identify and which can only be approximated. In this sense, such reasoning tasks

[1] http://www.hss.cmu.edu/departments/sds/ddmlab/modeldsf/problem.html

are very similar to analogical tasks and intelligence tests like Raven, where the underlying function has to be identified. The only difference is that in this waterlevel task an immediate response – a feedback – is possible and the reasoner can adapt.

The paper is structured as follows: At first we present an overview of related research to dynamic systems and how humans handle them. Then, we analyze the DSF task in more detail and classify the task according to aspects of artificial intelligence and cognitive science. Afterwards, we briefly present our own conducted studies and present identified (putative) principles in human behavior. Our findings result in a cognitive model for the DSF task which we will present in the last section. Finally, a short discussion concludes the article.

2 Task Analysis

2.1 Related Research

Fundamental research analyzing dynamic systems was done by Forrester. He identified crucial components like accumulation of flows, feedback, and time delays and invented a formal methodology for analyzing and modeling such systems [2]. He first described the difficulties humans have in controlling such systems based on counter-intuitive system behavior.

Several works investigated how humans behave when controlling dynamic systems and revealed that even for apparently easy tasks humans perform poorly. Booth-Sweeney and Sterman conducted a pen-and-paper experiment where the subjects should predict the water-level of a bathtub based on water inflow diagrams. Their results show that humans tend to correlate the system stock with the system flow behavior [3]. Subjects wrongly assumed that the system stock decreases with a decreasing but still positive inflow.

Dutt and Gonzalez analyzed the human strategies in the similar DSF task that was designed to study human dynamic decision making processes [1, 4]. In contrast to the pen-and-paper experiment by Booth-Sweeney et al., the subjects had to maintain a certain water level in a tank. The experiment was conducted using a computer simulation with a graphical representation of the water tank. The water level was influenced by an unknown environmental inflow and outflow as well as subject's inflow and outflow actions. Over a time period of 100 steps the subjects had to reach and maintain a specific level by specifying the amount of water that should flow into and/or out of the tank. The sum of the current amount and the net flow of environmental and user flows results in the amount of water for the next time step. The environmental flow dynamics were controlled by different functions (e.g. linear, non-linear) but so far they only investigated controlling the environmental inflow. One main result of their experiments is that humans have more difficulties in maintaining the goal level if the functions underlying the environmental flows have a negative slope (slope effect) [1, 4]. As mentioned above, they also identified the correlation heuristic, e.g. the strategy to copy environmental inflow values to user outflow actions.

Lebiere et al. established a model comparison challenge for the DSF task using the experiment data of Dutt and Gonzalez. The data is partially provided to the participants to develop a model for simulating human performance on this task. We also participated in this challenge and conducted further experiments to first, get a more reliable dataset and second, to investigate more complex settings (e.g. testing of several function types).

2.2 Task Details

The level of water in the tank is the stock that increases with the inflows and decreases with the outflows. The two types of inflows and outflows in this task have been classified as *exogenous* (outside of the decision makers control) and *endogenous* (under the decision makers control). The exogenous flows in the task are the Environmental Inflow (increasing the level of the stock without the users control) and the Environmental Outflow. The endogenous flows are Users Inflow and Outflow.

The stock level s(t) at time t can be defined as follows [4]:

$$s(t) = s(t-1) + \underbrace{[\phi_i^e(t-1) + \phi_i^u(t-d)]}_{inflow} - \underbrace{[\phi_o^e(t-1) + \phi_o^u(t-d)]}_{outflow}$$

with

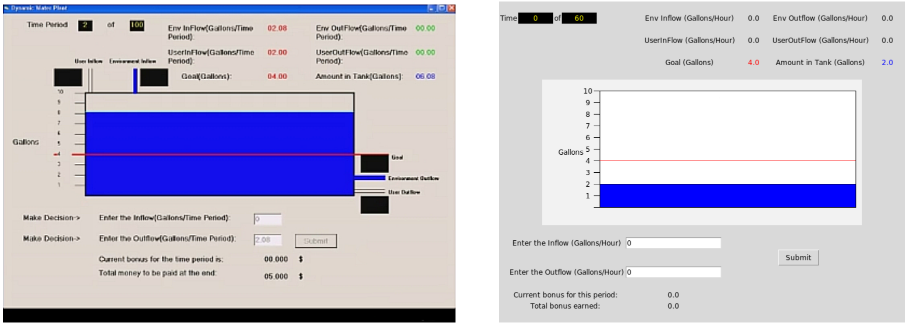
$$\begin{array}{ll} \phi_i^e(t) : \mathbb{N} \to \mathbb{R} & \text{the environmental inflow at time } t, \\ \phi_o^e(t) : \mathbb{N} \to \mathbb{R} & \text{the environmental outflow at time } t, \\ \phi_i^u(t) : \mathbb{N} \to \mathbb{R}^+ & \text{the user inflow at time } t, \\ \phi_o^u(t) : \mathbb{N} \to \mathbb{R}^+ & \text{the user outflow at time } t. \end{array}$$

The parameter d controls the delay of the user action, such that it is not executed directly (default $d = 1$). At each time period users see the values of environmental inflow and outflow, user inflow and outflow, the amount of water in the tank (stock) and the goal level. At each time step, submit their action by specifying values for user inflow and outflow. These inputs can have any positive value, including zero.

The DSF task is performed in discrete time steps and after each step the user can set inflow and outflow values that will be submitted after pressing a submit button (see Fig. 1). There is no time restriction for user decision period. After the user's action, the inflow and outflow values are processed the water-level change is animated. The user is provided with the information of the environmental and the user's flows of the last time steps. Furthermore, the water-level is represented graphically together with a marker line for the goal level.

2.3 Problem Classification

There are several schemata for classifying problem environments according to different abstraction levels. For artificial intelligence (AI) Russel and Norvig classified task environments according to the following properties([5]):



Time Period 2 of 100

Env InFlow(Gallons/Time Period): 02.08

Env OutFlow(Gallons/Time Period): 00.00

UserInFlow(Gallons/Time Period): 02.00

UserOutFlow(Gallons/Time Period): 00.00

Goal(Gallons): 04.00

Amount in Tank[Gallons]: 06.08

Make Decision-> Enter the Inflow(Gallons/Time Period): 0

Make Decision-> Enter the Outflow(Gallons/Time Period): 2.08

Current bonus for the time period is: 00.000 $

Total money to be paid at the end: 05.000 $

Time 0 of 60

Env Inflow (Gallons/Hour) 0.0

Env Outflow (Gallons/Hour) 0.0

UserInFlow (Gallons/Hour) 0.0

UserOutFlow (Gallons/Hour) 0.0

Goal (Gallons) 4.0

Amount in Tank (Gallons) 2.0

Enter the Inflow (Gallons/Hour) 0

Enter the Outflow (Gallons/Hour) 0

Current bonus for this period: 0.0

Total bonus earned: 0.0

(a) (b)

Fig. 1. Comparison of the original GUI used by Dutt and Gonzalez (a) and our simplified version we used in our experiments (b). The interface is divided into three parts. At the top, information about the current time step, the previous environmental flow and the current amount in tank is given. In the middle, a graphical representation of the tank is displayed. The blue canvas indicates the current water-level and the red line displays the goal level. The lower part provides two input fields for the user's inflow and outflow values which are submitted after clicking a button. In our version we left out the repeated display of flow information.

- Accessible vs. Inaccessible: If during reasoning all aspects and information is available then the problem can be called accessible.
- Deterministic vs. Non-deterministic: If the next state is completely determined by the current state and the selected actions.
- Static vs. Dynamic: If the problem structure can change while the reasoner is deliberating then the problem is said to be dynamic; otherwise it is static.
- Discrete vs. Continuous. If there are a limited number of distinct, clearly defined possibilities and actions we say that the environment is discrete.

From this definition it follows that the DSF-tasks are accessible because there are no hidden influences. Since the environmental flows might depend on stochastic processes, the task is nondeterministic. It is a static and continuous environment because the system does not change while the user is not performing any action and the actions consist of real-valued inputs.

In contrast, typical properties of complex problems in human reasoning are characterized by Funke [6]:

- *Cognitive complexity:* measured by the number of involved variables.
- *Interconnectedness:* measured by the high interdependency of variables.
- *Dynamicity:* measured by the temporal changes during the reasoning process (cf. [5]).
- *Polytelie:* (multiple goal availability) measured by the number of sub goals that have to be optimized.

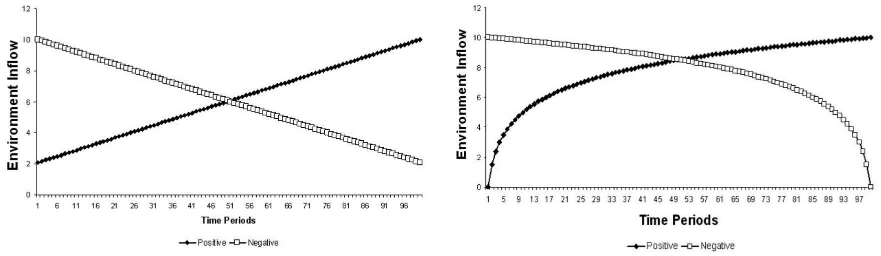
According to Funke, interconnectedness and dynamicity are the main characteristics for complex problems [7]. Here the dynamicity qualifies this problem as a complex problem. An integrated approach is from Quesada et al. [8] which included in their taxonomy a combination of formal (which are similar to classifications in AI, e.g. [5])) and psychological descriptions. The main criteria for the inclusion are:

- skill-based vs. planning-based (the first one requires a more reactive behavior while the second one allows for a predictive behavior)
- Knowledge-lean vs. knowledge-intensive problems
- Learning vs. non learning during problem-solving
- Understanding-based vs. search-based problems
- Ill-defined vs. well-defined problems.

Although further training might increase the performance, the obviously well-defined DSF task is easy to understand without any previous knowledge. It is mainly a skill-based task since for complex environmental dynamics it is hard for the user to predict the successive system state. During the task, the subject has to figure out the underlying environmental changes to be able to maintain the goal-level. Thus, the task is knowledge-lean and requires learning. It is an understanding-based problem, since a search-based approach is not reasonable in this continuous domain.

2.4 Experiment Setting

As mentioned above, the organizers of the DSF Challenge provided partial experiment data^2 from Dutt and Gonzalez. Four different datasets were given, each for a different underlying environmental inflow function (see Fig. 2). So far, different aspects remain unknown: First of all, what happens if different functions



(a) linear inflow

(b) non-linear inflow

Fig. 2. Functions underlying the environmental dynamics used by Dutt and Gonzalez [4, 1]. The outflow was constant zero in all cases.

[2] The experimental data can be retrieved from: http://www.hss.cmu.edu/departments/sds/ddmlab/modeldsf/data.html

are used for environmental inflow and outflow? Would participants use the correlation heuristic? Or would they rather use a function which is an appoximation of the difference between in- and outflow?

Since humans are not good at identifying logarithmic functions (i.e. the non-linear functions in the experimental data) – how do they behave if they get quadratic functions, gauss-curves, sigmoid functions and constant functions? To answer these questions, we reproduced the experiment by Dutt and Gonzalez and tested further function types and combined the use of inflow and outflow functions which not has been done so far.

Methods, Participants, Procedure. Our experiment was conducted similarly to the original study by [1]. In addition to the originally used functions, we tested the following environmental functions:

Task 1:	$\phi_i^e(t+1) = 0.08 \cdot t^2$	$\phi_o^e(t+1) = 0.08 \cdot t$
Task 2:	$\phi_i^e(t+1) = 0.08 \cdot t + 2$	$\phi_o^e(t+1) = 2$
Task 3:	$\phi_i^e(t+1) = 0.08 \cdot (30 - t) + 2$	$\phi_o^e(t+1) = 2$
Task 4:	$\phi_i^e(t+1) = \frac{10}{8 \cdot \sqrt{2\pi}} \cdot \exp(-0.5 \cdot ((t-15)/8)^2)$	$\phi_o^e(t+1) = exp(1)$

We asked the participants to attempt to control the system over the course of 30 time steps. We choose 30 instead of 100 since in all previous studies (cf.[1, 4]) nearly all participants had already adapted after 20 moves to the function of the task. The participants were presented with our own GUI (see Fig. 1). The goal was to maintain the water tank level at 4 Gallons, beginning with an initial level of 2 Gallons. 32 participants were tested in this experiment and received course credits for their participation. We randomly assigned all 32 students to one of the above tasks. None of the participants received any information about the kind of function.

2.5 Experiment Findings

Analyzing the provided datasets reveals several issues:

The graphical representation plays an important role for human behavior: The original GUI (left in Fig. 1) has a a higher amount of visual information and animated features. We used a simplified GUI (cp. Fig. 1) to reduce visual overload without lack of information. The participants in our experiments could adapt faster to the functions than in the experiments by Dutt and Gonzalez. But we did not conduct control experiments to directly compare the two visualizatons.

It is difficult for humans to deal with non-linear functions and they are worse at decreasing than increasing functions: While each participant was easily able to adapt to the linear increase function the fluctuation was much higher with the non-linear and decreasing functions. This result also has been reported by [4].

Heuristics are used in the beginning: It seems to be the case that in the first steps participants used a means end analysis [9] between amount in tank and goal (neglecting the environmental in- or outflow). However, after these steps they use a refined heuristic and approximate the net flow to compute the next action.

Adaption range: Nearly all participants reach the goal interval[3] within 20 steps.

Rounding: The participants in general tend to use rounded values. They tend to use more precise predictions as they get closer to the goal.

Memory: It is not possible to make exact predictions without storing current information in the working memory (allowing for additions and subtractions to identify the different values).

A full presentation of the statistical analysis would go beyond the scope of the paper. We have found like [1] that on average the Stock differed between positive and negative environmental of the combined function (inflow-outflow function). The difference between amount in tank and goal was much smaller in the first case. Using this measure participants using Task 1 could adapt much better to the underlying function than participants in Task 4, while Task 2 and 3 were in-between. All but one participant specified either inflow or outflow values but not both. This means, the participants have computed the total function.

We could reproduce the effect that participants are better in dealing with increasing functions. We believe it depends on the relatively high values with which the decreasing functions start (10). This leads to a high deviation from the goal in the beginning which has to be corrected. This must be, however, validated in additional experiments. An astonishing fact is that only one participant (out of 32) has shown the usage of a correlation heuristic (i.e. setting $\phi_i^e(t) = \phi_o^u(t+1)$ and $\phi_o^e(t) = \phi_i^u(t+1)$).

3 Our Model for the DSF task

3.1 Requirements of a cognitive model

Our aim was to model the user's behavior in the DSF task. The empirical adequacy of computational models can be evaluated by comparing the observable trace of subjects behaviors when performing a task with the performance or trajectory of a model [10]:

- *Product correspondence*: Similarity of the final performances (such as success in solving a problem or classes of problems) on a specified scale.
- *Correspondence of intermediate steps toward problem solution* e.g. w.r.t. problem solving strategy
- *Temporal correspondence:* Similar latencies between participants and model
- *Error correspondence:* Comparability of the numbers and kinds of errors
- *Learning correspondence:* Comparability in rate of improvement of performance with practice in the same learning environments.

[3] 4.0 ± 0.1

3.2 Model Design

Fig. 3 gives a complete overview about how the model works. Starting with no information about the environment and the system dynamics, our model agent performs the most rational action as if acting in a non-dynamic environment. That means, it counterbalances the difference between goal and current amount by the missing or surplus amount. The model agent retains this policy, that we call *rule-of-thumb* (ROT), for a certain amount of steps controlled by the parameter t_f. Also, the ROT takes into account that with further steps the actions result in more exact amounts. We therefore compute a coarse prediction of the successive effective environmental inflow at time step t. The effective environmental inflow at time t is defined as

$$\phi_i(t) = \phi_i^e(t) - \phi_o^e(t)$$

For the next step, this value is approximated by

$$\phi_r(t+1) = sigm(t) \cdot \phi_i(t)$$

where

$$sigm(x) = \frac{1}{1 + e^{-c(x-s)}}.$$

The value for the counterbalancing action results in

$$v = s(t) - g + \phi_r(t+1)$$

such that

$$(\phi_i^u, \phi_o^u) = \begin{cases} (v, 0) & \text{if } v < 0 \\ (0, v) & \text{otherwise.} \end{cases}$$

Table 1. Model parameters.

parameter	description
δ_a	Ascend tolerance until underlying function is treated as linear.
t_f	Time steps until *rule-of-thumb* is used.
ϵ_d	Discrepancy when ascend will be re-estimated.
$\delta_1, \delta_2, \delta_3$	Discrepancies for different rounding precisions.
γ_e	Coefficient for extreme amount correction
s	Shift parameter for the sigmoid function.
c	Compression parameter for the sigmoid function.
γ_a	Coefficient for ascend deviation.

The parameters c and s modulate the steepness and the shift of the sigmoid function to adjust the learning progression, i.e. the increasing accuracy of the prediction. Since the actions depend on previous system states, the model needs some kind of memory to store perceived environmental attributes. In our model, only the last two states are accessible to simulate the limited human working memory capacity.

After t_f steps, the model agent stops using the ROT and starts to guess the type of the function underlying the changes of the environmental influences. Our main assumption is that it is difficult for humans to handle influences based on non-linear functions. Instead, they try to approximate them with linear functions based on the previous observations. The function for environmental influences is computed by comparing the ascend of ϕ between the last two state transitions. If the ascends are similar up to a small deviation (parameter δ_a) , these cases are handled as linear increase and decrease respectively. In all other cases, the underlying function is treated as non-linear without further function interpolation. Despite the used linearization principle is the same for all function types, the distinction is necessary because experimental results show that subjects acted differently for the several cases. In our model we therefore use different parameter settings to adjust the state prediction computations. When the function was guessed, the ascend a of the approximated linear function is memorized and used for the next predictions until the action leads to a deviation greater than a tolerance limit (ϵ_d). Then, the ascend for the linear function approximation will be re-estimated. The forecast inflow is computed by

$$\phi_f(t+1) = a \cdot \gamma_a + \phi_e(t)$$

such that the value that has to be counterbalanced is

$$v = s(t) - g + \phi_f(t+1).$$

The resulting action then results in

$$(\phi_i^u, \phi_o^u) = \begin{cases} (v, 0) & \text{if } v < 0 \\ (0, v) & \text{otherwise.} \end{cases}$$

A special case occurs when the amount in tank exceeds the visible limit above ten and below zero[4] gallons. This happens mostly in situations where the environmental influences change in a non-linear fashion. As mentioned above, humans tend to rely rather on the visual information than the exact absolute value. Experimental results show that in these situations the subjects react with strong correcting actions to return to a fill level between 0 and 10. In such cases, our model attempts to balance back the fill level by the discrepancy between goal and current amount multiplied by a constant real value we refer to as *extreme amount coefficient* (γ_e). The agent remains in this correcting mode as long as the amount in tank is out of scale. When reaching a visible fill level again, it switches back to function guessing.

The values the model agent has to handle are always rounded according to the discrepancy between goal amount and amount in tank. This reproduces

[4] We assume this is a bug in the DSF implementation but still, we considered this case since experiment data was collected with this program.

the subject's behavior to not calculate with exact values in cases of a high discrepancy. We integrated three rounding methods into the model. When the discrepancy is higher than δ_1, the values are rounded to the next closest integer or .5 value. For lower discrepancies we also distinguished the case for rounding to two (δ_2)and three (δ_3) decimal places.

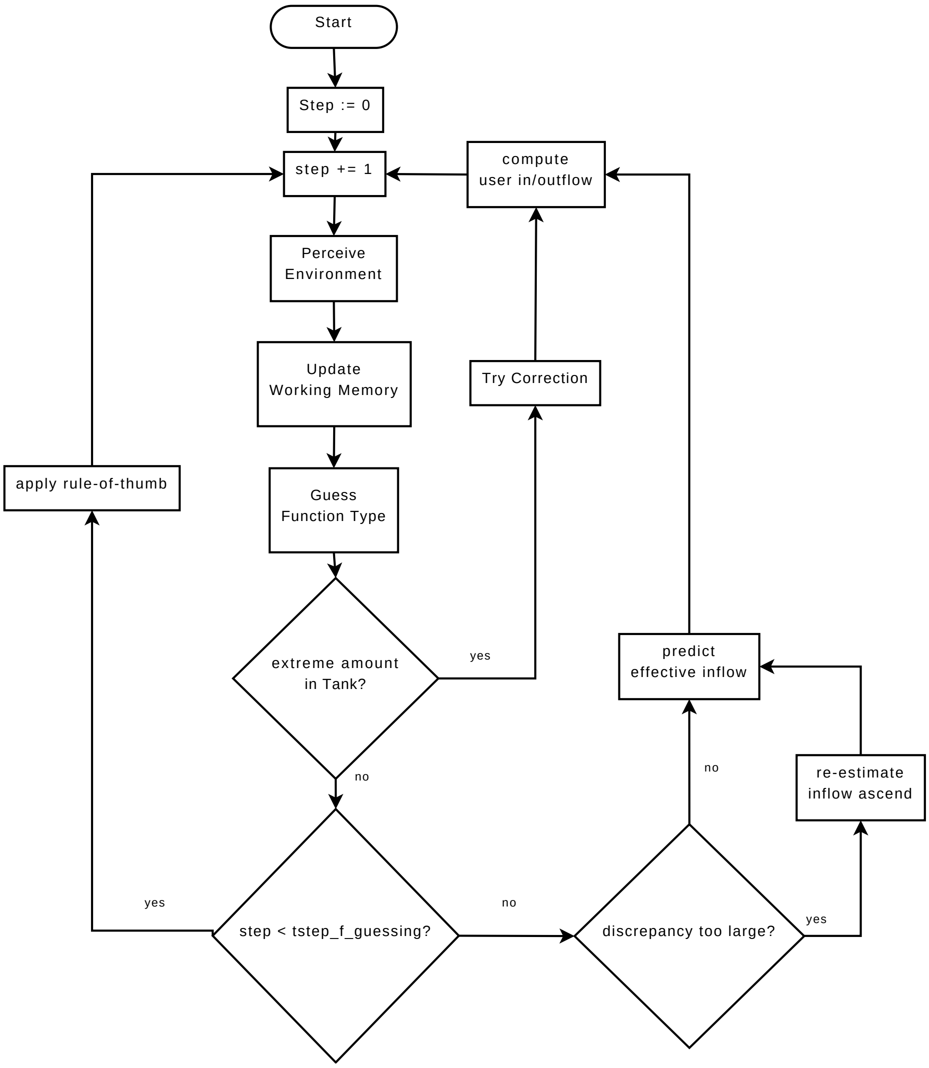


Fig. 3. Flowchart of the model.

3.3 Model implementation

We have implemented our model in Python[5]. The central part is the Operator class which is designed as a classical agent that perceives and acts [5]. The general and function type specific parameters were stored outside the operator class in an associative array. We narrowed the parameter boundaries in advance and optimized them by a local search algorithm to best fit the experiment data.

In each step the operator perceives the environmental state and updates its internal working memory (perceive). Afterwards, it computes the counterbalancing action for the current state (act). As mentioned above, our main approach is the idea that the participants use (for a certain number of steps) a rule of thumb. Then, they approximate the function by linearizations (with respect to some error rates). These strategies are implemented by the functions apply_ROT() and predictFlow(). In the function predictFlow() the operator computes the forecast and appropriate action. In contrast to the rule-of-thumb, the predictFlow() function predicts the forecast value more accurately.

3.4 Model evaluation

For evaluation we first trained our model with the given datasets from Dutt and Gonzalez and also with the data collected from our own experiments. The trained model was tested against these datasets and we could reproduce the behavior of average users. The model was also tested by the organizers of the DSF challenge against unknown datasets, whereby our model achieved a high accuracy. This means that our model reproduces and predicts human behavior even in cases for which the model was not trained for. The heuristics identified in the experiments are embedded into the system which enables the model to replicate sub-optimal[6] human behavior.

4 Conclusion and Discussion

Since the underlying function of a DSF-task can be any computable function, this problem can be considered as highly computationally complex. Noise and delayed actions of the reasoner's response pose additional difficulty on the human reasoning and planning process. To capture the complexity of these kind of dynamic problems we classified this task with respect to AI problem and cognitive science characteristics [6].

Analyzing the classical DSF-task reveals that the original representation from [4] had a high visual complexity factor. We therefore reimplemented the GUI and reconducted our own experiments. We could not verify the so-called correlation heuristic, i.e. that the output should be similar to the input [11]. This heuristic can best be tested by combining different in- and outflow functions. Here, the main question is: do participants simply "carry" the environmental outflow over

[5] http://www.python.org

[6] From a computational perspective.

to the user inflow (analogously for the environmental inflow and user outflow)? Or do they calculate the difference between environmental inflow and outflow?

We have put these questions to the test using four different functions. Our experimental results could not support the "correlation" heuristic. Taken together, the correlation heuristic might be due to the higher visual complexity of the original study and the simpler functions they had tested.

Our intention was to develop a good (adhering to the principles of [10]) cognitive model for the DSF-Competition. Thus, our main assumption is that humans are not very good at dealing with any kind of non-linear functions. If they have to deal with such kind of functions they start to approximate the original function by a linearization (i.e. use tangents) as long as possible and update this function if the utility decreases. A good utility measure is the difference between the goal and the current amount in the tank. Further parameters (cf. Table 1) became necessary to integrate into the cognitive model to approximate human behavior in learning the functions underlying the environmental changes.

Future work must cover the investigation of additional functions and the control of several water tanks which have an influence on each other.

References

1. Dutt, V., Gonzalez, C.: Slope of inflow impacts dynamic decision making. In: Proceedings of the 25th International Conference of the System Dynamics Society, Boston MA: System Dynamics Society (10(3) 2007) 79
2. Forrester, J.W.: Industrial dynamics: A major breakthrough for decision makers. Harvard Business Review **36**(4) (1958) 37–66
3. Booth-Sweeney, L., Sterman, J.: Bathtub dynamics : initial results of a systems thinking inventory. Working papers WP 4132-00., Massachusetts Institute of Technology (MIT), Sloan School of Management (April 2000)
4. Gonzalez, C., Dutt, V.: Learning to control a dynamic task: A system dynamics cognitive model of the slope effect. In 8th International Conference on Cognitive Modeling, ed.: Lewis and Polk and Laird (Eds.), Ann Arbor, MI (2007) 61–66
5. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd edn. Prentice Hall (2003)
6. Funke, J.: Komplexes Problemlösen. In Funke, J., ed.: Denken und Problemlösen. Enzyklopädie der Psychologie. Volume 8. Hogrefe, Göttingen (2006)
7. Funke, J.: Dynamic systems as tools for analysing human judgement. Thinking and Reasoning **7** (2001) 69–89
8. Quesada, J., Kintsch, W., Gomez, E.: Complex problem solving: A field in search of a definition? Theoretical Issues in Ergonomic Science (2005) 5–33
9. Anderson, J.R.: Kognitive Psychologie. Spektrum Akademischer Verlag, Heidelberg (2001)
10. Simon, H.A., Wallach, D.: Cognitive modeling in perspective. Kognitionswissenschaft **8(1)** (1999) 1–4
11. Cronin, M., Gonzalez, C., Sterman, J.D.: Why don't well-educated adults understand accumulation? a challenge to researchers, educators and citizens. Organizational Behavior and Human Decision Processes **23(1)** (2009) 108

On Optimization of the Interface between Subsymbolic and Symbolic Representations and the Symbol Grounding Perspective

Oliver Kramer

Department of Computer Science
Technische Universität Dortmund
44227 Dortmund, Germany
oliver.kramer@tu-dortmund.de

Abstract. From the point of view of an autonomous agent the world consists of high-dimensional dynamic sensorimotor data. Translated into symbols the data is easier to handle for cognitive processes. I propose to formulate the interface design between subsymbolic and symbolic representations as global optimization problem. The objective becomes to maximize the success of the overlying cognitive algorithm. For implementation of the interface, various known algorithms from data mining and machine learning turn out to be adequate choices that exploit intrinsic structures of the data space and allow a flexible adaptation of the interface at the same time. From the point of view of the symbol grounding problem the meaning of a symbol arises implicitly from this optimization formulation and fulfills the zero semantical commitment condition.

1 Introduction

In natural systems the transition of subsymbolic data into symbols is known as perception. Auditory, visual or tactile data from various kinds of sensors are reduced and subject to neural pattern recognition processes. The human visual system is an excellent example for such a complex transformation. Also in the design of artificial intelligent systems many examples exist where subsymbolic data is manipulated by means of arithmetic operations and transferred to a symbolic level. These high-level symbols represent concepts that are comparable to mental models in cognitive science. They allow cognitive manipulations like inference processes and logical operations: thinking can be referred to as manipulation of symbols, similar to computation [8]. Cognition becomes implementation-independent, systematically interpretable symbol-manipulation.

The questions arises how to define symbols and their meanings for artificial systems, e.g. for artificial autonomous agents. Which subsymbolic elements belong to the set that defines a symbol, and – with regard to cognitive manipulations – what is the interpretation of this symbol? These questions are closely related to the symbolic grounding problem that has been introduced by Harnad [7, 9] with the Chinese Room Argument (see Searle [16]). How is the meaning and the interpretation of a symbol grounded? To answer this question a couple of strategies have been proposed that are reviewed by Taddeo *et al.* [19] – many of them more or less successful.

To my mind the definition of a symbol and its interpretation is mostly of functional nature. The intention and the success in solving problems to achieve goals must guide the meaning and thus the definition of symbols. Hence, it seems reasonable to formulate the symbol definition as optimization problem. Optimal symbols and their interpretations yield optimal success of an autonomous agent. In many artificial systems symbols are defined by an interface algorithm that maps sensory or sensorimotor data to symbol tokens, e.g. class labels. Optimizing a symbol with regard to the success of cognitive operations means optimizing the interface design. In many artificial systems the interface design is part of an implicit system modeling process – regrettably often without much effort spent on an optimal architecture.

The paper is structured according to three perspectives it introduces. First, the A *formal* perspective in Section 2 will formulate the interface design as global optimization problem. The concepts of symbols and higher cognitive operations are formalized. The interface between subsymbolic and symbolic representations is introduced in an optimization formulation while potential objectives, free parameters and a two-level optimization process are discussed. An *algorithmic* perspective is shown in Section 3 where I discuss typical data mining and machine learning tasks like classification, clustering and dimensionality reduction in the context of interface design and symbol grounding. I propose not to restrict to connectionist approaches, but to make use of recent data mining and machine learning techniques – from k-means to kernel methods. Last, the *cognitive* perspective of Section 4 discusses the consequences of the interface optimization formulation on the symbolic grounding problem. To my mind – as only the agent's objective has to be formulated explicitly, and this is implicit to any biological form of life[1], the optimization formulation is close to fulfilling the zero semantical commitment condition.

2 Interface Design as Optimization Problem

Cognitive operations operate on a symbolic level. After the characterization of symbolic algorithms, I formulate the definition of a symbol via its connection to subsymbolic representations. An interface algorithm maps the subsymbolic data onto symbols. With regard to the objectives of the cognitive system the interface design is formulated as global optimization problem.

2.1 Symbolic Algorithms

The definition of higher cognitive operations of autonomous agents is no easy undertaking and faces similar problems like the definition of *intelligence* in cognitive sciences and psychology [15]. Most of the higher cognitive operations involve

[1] survival, reproduction, and each objective that is connected to the first two ones

the perception of sensorial information. Spatial reasoning involves visual perception, while the usage of language involves auditory perception. Hence, higher cognitive operations include an appropriate interface I *and* algorithmic operations on the higher level, so called symbolic algorithms. Because of the difficulties we face with regard to a definition of what intelligent algorithms are, one can characterize symbolic operations by giving examples, e.g. deduction processes in propositional logic, or evolvement, understanding and usage of language, as well as spatial reasoning. This characterization takes into account what one can only loosely define as "more sophisticated" intelligence. In general, most algorithms from classic artificial intelligence – from depth-first search to planning and reasoning – belong to the class of symbolic algorithms. In most cases – and this is frequently claimed to be important – a cognitive system is situated into a real environment, this is denoted as *embodied intelligence*, see [13, 20].

In the following, I assume that an autonomous agent performs cognitive operations with a symbolic algorithm, i.e. an algorithm that operates on the level of symbols.

Definition 1 (Symbolic Algorithm). *A symbolic algorithm $\mathcal{A}$ performs (cognitive) operations on a set of symbols $\mathcal{S}$.*

If possible, we measure the success of the operations by a quality measure $f_{\mathcal{A}}$.

2.2 Symbols and Interfaces

An interface algorithm I perform a mapping $I : \mathcal{D} \rightarrow \mathcal{S}$ to the set of symbols $\mathcal{S}$. The result is a cognitive symbol $c \in \mathcal{C}$, and $\mathcal{C}$ is the set of cognitive symbols, that we only define for the sake of a better understanding. A cognitive symbol c comprises a subset of subsymbolic data samples $\mathcal{D}_c \subset \mathcal{D}$ that are assigned to a certain symbol shape $s \in \mathcal{S}$.

Definition 2 (Cognitive Symbol). *A cognitive symbol c consists of a data set $\mathcal{D}_c \subset \mathcal{D}$ of subsymbolic data and a corresponding symbol $s \in \mathcal{S}$ that is subject to cognitive manipulations.*

A cognitive symbol is the basis of many approaches in artificial intelligence – although not always explicitly stated. The definition of this concept helps to become aware of the importance of a well-designed interface I. In case of a self-organizing map, see Paragraph 3.2, a cognitive symbol is a Voronoi cell in data space and its corresponding winner neuron n^*, i.e. the corresponding symbol s. The meaning of a symbol $s \in \mathcal{S}$ is based on its interpretation on the symbolic level. On the one hand symbols are only tokens – and may be defined shape-independent [8]. But, the effect they have on the symbolic algorithm $\mathcal{A}$ can be referred as *meaning* or *interpretation* of the symbol. I assume that in autonomous agents an interface I exists that maps the subsymbolic data from a high-dimensional set $\mathcal{D}$ onto the set of symbols $\mathcal{S}$.

Definition 3 (Interface from Subsymbolic to Symbolic Representations). *The interface from subsymbolic to symbolic representations $I : \mathcal{D} \rightarrow \mathcal{S}$ maps each data sample $d \in \mathcal{D}$ to a symbol $s \in \mathcal{S}$.*

I defines the set of symbols and may be implemented by any interface algorithm. From the perspective of cognitive economy, it makes sense that $|\mathcal{S}| << |\mathcal{D}|$, and that subsets of $\mathcal{D}$ are mapped to $\mathcal{S}$, i.e. $I : \mathcal{P}(\mathcal{D}) \rightarrow \mathcal{S}$. In Section 3, I propose interface algorithms from data mining and machine learning. These interfaces exploit the intrinsic structure of data space, but most are parameterized and thus can be subject to optimization with regard to certain properties. A bias of the search problem is necessary since the learning of representation and training of the learning algorithm is a hopeless undertaking due to an exponential increase of the search space as already Mayo [12] states. The exploitation of the intrinsic structure of the subsymbolic data space is such a bias, and to my mind the most adequate. Interface design concerns the choice of a proper interface algorithm, appropriate parameterizations and also the choice of adequate features. Note, that feature selection – a very successful technique to reduce the solution space – is also a special case of interface optimization.

2.3 Interface Optimization

Now, we formulate the design of interface I as optimization problem: we want to find the optimal mapping $I^* : \mathcal{D} \rightarrow \mathcal{S}$ with regard to the success $f_{\mathcal{A}}$ of the symbolic algorithm $\mathcal{A}$.

Definition 4 (Interface Optimization). *The optimal interface I^* maximizes the success $f_{\mathcal{A}}$, i.e. $I^* = \arg\max_I \{f_{\mathcal{A}}(I) | I \in \mathcal{I}\}$.*

For this optimization formulation we have to define a quality measure $f_{\mathcal{A}}$ with regard to the symbolic algorithm A. The set of interfaces $\mathcal{I}$ may consists of the same algorithm with various parameterizations, e.g. k-means with different numbers of clusters k. The optimization problem may be solved offline, i.e. the systems runs until a termination condition is met. Afterwards, the feedback $f_{\mathcal{A}}$ from the symbolic algorithm is sent to the optimizer. The optimizer chooses an interface variant or a new parameterization and so forth. If feedback $f_{\mathcal{A}}$ allows, an online-adaptation of the interface is another promising possibility. In this scenario the optimizer adapts the interface during runtime of the system – this is only possible if the feedback is available online. In the following, we will discuss typical free parameters and possible feedback for the interface optimization process.

In practice the model constructor does not spend effort into the explicit design of the interface between subsymbolic and symbolic representations. It is frequently an implicit result of the modeling process. Classification algorithms are applied taking into account the learning signal that a separate source delivered as class label. Clustering algorithms find the structure in the data with regard to special properties, e.g. data density. Most system designers rely on the correctness of the class label or on the abilities of the clustering algorithms not concentrating on the requirements of the symbolic algorithm. The definition as optimization problems helps to get aware that the design is important and to make the application of optimization techniques more obvious,

2.4 Optimization Criteria

For the adaptation of an optimal interface I a clear optimization objective has to be specified. The main objective is to map high-dimensional sensory data to a *meaningful* set of symbols (of arbitrary shape). How can this mapping be measured in terms of a feedback $f_{\mathcal{A}}$ from the symbolic algorithm? The feedback depends on the goal of the autonomous agent. If it can explicitly be expressed by a measure $f_{\mathcal{A}}$, an optimization algorithm is able evolve the interface. In general, we see the following scenarios to get the feedback of the symbolic algorithm, of which only the first two fulfill the zero semantical commitment condition of the symbolic grounding problem – a definition and discussion of the symbol grounding aspects will follow in Section 4.

1. *Offline-feedback response.* In the offline approach the symbolic algorithm runs for a defined time, e.g. until a termination condition is met, and propagates feedback $f_{\mathcal{A}}$ that reflects its success back to the optimization algorithm. If interface design is the only optimization objective – see Paragraph 2.5 for thoughts about a two-level optimization process that considers learning on the symbolic level as well – the system will adapt the interface to achieve a maximal response. This process might be quite slow if the symbolic algorithm is supposed to run for a long time to yield $f_{\mathcal{A}}$.
2. *Online-feedback response.* If the symbolic algorithm delivers the feedback $f_{\mathcal{A}}$ during runtime, this feedback can be used to define symbols online. For example in a reinforcement learning scenario where artificial agents have to learn from rewards in uncertain and dynamic environments, the temporal information of the rewards can guide the interface process online. If an agent is in a place of the environment where many varying rewards are available the online-feedback response might lead to a more granular resolution of states in comparison to places where no feedback is available.
3. *Data driven.* The intrinsic structure of the sensorial data itself usually guides clustering approaches and might also be important as strategy to ground the meaning of symbols, e.g. to differentiate between concepts that do not belong to the same cluster. With regard to its intrinsic structure, clustering yields a reasonable discretization into meaningful symbols in these cases.
4. *User-driven.* The practitioner should include as much knowledge as possible into the optimization process. The knowledge can guide the response $f_{\mathcal{A}}$ manually. But the user can also integrate his knowledge in form of constraints for the optimization problems, e.g. defining the number of symbols a priori.

2.5 A Two-Level Learning Problem

The symbol grounding problem is not the only problem an autonomous agent has to learn. More effort is usually spent on learning of strategies and behavior and the symbol definition remains an often neglected subproblem. An appropriate approach would be to treat interface optimization and learning of strategies as a two-level learning problem. One approach would be to optimize both levels alternately: With a fixed set of symbols the learning problem can be optimized, with a fixed learning strategy the set of symbols can be optimized. Whether interface optimization and learning can be solved in parallel surely depends on whether the increase of the search space does not make the whole optimization problem unsolvable. The two-level learning problem is related to the two-level mapping from the subsymbolic level to the meanings of symbols that I will discuss in Section 4.

3 Machine Learning Interfaces

The translation of high-dimensional subsymbolic data into symbols are tasks that are well known in data mining and machine learning under the terms classification, clustering and dimensionality reduction. Many symbol grounding related work exclusively concentrated on neural networks in the past [6, 7, 2, 18], perhaps due to a historical affinity to connectionist models. To overcome the restriction this Section shows the relation between symbol grounding and machine learning: assigning unknown objects to known concepts is known as *classification*, grouping objects is known as *clustering*, finding low dimensional representations for high-dimensional data is denoted as *dimension reduction*.

3.1 How are Machine Learning Algorithms Related to Symbol Grounding?

The problem of iconization, discrimination and identification formulated by Harnad [6] is closely related to the question how to map high-dimensional data to classes or clusters. Classification, clustering and dimensionality reduction are similar in this context. They perform a mapping from a high-dimensional data space $\mathcal{D}$ to a low dimensional set of symbols $\mathcal{S}$ that may be a class, a cluster, or a low dimensional manifold. The three machine learning tasks implement the nature of dimensionality reduction as follows: Classification algorithms deliver a subsymbolic to symbolic mapping $I : \mathcal{D} \rightarrow \mathcal{S}$ with regard to explicitly labeled data samples in a supervised way. In a training phase mapping I is learned by reducing the classification error. A learned interface is used to classify unknown data, i.e. assign symbols to classes of similar high-dimensional input data. Clustering algorithms deliver the subsymbolic to symbolic mapping $\mathcal{D}$ to a set of clusters $\mathcal{S}$ with regard to the intrinsic structure of the subsymbolic data and the properties of the algorithm in an unsupervised way. Frequently, the dimensionality of observed data is much higher then the intrinsic dimensionality. A 3D-object for example has got an intrinsic dimensionality of 3, but on a digital image the dimensionality of the data vector is much higher depending on the resolution of the picture. Last, dimension reduction methods have a similar task like classification and clustering. For high-level data low-level representation have to be found, e.g. a mapping from subsymbolic to symbolic data $I : \mathcal{D} \rightarrow \mathcal{S}$ or the mapping from $\mathbb{R}^m \rightarrow \mathbb{R}^n$ with $m > n$. I come to the conclusion that classification, clustering and dimensionality reduction algorithms from machine learning are eligible algorithms for the interface I from subsymbolic to symbolic representations.

3.2 Examples for Related Machine Learning Algorithms

In the last years kernel methods became quite popular in machine learning and data mining. It is not the scope of this work to review these methods. For a detailed introduction I refer to textbooks like Bishop's [1] or Hastie's [10]. Here, I only comment on the properties of three possible interface algorithms with regard to the interface problem.

A simple but successful clustering technique is k-means clustering [1]. K-means needs one essential parameter: the number of clusters k – that we denote as number of symbols. Each cluster $C_j \in \mathcal{S}$ with $1 \leq j \leq \ k$ can be described by its cluster center c_j, the barycenter of the cluster elements. This concept shows that both clustering and cognition share similar ideas: If the distances between the elements in the data space and the cluster centers are minimal, then clusters of elements should be represented by the same center whilst far-out accumulations of elements belong to different centers. This principle is similar to the idea of semantic distances of mental models. K-Means work as follows. At the beginning it randomly generates k initial cluster centers c_j. In order to minimize the sum of distances, k-means works iteratively in two steps. In the first step each data element x_i is assigned to the cluster C_j with minimal distance. In the next step k-means computes the new cluster centers c_j as average of the data elements that belong to C_j. K-means continues with the cluster center computation, and so forth. The algorithm ends if the cluster assignment does not change or if the change falls below a threshold value ϵ. The process converges, but may get stuck in local optima. K-means allows to specify the number of clusters. If we use k-means as interface algorithm, we can treat k as free parameter that can be optimized with regard to $f_{\mathcal{A}}$. The optimal number of symbols to solve cognitive tasks may frequently not be known in advance. Perspectives are the number of states in reinforcement learning scenarios or the number of words in language learning scenarios. But also other clustering algorithms may be applied, e.g distance based approaches like DBSCAN that are based on the distances between the data samples [4].

In comparison to clustering algorithms, most dimensionality reduction algorithms maintain the structure of the data space, e.g. neighbored data samples in data space are neighbored on a low-dimensional manifold. A recommendable example is the self-organizing map by Kohonen [11]. Its number of neurons and the learning parameters are eligible free parameters for optimization. In each generation the self-organizing map updates the weights $\mathbf{w}$ of a winner neuron and its neighborhood with the help of learning parameters η and a neighborhood parameter h, so that they are pulled into the direction of data sample $\mathbf{x}$ The algorithms lead to a mapping from the feature space $\mathcal{D}$ to the map. The mapping maintains the topology of the neighborhood: Close data samples in the high-dimensional space lie close together on the map. Whether this property is important for the interface depends on the interpretation of the symbols.

3.3 Optimization Algorithms

When the optimization objectives are clearly specified, and a feedback $f_{\mathcal{A}}$ of a given interface I is available, the choice of an adequate optimization algorithm has to be answered. If no more information is available than the feedback, i.e. no explicitly given functions nor derivatives, we recommend to apply evolutionary algorithms. A comprehensive survey of evolutionary algorithms is given by Eiben [3]. Evolutionary computation comprises stochastic methods for global optimization, i.e. optimization problems with multiple local optima. They are biologically inspired and imitate principles that can be observed in natural evolution like mutation, crossover and selection. If the optimization problem is not supposed to suffer from multiple local optima, deterministic direct search methods like Powell's conjugate gradient algorithm [14] or similar optimization algorithms for convex optimization may be applied.

4 Perspective of the Symbol Grounding Problem

Now, I discuss the interface optimization problem from the perspective of the symbol grounding problem. After its short introduction, I describe the implicit evolvement of symbol meaning. Guided by seven features of a valid solution to the symbol grounding problem I will discuss the optimization formulation as valid solution of the symbol grounding problem.

4.1 Symbol Grounding

Harnad [7] argues that symbols are bound to a meaning independent of their shape. The symbol grounding problem is the problem to answer how words get assigned to meanings and what meaning is. Floridi [5] emphasizes the importance of the symbol grounding problem as an important question in the philosophy of information. The symbol grounding problem has been intensively discusses within the last decades [6, 7]. Taddeo *et al.* [19] reviews various approaches to the symbol grounding problem classing them into approaches based on representationalism, semi-representationalism and non-representationalism.

4.2 Implicit Evolvement of Meaning

From the perspective of the symbol grounding problem the optimization formulation yields valuable insights. Learning and cognitive information processing become a two-level mapping, first from the space of subsymbolic data $\mathcal{D}$ to the space of symbols $\mathcal{S}$, second, from there to the meaning of symbols. Their semantics are implicit bound to the cognitive process[2] whose success guides the optimization process. During interface design, the first part of the mapping is subject to optimization while the second parts guides this optimization process. The whole process yields a grounding of symbols – arbitrary of their shape, but

[2] above indicated as symbolic algorithm $\mathcal{A}$

based on objectives on the functional level of semantics. Decontextualization, i.e. to abstract from particular patterns and the ability of a symbol to function in different contexts is less an interface design, but a problem on the symbolic level, see condition 4 of Paragraph 4.4.

4.3 Related Approaches

Harnad [6] proposes three stages of grounding, i.e. iconization, discrimination and identification – that comply with the tasks that data mining algorithms solve, see Section 3. In the taxonomy of symbol grounding approaches the idea to bind the representation to functional or intentional properties can be found in the representational approaches. Mayo *et al.* [12] proposes a functional organization of the representations and introduces task-specific categories where symbols are formed in order to solve task-related problems. They introduce a bias to put sensory data into a category that best contributes to the solution of a particular problem.

Sun's [17] intentional model is also related to the optimization view. He introduces a two-level approach whose first level concerns behavior guided by the external world and innate bias. Conceptual representations are learned on the second level. On the first level the autonomous agent explores the world by trial-and-error bound to its objectives. On the second level the first level intentional data is used to evaluate courses of action to achieve objectives. The top down process that guides selection of actions is similar to the top-down interface optimization feedback principle introduced in this work.

4.4 A Valid Solution to the Symbol Grounding Problem?

On page 33 of their work Taddeo *et al.* [19] postulate seven features a valid solution of the symbol grounding problem needs. We will shortly discuss the interface optimization problem in the context of these seven properties citing the postulates.

1. The optimization approach is a "bottom-up, sensorimotor approach" as subsymbolic sensorimotor data is mapped from the bottom to the level of symbols and from there implicitly to their meanings.
2. It is a "top-down feedback approach that allows the harmonization of top level grounded symbols and bottom level, sensorimotor interactions with the environment": The feedback $f_{\mathcal{A}}$ from the overlying symbolic algorithm guides the grounding of the symbols explicitly during optimization. Hence, the mapping is performed bottom-up while the feedback for the optimization process is fed top-down and controls the harmonization between both levels.
3. "The availability of some sort of representational capacities in the autonomous agent" holds true as the interface is defined as general mapping. Any mapping with appropriate representational capacities may be chosen. In particular data mining algorithms map parts of $\mathcal{D}$ to $\mathcal{S}$ according to intrinsic structures of the subsymbolic data. Their representational capacities are the reason for their success.

4. "The availability of some sort of categorical/abstracting capacities in the autonomous agent" is an apparent feature when mapping subsymbolic to symbolic data with an interface algorithm: symbols abstract from sensorimotor patterns and form a category (class or cluster). Decontextualization may be implemented on the symbolic level.
5. "The availability of some sort of communication capacities" to "avoid the Wittgensteinian problem of a private language" is also valid from the optimization point of view. First, the development of language is a task that can itself be seen as objective and for that a feedback $f_{\mathcal{A}}$ can be defined. With a 'teacher' the mapping can also be learned in a supervised way, i.e. it can be treated as classification approach. Second, symbols and their meaning may be exchanged exclusively on the symbolic level. Both points of view are consistent with the communication capacity condition.
6. "An evolutionary approach in the development of (1) to (5)." the whole symbol grounding and semantic elaboration process is a process the autonomous agent has to evolve. My formulation of the symbol grounding problem as optimization problem strengthens this assumption. Evolutionary algorithms are an appropriate choice to solve the optimization problem. This issue has already been discussed in Paragraph 3.3.
7. Satisfaction of the zero semantical commitment condition in the development of (1) to (6).
   To my mind the optimization formulation fulfills the first five conditions. Due to its importance, the last property is discussed in the next paragraph.

4.5 Zero Semantical Commitment Condition

It may not be difficult to ground symbols somehow, but to answer the question how an autonomous agent is able to solve this task on his own, to elaborate his own semantics. Genetic preconditions, interaction with the environment and other autonomous agents seem to be the only sources this elaboration is based on in biological systems. The interpretation of symbols must be an intrinsic process to the symbol system itself without extrinsic influence. From this assumption Harnad *et al.* [7, 9] derives three conditions, i.e. 1. no semantic resources are preinstalled in the autonomous agent (no innatism), 2. neither semantic resources are uploaded from outside (no externalism), and 3. the autonomous agents possesses his own means to ground symbols (such as sensors, actuators, computational capacities, syntactical and procedural resources, etc.). Taddeo [19] names this zero semantical commitment condition.

Is the zero semantical commitment condition fulfilled in the interface optimization view? The interface part – as parameterized mapping function – as well as the optimization algorithm are both computational and procedural resources that are allowed in condition 3. Up to here, we can assume that no knowledge about the meaning of symbols is integrated into the autonomous agent. Neither are semantic resources uploaded from outside. The only external knowledge that is used during learning and optimization is a learning signal from the symbolic level that reflects the success. Such a learning signal must exist in every learning scenario. Without any kind of learning signal, learning is not possible at all. If available in form of neurotransmitters in case of nervous systems or in form of survival and reproduction in case of the Darwinian principle of surviving of the fittest, external feedback is the basis of every biological kind of learning. Consequently, to my mind the optimization view does not violate any of the above conditions for symbol grounding with respect to any innatism or externalism, and therefore is a perspective towards the fulfillment of the zero semantical commitment condition.

The formulation of objectives and the appropriate definition of feedback $f_{\mathcal{A}}$ is an open and problem-dependent question. Similarities to reinforcement approaches are obvious. But it has to be considered that the objective function itself does not violate the zero semantical commitment condition as externalism may be introduced, if not the general objective of the autonomous agent is reflected in the feedback, but symbol grounding information in an explicit form. Hence, the only condition for the feedback is to exclusively reflect the fulfillments of the agent's needs and other general objectives.

5 Summary and Conclusion

From the formulation of the interface between subsymbolic and symbolic representations as optimization problem various consequences arise. The optimization process will improve the performance of interfaces and hence the success in solving cognitive tasks. Learning becomes a two-level optimization problem: interface learning and learning on the symbolic level. Many approaches from machine learning for dimensionality reduction, clustering and optimization are adequate methods for the interface problem. The bias of the intrinsic structure in the data that is exploited by data mining algorithms leads to a reasonable reduction of the solution space.

Having this optimization nature in mind, the creator of a cognitive system can invest more time into careful tuning and control of interface properties. Although the flexibility of most current dimensionality reduction and clustering methods is quite high, in the future the creators of artificial intelligent systems may spend more effort in the development of adaptive and evolvable interface algorithms, in particular in online-scenarios as conditions, e.g. the structure of high-dimensional data, may change significantly in time. A solely mathematical and algorithmic formulation only allows a narrow view on the optimization problem. But a fruitful impact for cognitive modeling and the answering to the question how to measure the success of higher cognitive functions from cognitive sciences and psychology becomes important.

Last, the binding of the symbol grounding to the objective of the acting autonomous agent leads to the fulfillment of the zero semantical commitment condition as neither internal nor external knowledge, except the objective of the autonomous agent and its learning algorithms are explicitly integrated into the agent. Interface and optimization algorithm are computational and procedural resources.

References

1. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, August 2006.
2. A. Cangelosi, A. Greco, and S. Harnad. Symbol grounding and the symbolic theft hypothesis. *Simulating the Evolution of Language*, pp. 191–210 Springer, London, 2002.
3. A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing.* Springer, Berlin, 2003.
4. M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise, 1996.
5. L. Floridi. Open problems in the philosophy of information. *Metaphilosophy*, 35:554–582, 2004.
6. S. Harnad. Categorical perception: the groundwork of cognition. pp. 287–300, New York, 1987. Cambridge University Press.
7. S. Harnad. The symbol grounding problem. *Physica D*, pp. 335–346, 1990.
8. S. Harnad. Computation is just interpretable symbol manipulation: Cognition isn't. *Minds and Machines*, 4:379–390, 1994.
9. S. Harnad. Symbol grounding problem. *Scholarpedia*, 2(7):2373, 2007.
10. T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning.* Springer, August 2001.
11. T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
12. M. J. Mayo and A. S. Ai. Symbol grounding and its implications for artificial intelligence. *Twenty-Sixth Australian Computer Science Conference*, pp. 55–60, 2003.
13. R. Pfeifer and F. Iida. Embodied artificial intelligence: Trends and challenges. In *Embodied Artificial Intelligence*, pp. 1–26, 2003.
14. M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964.
15. E. Roth. *Intelligenz – Grundlagen und neuere Forschung.* Kohlhammer, Stuttgart, Berlin, Köln, 1998.
16. J. R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3:417–457, 1980.
17. R. Sun. Symbol grounding: A new look at an old idea. *Philosophical Psychology*, 13:149–172, 1999.
18. R. Sun and T. Peterson. Some experiments with a hybrid model for learning sequential decision making. *Information Sciences*, 111:83–107, 1998.
19. M. Taddeo and L. Floridi. Solving the symbol grounding problem: a critical review of fifteen years of research. *J. Exp. Theor. Artif. Intell.*, 17(4):419–445, 2005.
20. T. Ziemke. Embodied ai as science: Models of embodied cognition, embodied models of cognition, or both? In *Embodied Artificial Intelligence*, pp. 27–36, 2003.

Sensorimotor Self-Motivated Cognition

Martin V. Butz

University of Würzburg, Department of Psychology III
Röntgenring 11, 97070 Würzburg, Germany
{butz}@psychologie.uni-wuerzburg.de
http://www.coboslab.psychologie.uni-wuerzburg.de/

Abstract. Over the last years, sensorimotor codes, which are often termed schemata in their symbolic form, are becoming increasingly accepted as a fundamental structural code for biological cognitive systems. In their most embodied form, sensorimotor structures encode the correlations of behavioral activities with their immediate perceptual consequences. Behavior can thereby be a particular muscular contraction, more elaborate motor synergistic muscular activity, or even complex (trained) dynamic movements. Perceptual consequences may be very immediate proprioceptions, more elaborate sensory changes, or even complex perceptual dynamics. Besides their immediate behavioral relevance, sensorimotor codes form the glue that links sensory and behavioral codes together forming cognitive maps that enable the execution of complex goal-directed behavior. Together, sensorimotor bodyspaces are formed in which distances in space are not sensory but they reflect the motor effort necessary to bridge a particular distance. To create intrinsically motivated behavioral systems, motivations can be added to these modular bodyspace representations. We show that different types of motivations need to be distinguished for an effective design of interactively multi-motivated systems. Moreover, we show that such designs can be easily integrated into the sensorimotor bodyspace representations. In conclusion, we propose that motivations may not only be necessary to induce goal-directed behavior, but they may also be a highly important component for shaping higher level cognitive modules.

Key words: Sensorimotor Codes, Cognitive Maps, Bodyspaces, Motivations, Homeostasis, Self-Motivated Reinforcement Learning, levels of cognition

1 Sensorimotor Bodyspaces

From a high-level conscious view-point, we often perceive ourselves in the surrounding space from a somewhat allocentric, abstract perspective. During communication, we might localize ourselves in the environment as currently being, for example, in a certain room, a city, or country. If relevant for the conversation, however, we turn to more ego-centric information such as, for example, attending a certain event, facing a certain object or person, listening to a particular musical piece, or watching a certain program on TV. Thus, our self-perception can have many perspectives and is integrated into various more-or-less egocentric points of view.

What constitutes these perspectives? That is, in which forms of representations are such perspectives embedded? Various research directions suggest that the basis for our spatial perceptions are sensorimotor codes, which are necessarily purely egocentric. Wolff reviewed eye movement studies and concluded that a spatial representation has do have a dominant, sensorimotor component [1]. Studies in reaching movements led to the suggestion that multiple coupled forward-inverse models exist for motor control [2]. Experimental evidence is available that shows that dynamic sensorimotor encodings can transfer bidirectionally between different tasks and different sensory modalities. For example, rotations can transfer bidirectionally from eye tracking to pointing [3]—which led the authors to the conclusion that the investigated adaptation mechanism lies in a common dynamic code that can transfer between categories. Similarly, dynamic hand movements can transfer between hands [4]. Thus, dynamic movements are partially encoded by a common code, as, for example, proposed in the theory of event coding (TEC) [5] and anticipatory, sensorimotor structures, or schemata, constitute the basis of this coding scheme [6, 7].

Interestingly, from a much more computationally oriented perspective, it was shown that correlations between sensory and motor codes may reveal the dimensionality in which interactions take place [8]. The authors show that the number of components represented in a correlation mapping allows the deduction of the dimensions of physical space. An integrated perspective of the body integrated into multimodal, highly modular sensorimotor bodyspaces can be found elsewhere [9].

As a consequence, sensorimotor bodyspace representations do not encode the space purely sensory, but space is represented with various sensorimotor codes, which implies that distances in space are represented motor-dependently. And in fact, the conscious representation of spatial distances depends on the motor effort necessary to bridge the questioned distance [10]. It was even shown that tool-use can alter the distance perception as well—especially when tool-use is intended and an object is in reach with the tool but not without it [11].

In neuroscience studies with monkeys, it was even shown that single premotor cortical neurons with mirror-neuron properties distinguish between reachable and non-reachable locations in space [12]. Moreover, the structure of the monkey cortex was shown to be partitioned into various ethologically-relevant functions, besides limb topology and simple movement typological distinctions [13]. The different aspects are encoded with somewhat distinct regional, neural population codes. A single neuron in such a code may control a different facet of the encoded overall behavior and can be context-dependently modulated. Similar population codes are also found in the parietal cortex, where peripersonal body spaces are encoded [14], which surround the body and the reachable space with distributed neural codes. Thus, population codes are a fundamental encoding scheme of the brain. Each neuron in a population code represents a particular aspect of the code, such as a particular arm constellation in motor cortex or a particular hand-relative object location in parietal cortex.

Although most of the studies above investigate body-part-relative movements and representations (especially hand and arm but also eyes) rather than whole body movements, there are several indications that also spatial representations that have whole body movements as their motor-code use similar sensorimotor encoding strategies. In the hippocampus, information converges for the formation of episodic memory. During movements through space, place cells and head-direction cells were localized (among others) in the rat's hippocampus and were mimicked by associative population codes [15]. Various indications now suggest that the hippocampus is not only involved in the integration of allocentric and egocentric representations but also play an important role for the goal-oriented execution of behavior in space [16, 17].

In sum, sensorimotor highly interactive and dynamic bodyspace representations are omnipresent in the brain. They range from simple reaching movements to elaborate body movements and categorical movements, which are each encoded with modular sub-populations of neurons. Each sub-population covers a particular behavioral task or aspect of interaction with the environment, including manipulating behaviors as well as navigating behaviors. These representations however do not only serve as spatial representations and immediate behavioral control components, but also appear to constitute the basis for even higher forms of cognitive representations, leading eventually to complex social interaction, language, and abstract thought capabilities [18, 19].

2 Sensorimotor Models

Over the recent years, our research group has developed several sensorimotor models that are self-organized and developed for goal-directed behavioral control. Two models will be shortly reviewed here.

The sensorimotor redundancy resolving architecture SURE_REACH [20–22] is a psychological plausible model of arm reaching behavior. It consists of two population codes that interactively represent and control the movement to hand locations or arm postures in the reachable space. An associative, inverse kinematics mapping correlates hand locations with redundant arm postures (one-to-many mapping) and a sensorimotor model self-associates arm postures motor-dependently. The latter essentially forms sensorimotor connections between behaviorally close postures, where each connection stores the motor vector that is necessary to reach the one posture from the other.

It was shown that this representation may be regarded as a neural implementation of the posture-based motion planning theory [23] with the additional capabilities of anticipated subsequent end-state considerations while reaching for a current target and the resulting closed-loop control of the arm. Most recently, the architecture was also applied to the control of a dynamic arm system in a realistic, physical 3D simulation environment [24].

While the inverse kinematics and sensorimotor mapping is learned in the original SURE_REACH implementation [20], the population codes were uniformly distributed in hand-location and arm-posture space. To overcome this shortcoming, we enhanced self-organizing neural network techniques to be able to connect perceptual spaces motor-dependently. The time-growing neural gas network (TGNG) grows a population code that covers a particular space, while the neural connectivity is motor-dependent [25]. It was shown that the resulting representation implicitly encodes motor-dependent distances in the explored space. So far, TGNG was applied to realize goal-directed robot movements in various maze-like environments. However, in principle the TGNG approach could also grow the population codes utilized in SURE_REACH.

Goal-directed behavior is realized in both systems by model-based reinforcement learning mechanisms, which is essentially discrete dynamic programming realized within the population encodings [26, 20, 25]. Given a particular external goal activation a_i^e of neuron i in the population code, the activity is propagated by

$$a_i \leftarrow \max\left\{a_i^e; \max_j\left[\gamma a_j\right]\right\}, \tag{1}$$

where a_i denotes the current activity of neuron i and index j iterates over all neurons j that are connected to neuron i via sensorimotor connections. Given the system state (such as the posture or location) is currently represented by a neuron i (usually a set of neurons represents the state of the system), then the behavior is determined by the motor activity that is stored in the sensorimotor connection that connects to the most activated neuron j, that is, arg $\max_j a_j$.

In sum, two sensorimotor population-encoded models exist and can be applied for the flexible, goal-directed control of arm-movements and full body movements. In the following, we show that these encodings are highly suitable to add motivational constraints and activated self-motivated behavioral goals.

3 Self-Motivated Behavior

Until recently, the utility of the introduced population encodings was shown due to their (1) psychological and neuroscientific validity and (2) their capability to plan and control flexible, goal-directed behavior. However, for the design of an autonomous, cognitive system, goals and constraints need to be self-activated. Thus, we now give an overview of how such self-motivated activities may be included in these systems. We essentially propose that the system should strive for inner homeostatic states, which may be encoded in a reservoir framework. These states may represent the internal needs of the particular system, such as hunger or thirst, as well as even more abstract homeostatic needs, such as the urge for safety, which can lead to a scared system, or knowledge discovery, which can lead to a curious system. As proposed elsewhere [27], we propose to distinguish between these two types by terming the former consummatory motivations and the latter property-based motivations.

3.1 Homeostatic Reservoirs

In a similar vein to other recent publications [28, 29], we proposed to use reservoir encodings to reflect the actual needs of a system [27]: Each reservoir x can be represented by an internal state value $\sigma_x(t) \in [0, 1]$, which encodes how full the reservoir is at time t. Moreover, each reservoir may have an associated update function $\phi_x : [0, 1] \to [0, 1]$ and a weighting function $\omega_x : [0, 1] \to [0, 1]$. The update function specifies the change in the reservoir level over time given current interactions with the environment. The weighting function further controls the impact of the current reservoir state on behavior. Intuitively, this function encodes the importance of re-filling reservoir x given its current state. In addition, a constant priority weighting p_x for each drive describes the importance of this drive compared to the others. Thus, the overall importance can be computed as:

$$\iota_x(t) = \omega(\sigma_x(t))p_x. \tag{2}$$

The equation essentially reflects the importance of drive x and thus can be used to motivate current behavior. Given all current importance values, behavior can be controlled to still the currently most-important need satisfying other needs on the way given an appropriate opportunity. Thus, self-motivated behavior can be realized.

A fundamental distinction, however, can be drawn between motivations that can be stilled by a typical consummatory behavior, such as eating or drinking, or by obeying particular constraints, such as not entering certain regions. While the proposition of this general distinction was made elsewhere [27], terming it location-based and property-based motivations, here we further detail this distinction and embed it into a wider context. We consequently also refer to the location-based motivations more generally as *consummatory* motivations.

3.2 Consummatory Motivations

The impact of each motivations depends on its update function ϕ_x. Generally and without other context information, ϕ_x may be considered as continuously decreasing reflecting the continuous bodily consumption of energy. However, in consummatory motivations an increase in the reservoir state occurs only upon a (successful) consummatory behavior while an increase in property-based motivations occurs while the encoded property is increasingly satisfied. Also blends between the two types are certainly possible.

With respect to sensorimotor population codes, consummatory motivations are satisfied upon the successful execution of a particular behavior in a particular context, such as eating. Thus, consummatory motivations can be associated with particular behavior patterns that are executed in a particular behaviorally-relevant context. These behavior patterns may be represented by particular neurons in a population code. These neurons may be activated when the importance $\iota_x(t)$ grows to a certain level compared with the other importance values $\iota_y(t)$ of other motivations y. The neural activity can then serve as a particular goal representation, which can lead to goal-directed behavioral patterns, regardless if using the SURE_REACH encoding for arm control or the TGNG encoding for navigation control.

Interestingly, Graziano has shown that ethologically relevant behaviors are encoded in partially regionally distinct population codes. Thus, property-based motivations may be associated with particular ethologically relevant behavioral patterns and may even lead to the formation of such patterns in the first place.

3.3 Property-Based Motivations

In difference to consummatory motivations, property-based motivations rather strive for the maintenance or avoidance of particular situations. For example, we do not like to keep our arm in an uncomfortable position (such as an extreme twist) for an extended amount of time. This was shown in tasks in which a particular task had to be executed leading to a particular end-state. Results show that we chose, for example, our initial grip of a stick based on the anticipated final state, which is optimized for end-state posture comfort [30].

In the navigation domain, scared animals may avoid open areas striving for protection and thus preferring to move along walls or through tunnels. In this case a safety motivation may exist that leads to the modification of path planning given the current drive for safety. In population codes, both types may be encoded by a preference bias that is spread over the full population code. That is, while a consummatory motivation may activate a certain sub-population or even single neurons in a population code, property-based motivations pre-activate or pre-inhibit full population codes property-dependent.

Thus, property-based motivations will have a different effect on behavior than consummatory motivations.

3.4 Both Types Combined

When combining both types of motivations, it does not come as a surprise that they need to be handled differently. In fact, it was shown that property-based motivations need to modify the activity propagation of consummatory motivations through a population code in order to realize goal-directed behavior while satisfying property-based motivations.

As an example, it was shown that a simulated, "scared" robot may walk along walls in order to reach food locations by modifying the food-originating activity propagation by a wall closeness preference:

$$h_i \leftarrow \max\left\{h_i^e; \max_j\left[\gamma(h_j + (s_j - 1)\iota_f)\right]\right\}, \tag{3}$$

where h_i^e denotes the consummatory motivational activity in neuron i, h_i the propagated consummatory activity, s_i the property-based motivational activity, ι_f the current importance of the property-based motivation, and where j iterates over all neighboring neurons of neuron i (cf. [27]). Note that $(s_j - 1)\iota_f$ is always a negative value, which essentially denotes the cost of moving towards node j.

In general, constraint goal-pursuance can be realized by constraining consummatory activity propagation by the current property-based motivational activities. The result is a system that acts goal-directedly striving for consummatory behaviors while obeying other property-based motivations concurrently. For TGNG, it was shown that the combination results in a system that exhibits latent learning capabilities, exploits behavioral opportunities, and yields emergent behavioral patterns due to the concurrent combination of different motivations respecting their current priorities [27].

Since the SURE_REACH architecture essentially encodes sensorimotor spaces similar to the TGNG approach for navigational spaces, similar motivational combinations can be used also for SURE_REACH. So far, only a simple priority-based drive has been included in SURE_REACH in order to avoid extreme arm postures [31], but more elaborate combinations are in preparation.

4 Summary and Conclusions

Summing up, we proposed that population codes can be very suitably combined with motivational drives in order to realize systems that exhibit self-motivated, goal-oriented behavior. Such systems may thus not need to be explicitly programmed to execute particular behaviors or reach particular goals, but rather only need to be informed about which internal variables need to be kept in sufficient homeostasis. Learning and adaptation of the developed population code and their association with the particular motivations then leads to the pursuance of goal-directed behavior.

The changes in the internal variables can additionally distinguish between consummatory motivations and property-based motivations. Thus, the distinct propagation of both types of motivation and the influence of property-based motivations on the consummatory activity propagations can be realized without the need for pre-programming. This motivation concept can be added to both, the SURE_REACH system for flexible, end-posture oriented arm control [20] as well as the TGNG model in its navigation tasks [25].

In general, any system that utilizes sensorimotor codes for effective behavior control may be combined with the homeostatic motivation concept. For example, a sensorimotor representation was recently used to optimize self-localization based on the principle of information gain [32]. The utilized information gain principle may be coupled with the proposed curiosity-based motivation. In this way, the system my become even more knowledge-gain directed, as long as the curiosity drive is stronger than the current consummatory motivations. Similarly, more schema-based approaches such as anticipatory learning classifier system architectures [33, 34] may be combined with such reservoir-based motivational systems, as originally already envisioned by John H. Holland in his first learning classifier system implementation—the "cognitive system" CS1 [35].

While the thoughts pursued herein were so-far only tested on rather static, location-based sensorimotor population codes, it becomes apparent that many behaviors are not simple transitions in space or posture, but unfold in an extended fashion over time. As mentioned above, a recent neuroscience review of Graziano [13] suggests that the motor cortex concurrently encodes ethologically-relevant behaviors besides posture-based and directional codes of particular limbs or motor synergistic combinations of limbs.

Particularly the formation of ethologically relevant behavioral codes may be controlled not directly by genetic encodings, but rather by motivational encodings coupled with an appropriate bodily morphology and basic reflexes. Thus, motivations may serve as a fundamental brain structuring principle that may lead to the formation of those increasingly abstract sensorimotor representations that are relevant for the organism. By the encoding of relevancy (via motivations) rather than the full behavior, much more flexibility may be maintained in the evolution of the respective genetic encoding and in learning to satisfy the respective motivational drives during development. Future research will show to what extend this proposition holds for highly complex, adaptive, social, cognitive systems, such as us humans.

References

1. Wolff, P.: Wahrnehmungslernen durch Blickbewegungen. In Neumann, O., ed.: Perspektiven der Kognitionspsychologie. Springer-Verlag (1985) 63–111
2. Wolpert, D.M., Goodbody, S.J., Husain, M.: Maintaining internal representations: the role of the human superior parietal lobe. Nature Neuroscience **1** (1998) 529 – 533
3. Bock, S.A.O.: Transfer of sensorimotor adaptation between different movement categories. Experimental Brain Research **148**(1) (2003) 128–132
4. Thut, G., Cook, N.D., Regard, M., Leenders, K.L., Halsband, U., Landis, T.: Intermanual transfer of proximal and distal motor engrams in humans. Experimental Brain Research **108** (1996) 321–327
5. Hommel, B., Müsseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): A framework for perception and action planning. Behavioral and Brain Sciences **24** (2001) 849–878
6. Hoffmann, J.: Vorhersage und Erkenntnis: Die Funktion von Antizipationen in der menschlichen Verhaltenssteuerung und Wahrnehmung. [Anticipation and cognition: The function of anticipations in human behavioral control and perception.]. Hogrefe, Göttingen, Germany (1993)
7. Hoffmann, J.: Anticipatory behavioral control. In Butz, M.V., Sigaud, O., Gérard, P., eds.: Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems. Springer-Verlag, Berlin Heidelberg (2003) 44–65
8. Philipona, D., O'Regan, J.K., Nadal, J.P.: Is there something out there? Inferring space from sensorimotor dependencies. Neural Computation **15** (2003) 2029–2049
9. Butz, M.V.: Sensomotorische Raumreprsentationen. Informatik-Spektrum **31** (2008) 237–240
10. Witt, J.K., Proffitt, D.R.: Action-specific influences on distance perception: a role for motor simulation. J Exp Psychol Hum Percept Perform **34**(6) (2008) 1479–1492

11. Witt, J.K., Proffitt, D.R., Epstein, W.: Tool use affects perceived distance, but only when you intend to use it. Journal of Experimental Psychology: Human Perception and Performance **31** (2005) 880888
12. Caggiano, V., Fogassi, L., Rizzolatti, G., Thier, P., Casile, A.: Mirror neurons differentially encode the peripersonal and extrapersonal space of monkeys. Science **324** (2009) 403–406
13. Graziano, M.S.A.: The organization of behavioral repertoire in motor cortex. Annual Review of Neuroscience **29** (2006) 105–134
14. Holmes, N.P., Spence, C.: The body schema and multisensory representation(s) of peripersonal space. Cognitive Processing **5** (2004) 94–105
15. Arleo, A., Gerstner, W.: Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity. Biological Cybernetics **83** (2000) 287–299
16. Fleischer, J.G.: Neural correlates of anticipation in cerebellum, basal ganglia, and hippocampus. In Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G., eds.: Anticipatory Behavior in Adaptive Learning Systems: From Brains to Individual and Social Behavior. Springer-Verlag (2007) 19–34
17. Hok, V., Lenck-Santini, P., Roux, S., Save, E., Muller, R.U., Poucet, B.: Goal-related activity in hippocampal place cells. The Journal of Neuroscience **27** (2007) 472 482
18. Butz, M.V.: How and why the brain lays the foundations for a conscious self. Constructivist Foundations **4**(1) (2008) 1–42
19. Gallese, V.: Motor abstraction: a neuroscientific account of how action goals and intentions are mapped and understood. Psychological Research **73**(4) (2009) 486 – 498
20. Butz, M.V., Herbort, O., Hoffmann, J.: Exploiting redundancy for flexible behavior: Unsupervised learning in a modular sensorimotor control architecture. Psychological Review **114** (2007) 1015–1046
21. Herbort, O., Butz, M.V.: Encoding complete body models enables task dependent optimal behavior. Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007 (2007) 1424–1429
22. Herbort, O., Ognibene, D., Butz, M.V., Baldassarre, G.: Learning to select targets within targets in reaching tasks. 6th IEEE International Conference on Development and Learning **ICDL 2007** (2007) 7 – 12
23. Rosenbaum, D.A., Meulenbroek, R.G.J., Vaughan, J., Jansen, C.: Posture-based motion planning: Applications to grasping. Psychological Review **108** (2001) 709–734
24. Pedersen, G., Butz, M.V., Herbort, O.: Integrating dynamics into a human behavior model for highly flexible autonomous manipulator control. IEEE Robotics and Automation Magazine (submitted)
25. Butz, M.V., Reif, K., Herbort, O.: Bridging the gap: Learning sensorimotor-linked population codes for planning and motor control. International Conference on Cognitive Systems **CogSys 2008** (2008)
26. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT Press, Cambridge, MA (1998)
27. Shirinov, E., Butz, M.V.: Distinction between types of motivations: Emergent behavior with a neural, model-based reinforcement learning system. 2009 IEEE Symposium Series on Artificial Life (ALIFE 2009) Proceedings (2009) 69–76
28. Hsiao, K.Y., Roy, D.: A habit system for an interactive robot. In Castelfranchi, C., Balkenius, C., Butz, M.V., Ortony, A., eds.: From Reactive to Anticipatory Cognitive Embodied Systems: Papers from the AAAI Fall Symposium, Menlo Park, CA, AAAI Press (2005) 83–90

29. Konidaris, G., Barto, A.: An adaptive robot motivational system. From Animals to Animats **9** (2006) 346–356
30. Weigelt, M., Kunde, W., Prinz, W.: End-state comfort in bimanual object manipulation. Experimental Psychology **53** (2006) 143–148
31. Butz, M.V., Pedersen, G.K.M.: The scared robot: Motivations in a simulated robot arm. 32nd Annual Conference on Artificial Intelligence **KI 2009** (in press)
32. Zetzsche, C., Wolter, J., Schill, K.: Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. Cognitive Processing **9** (2008) 283–297
33. Butz, M.V., Goldberg, D.E.: Generalized state values in an anticipatory learning classifier system. In Butz, M.V., Sigaud, O., Gérard, P., eds.: Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems. Springer-Verlag, Berlin Heidelberg (2003) 282–301
34. Gérard, P., Sigaud, O.: YACS: Combining dynamic programming with generalization in classifier systems. In Lanzi, P.L., Stolzmann, W., Wilson, S.W., eds.: Advances in learning classifier systems: Third international workshop, IWLCS 2000 (LNAI 1996). Springer-Verlag, Berlin Heidelberg (2001) 52–69
35. Holland, J.H., Reitman, J.S.: Cognitive systems based on adaptive algorithms. In Waterman, D.A., Hayes-Roth, F., eds.: Pattern directed inference systems. Academic Press, New York (1978) 313–329

Conflict Resolution while Coping with Critical Situations

Régis Newo[1], Klaus-Dieter Althoff[1], and Werner Greve[2]

[1] University of Hildesheim, Institute of Computer Sciences,
Laboratory of Intelligent Information Systems
Email: newo|althoff@iis.uni-hildesheim.de
[2] University of Hildesheim, Institute of Psychology
Email: wgreve@uni-hildesheim.de

Abstract. When facing critical situations, for example the loss of a job or the breakup of a partner, humans reactions are not predictable. In this paper we present apart of our conceptual design of a system, which simulates human's behavior in critical situations. We concentrate in this paper on conflict management strategies for our simulation Our system is based on the multi-agent systems technology and use planing algorithms. We show how we intend to resolve conflicts while coping with critical situations and present the current state of our work.

1 Introduction

How does someone react when he faces a critical situation in his life? In our everyday life, we consistently face situations which pose more or less immense challenges. Examples can be the breakup with a partner, the loss of a job, an illness or even the death of a relative. As different as those challenges can be, the reactions of the persons who are facing the same kind of challenges can be very different as well. The problem consists in finding out how someone reacts when he/she faces up a given challenge. The problem being a psychological one, there have been many research groups in psychology working in that direction, beginning in the early 1980s. They developed psychological models and paradigms in order to represent and analyze people's behaviors.

In this paper, we present our approach for the simulation of human's behavior in critical situations. From a psychological point of view, our approach is based on the theory on coping strategies developed by Brandtstädter and Greve [1]. On the other hand, from a computer sciences point of view, our approach relies on the use of multi-agent systems and Case-Based Reasoning (CBR) [2] as the main knowledge representation inference. CBR is based uses past experiences to solve new problems. We use it because human's way to act is mostly based on past experiences (first or second hand).

Past approaches of modeling psychic processes have remained on a macroscopical level, as it were, simply connecting functional devices such as "central executive", or "motivational center", etc. Switching to an agent-based approach, it will become possible to include dynamical interactions within such functional clusters (e.g., interactions between various goals a person holds or between heterogeneous emotional states - "mixed emotions" - within a person). This will certainly come much closer to what we actually are than roughly computer-metaphor inspired simulations. On the other hand, agent-based simulational models within the psychological realm have exclusively focused on interactions between persons (e.g., attitude change; [3]). Hence, the combination of (a) simulational approaches of intrapsychic processes (such as coping responses) and (b) agent-based technologies seems to be a highly promising constellation to further advance both the applicational options of simuational models and the theoretical integrity and clarity of psychological theories in the coping realm.

For this purpose we developed the SIMOCOSTS (SImulation MOdel for COping STrategy Selection) model. In the SIMOCOSTS project we are actually aiming at a threefold goal, namely (1) developing a research software tool for supporting psychologists, who are working on cognitive modeling and learning as roughly described above, in their research work, (2) realizing what we call "collaborative multi-expert-systems" (CoMES; [4]), and (3) instantiating the SEASALT software architecture [5] we developed in our research lab as a first step towards realizing CoMES.

Our approach for the simulation in this paper is based on the fact that each person is goal-driven. That means that the actions made by the person are intended to be a part of the achievement of a certain goal. The simulated person has many goals and each goal wants to be fulfilled independently of each other and plans have to be computed (for each goal) in order achieve the goals. A critical situation is thus a situation in which a goal can not be fulfilled. These goals all interact in a sort of market place. We use the terminology "market place" because we want to accentuate the competitiveness of the goals. This might lead to conflicts. Conflicts occur when different plans (which were computed for different goals) contain actions which are contradictory. We thus also have elaborate a conflict management methodology for our approach. This is the main focus of this paper.

We will first present in the next section some psychological background. After presenting some related work, we will explain how we are actually implementing our system by elaborating on the underlying concepts and the used algorithm for conflict resolution. We will present the current state and an outlook on the implementation of our system.

2 Psychological Background

Psychological coping research, during the past three decades, has largely rested on correlational questionnaire studies. Unfortunately, the causal connections between the various factors included in the available theoretical models can hardly be tested with these data. On the other hand, valid experimental studies can hardly be done in this highly sensible area, both for ethical and practical reasons. As a consequence, theoretical models have remain underdeveloped and seldom directly tested. Notwithstanding a bulk of empirical studies in this field, we still do not know the interplay of different facets and layers of the "psychic system" in its response to a threatening or burdensome experience or constellation. From a theoretical point of view, however, this interplay of intrapsychic factors is crucial for our understand of coping processes and, thus, for successful intervention. Moreover, the possible intersections to developmental theories (i.e., processing developmental challenges and tasks) is another underinvestigated issue.

At this juncture, simulational methods offer a highly useful way to sharpen theoretical assumptions (claims) and to test theoretical hypotheses on possible interactional processes of several psychic subsystems. In order to create a formalized model, an empirically corroborated theory is needed in the first place. In our work, we start from the two-process model of developmental regulation [6–8]. The starting point here is the consideration that stressful events, threats to identity, and developmental losses can be understood as problem situations with an underlying discrepancy between an is and a should be perspective of personal development, that is a regulatory deficit. However, in this approach, the differentiation between fundamental reaction modes is drawn along the boundaries of personal (i.e., behavior which is consciously and intentionally planned and governed by the person as the acting unit) and subpersonal (i.e., intraindividual processes such as information processing or emotional regulation which cannot be controlled or even initiated, often not even consciously be recognized by the person) perspectives against the background of an action-theoretical perspective of human development [9]. The model basically differentiates between two modes of coping with problems, designated as *assimilative* and *accommodative* processes [7]; these can be supported by a third mode of dealing with threats: *defensive* processes [1].

Assimilative Strategies: Intentional Self-Development. In the assimilative reaction mode, individuals try to change their life situation or their own behavior in the sense of a better alignment between their normative expectations and goals in relation to themselves [6]. For example, we can do sports to improve diminishing physical condition, or change our eating behavior to make our figure closer to our ideal in this respect. Characteristic for this mode is that personal standards and goals underlying the situational or developmental appraisal are maintained. Coping attempts in this mode are usually carried out intentionally, consciously, and controlled, and can thus appropriately be called coping strategies.

Accommodative Processes: Development as Adaptation. The attempt to remove or prevent developmental losses by means of active problem-solving can fail or be bound to difficulties and costs that are too high. Often in life fundamental revisions in life- and developmental blueprints become necessary beyond simply compensatory measures. Serious threats occur that cannot be actively removed and need to be resolved through reactive preference readjustments. In response to these burdens, the alternative option consists of revising standards and goals to the given action possibilities: This is the accommodative mode. Typical examples of accommodative reactions are the relinquishing and devaluation of blocked goals, processes of regulating standards, but also processes that lead to a more readily acceptable reinterpretation of the given situation. According to Brandtstädter's view, neither of the modes has primacy. For a given situation it is not only open which of the modes is "appropriate" or even "successful"; and it is also an empirical question with which modus the person will initially react in a stress situation; from a dynamic perspective it might often even be that just the combination of both forms is effective.

Defensively Dealing with Problems: Escape or Detour? From a coping point of view, however, it seems to make sense to add a third reaction mode to the developmental model that several of the above-mentioned models included: Individuals can apparently also completely ignore a problem, denying its meaning or even its existence. In this case they change neither the problem nor themselves: Neither personal goals, preferences, standards, nor aspects of the self-image get adjusted nor does the problem get solved actively. This defensive mode operates, as it were, entirely behind the back, as the mechanisms as well as the effects of these processes principally remain hidden from the individual.

3 Related Work

As already discussed in [10] and [11], there already exists many agent-based simulation approaches (like EOS [12] and Sugarscape [13]) that deal with human behavior. However we can not use them for our simulation because they do not deal with coping. Furthermore, cognitive architecture have also been developed (like ACT-R [14] and EPIC [15]). Yet they do not deal with critical situations. In [10] and [11], we showed a process based architecture for SIMOCOSTS. Our work is based on it. In this section we will introduce some techniques used in our simulation.

As said earlier, the simulated person has several goals and each of them needs to compute a plan on how they will be achieved. The type of planning we will use for our simulation is derived from logic-based planning (see [16]). The reason is that the simulated person is mainly represented with predicates, which give us the current state (physical and psychological) of the person. In the area of (agent) planning, some work has also already be done. We can for example see in [17] that there exists several kinds of planning algorithms. The first type of planning algorithms are the so-called linear algorithms. The particularity of linear algorithm is that the generated plan consists of actions which are chosen by only considering the preconditions and postconditions of the actions. Essentially 2 types of linear planning algorithms exists

- Progression (also called set based planning): the actions are chosen while transforming the initial state into the goal state. STRIPS [18] is a prominent example.
- Regression: the actions are chosen while transforming the goal state into the initial state.

Other methods include plan based planning and graph based planning (see [17]). The main difference between linear algorithms and the other planning methods is that dependencies between the actions are not considered in linear algorithms.

As for agent planning while dealing with conflicts, some work was already done the area also. One good example is done by Timm in [19]. Timm actually differentiate between 2 types of conflicts management methods (i.e. internal and external). The internal one considers that an agent might have several goals and trying to accomplish them might create conflicts. He developed a method (called cobac) in order to solve those conflicts. Yet we can not use it, because the agents in our simulation just have one goal they want achieve. Each simulated person has many goals and each goal is implemented as an agent. We thus do not have internal conflicts in our simulation.
His external method deals with the conflicts that might occur between several agents. His proposed algorithm (called oac) mainly uses communication between the agents in order to resolve the conflicts. That means, the agents automatically try to solve the conflicts by communicating with each other and proposing solutions. Because of the fact that we consider that the simulated person acts in a critical situation, it is not realistic to suppose that the person would take his time and elaborate a perfect solution to his problems or conflicts. The person would rather try to find quick solutions for his most important goals. The second method of Timm is thus not appropriate for our simulation either. We will explain in the next how we intend to deal with conflicts in our simulation.

4 Simulation

In this section we will explain how we intend to implement the simulation with a focus on conflict management.

In Figure 1, we can see the course of the simulation. At the beginning we have an initial (generated) situation. When facing a new situation, each agent tries to find out if the situation is critical for him. This will be done by comparing the goal state of the agent with the current state (inclusive the new situation) of the person.

Let us take a student, called Mr. X, as our example. Mr. X has three goals

1. have a stable financial situation

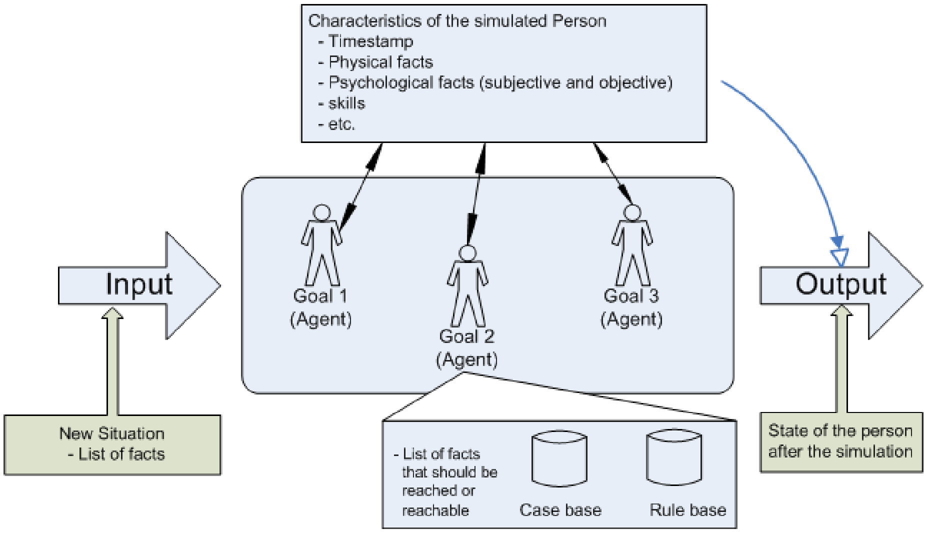

Characteristics of the simulated Person
- Timestamp
- Physical facts
- Psychological facts (subjective and objective)
- skills
- etc.
Input
Goal 1 (Agent)
Goal 2 (Agent)
Goal 3 (Agent)
Output
New Situation
- List of facts
- List of facts that should be reached or reachable
Case base
Rule base
State of the person after the simulation

Fig. 1. Illustration of the simulation

2. finish his master's degree in Business Informatics
3. create a family

We can further suppose that none of the listed goals has already been reached. Nevertheless, plans have have already been computed in order to achieve these goals. That means that the goals are reachable and the actual situation is therefore not critical. Knowing for example that three out of four basic courses in computer sciences (let say CS1 till CS4) are needed (among other courses) in order to finish his master's degree, a valid plan for that goal might contain the actions which prompt Mr.X to take the courses CS1, CS2 and CS3.
A new situation with the information "CS3 failed" would be a critical situation for Mr. X because the second goal is not reachable anymore (with the actual plan).

We saw in earlier that the market place, in which the goals interact, is the most important part of the simulated person (and thus of the simulation). We will now explain how the intend to implement it.
The market place contains many competitive goals. As it is the case for humans, the goals are prioritized. The market place is implemented as a multiagent system with each goal being implemented by an agent. In fact these agents will be implemented following the Belief-Desire-Intention (BDI) principle (see [20]). The different properties of the person (which are the same for each agent) represent the situation. The simulated person also have many actions (e.g. take the course CS3), which can be used by any agent. In each situation, each agent tries to find out if it is critical for him. If it is the case, the agent tries to find a solution by computing a (linear) plan. The computed plan consists of several actions which

might or should affect the situation. The situation is thus updated after each action. The actions contain all facts, which will be modified in the situation after it has been applied. They also contain a time stamp which indicates the time needed before an action is completed.

A conflict occurs when the plan computed by one agent affects another agent. That is, when applying an action, the situation might change so that it becomes critical for another agent. In our example, a new plan for the second goal might include taking the course CS4 in the next term (which will be used instead of CS3). Yet if we suppose that the plan of the first goal includes having a job (in the next term) at which the person has to be at the same time on which CS4 takes place, we would have a conflict.

We developed a methodology to resolve conflicts in our simulation which is based on the prioritization of the goals. Our methodology is based on the fact that a human will first try to achieve his most important goals before achieving the others. The algorithm can be seen below.

```
if any conflict exists then
  $ConflictedAgents \leftarrow \{A_1, \dots, A_n\}$
  while $ConflictedAgents \neq \emptyset$ do
    Recompute the plan for the agent $A_i$ with the highest priority while
    considering the situations generated by previous plans.
    Save the all different situations from the beginning of the plan to the end.
    remove $A_i$ from $ConflictedAgents$
  end while
end if
```

When applying it to the example, this simply means that we would solve the conflict by first trying to recompute the plan of the most important goal, which states that the person wants to have a stable financial situation. We will suppose that the initial plan for this goal do not change, which means that the person will keep his job. Then a new plan for the next goal in the set of conflicted agents (master's degree in business Informatics) should be computed. The agent might know from its knowledge base that the course CS3 takes place each year. That would lead to a plan stating that Mr. X should take that course again in the following year. In this case the conflict would be solved.

There are a few thing that should be noticed for our conflict management methodology. First, recomputing plans does not always lead to valid one. In this case, we will have to reconsider the intentions (i.e. the goals). That means, either the goal itself or its priority will be changed (leading to an accommodative process).

Second, for a better conflict management, the new plans of each agent should be computed while taking the modifications of the situation by higher prioritized agents into account.

The output of our simulation is the state of person (i.e. the situation) after the execution of all plans and also the plans as justifications. An important point to our simulation is that all computed plans are not executed by the person. They rather represent what the person "thinks", he should do. The person would in fact just execute the final solution (i.e. the output of the simulation)

5 Conclusion and Outlook

We presented in this paper how we intend to implement our approach SIMOCOSTS for the simulation human's behavior in critical situations. Our approach is based on the theory on coping strategies developed by Brandtstädter and Greve. Our simulation is based on the multi-agent system technology. We explained the underlying algorithm and showed by means of an example how it works. In this paper we focused on the conflict management of our agent-based system. The methodology used for this purpose derivate from human's behavior in critical situations.

Currently, we are still implementing the system. We can already represent the psychological and physical state of a person, which is an important step for the representation of the situations. We are now implementing the course of of the presented algorithm. Nevertheless a first prototype will be available in a few months. The system, once implemented, will be handed to psychologists who will conduct experiments with it and will also fill it with the required knowledge for the experiments. We will just have a few examples at our disposal which will be used for testing while implementing the systems.
We also, with this system, fulfill our aim of extending our CoMES [4] environment with another distributed knowledge-based system. Another goal is to provide a more generic architecture for the simulation, such that it can be applied in other domains like economy. It would then be possible to simulate different scenarios in a stock market for example.

References

1. Brandtstädter, J., Greve, W.: The aging self: Stabilizing and protective processes. Developmental Review **14** (1994) 52–80
2. Aamodt, A., Plaza, E.: Case-based reasoning : Foundational issues, methodological variations, and system approaches. AI Communications **1**(7) (March 1994)
3. Mosler, H.J.: Computersimulation sozialpsychologischer Theorien. Weinheim: Beltz (2000)
4. Althoff, K.D., Bach, K., Deutsch, J.O., Hanft, A., Mänz, J., Müller, T., Newo, R., Reichle, M., Schaaf, M., Weis, K.H.: Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. In Baumeister, J., Seipel, D., eds.: Proc. of the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007), Osnabrück, Germany, Berlin, Heidelberg, Paris, Springer Verlag (2007)
5. Bach, K., Reichle, M., Althoff, K.D.: A Domain Independant System Architecture for Sharing Experience. In Hinneburg, A., ed.: Proceedings of the LWA 2007: Lernen - Wissen - Adaptivität. (September 2007) 296–303

6. Brandtstädter, J.: Action perspectives on human development. In Lerner, R., ed.: Theoretical models of human development (Handbook of child psychology). Volume 1. 6th edn. Wiley, New York (2006) 516–568
7. Brandtstädter, J., Rothermund, K.: The life course dynamics of goal pursuit and goal adjustment: A two-process framework. Developmental Review **22** (2002) 117–150
8. Greve, W.: Maintanining personality: The active and adaptive self as core of individuality and personhood. In Greve, W., Rothermund, K., Wentura, D., eds.: The adaptive self. Personal continuity and intentional self-development. Hogrefe & Huber, Cambridge, MA (2005) 49–70
9. Greve, W., Wentura, D.: Personal and Subpersonal Regulation of Human Development: Beyond Complementary Categories. Human Development **50** (2007) 201–207
10. Newo, R., Müller, T., Althoff, K.D., Greve, W.: Learning to Cope with Critical Situations - A Simulation Model of Cognitive Processes using Multiagent Systems. In Hinneburg, A., ed.: Proceedings of the LWA 2007: Lernen - Wissen - Adaptivität. (September 2007) 159–164
11. Müller, T.: Simulation kognitiver Prozesse mit Multiagentensystemen. Master's thesis, Universität Hildesheim (2006)
12. Klügl, F.: Aktivitätsbasierte Verhaltensmodellierung und ihre Unterstützung bei Multiagentensimulationen. PhD thesis, Universität Würzburg (2000)
13. Epstein, J., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press, Washington D.C. (1996)
14. Anderson, J.R., Byrne, M.D., Douglas, S., Lebiere, C., Qin, Y.: An integrated theory of the mind. Psychological Review **111**(4) (2004) 1036–1050
15. Kieras, D.E.: EPIC Architecture Principles of Operation (2004)
16. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
17. Schneeberger, J.: Planen. In Oldenburg, ed.: Handbuch der Künstlichen Intgelligenz. 4 edn. G. Görz and C.-R. Rollinger and J. Schneeberger (2003) 491–515
18. Fikes, R.E., Nilsson, N.J.: STRIPS: a new approach to the application of theorem proving to problem solving. Artificial Intelligence **2** (1971) 189–208
19. Timm, I.J.: Dynamisches Konfliktmanagement als Verhaltenssteuerung Intelligenter Agenten. PhD thesis, Universität Bremen - Technologie-Zentrum Informatik (2004)
20. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons (June 2002)

CASIMIR – A Computational Architecture for Modeling Human Spatial Information Processing

Thomas Barkowsky

SFB/TR 8 Spatial Cognition, Enrique-Schmidt-Str. 5
28359 Bremen, Germany
barkowsky@sfbtr8.uni-bremen.de

Abstract

The computational cognitive architecture CASIMIR aims at modeling a wide range of phenomena, representations, and processes involved in human spatial reasoning and problem solving. While performing on a spatial problem, pieces of information stored in semantically organized long-term memory structures are retrieved and aggregated for being used to build up spatio-analogical working memory representations.

These working memory representations are specifically adapted to the task being performed. That is, from a structural point of view, they are highly economic in that they only require storage capacity and processing power in the order required by the amount and complexity of the spatial knowledge involved. Depending on the variety and the types of spatial knowledge dealt with, working memory representations gradually may be extended in a qualitative way thus forming flexible representation structures that may range from basic forms of spatial representations (e.g. reflecting spatial ordering information) up to fully fledged mental images involving a full range of visual features.

As a future perspective, CASIMIR will be extended to interact with external pictorial representation thus providing the option to apply the cognitive architecture in interactive intelligent assistance scenarios, for instance in spatial reasoning or planning tasks.

Bamberger Beiträge zur Wirtschaftsinformatik

Stand Oktober 7, 2009

Nr. 1 (1989) Augsburger W., Bartmann D., Sinz E.J.: Das Bamberger Modell: Der Diplom-Studiengang Wirtschaftsinformatik an der Universität Bamberg (Nachdruck Dez. 1990)

Nr. 2 (1990) Esswein W.: Definition, Implementierung und Einsatz einer kompatiblen Datenbankschnittstelle für PROLOG

Nr. 3 (1990) Augsburger W., Rieder H., Schwab J.: Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Nr. 4 (1990) Ferstl O.K., Sinz E.J.: Objektmodellierung betrieblicher Informationsmodelle im Semantischen Objektmodell (SOM) (Nachdruck Nov. 1990)

Nr. 5 (1990) Ferstl O.K., Sinz E.J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM)

Nr. 6 (1991) Augsburger W., Rieder H., Schwab J.: Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Nr. 7 (1991) Augsburger W., Rieder H., Schwab J.: Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVU / Ein Verarbeitungsmodell für eine modulare Bewertung von Kennzahlenwerten für den Endanwender

Nr. 8 (1991) Schwab J.: Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung

Nr. 9 (1992) Gross H.-P.: Eine semantiktreue Transformation vom Entity-Relationship-Modell in das Strukturierte Entity-Relationship-Modell

Nr. 10 (1992) Sinz E.J.: Datenmodellierung im Strukturierten Entity-Relationship-Modell (SERM)

Nr. 11 (1992) Ferstl O.K., Sinz E. J.: Glossar zum Begriffsystem des Semantischen Objektmodells

Nr. 12 (1992) Sinz E. J., Popp K.M.: Zur Ableitung der Grobstruktur des konzeptuellen Schemas aus dem Modell der betrieblichen Diskurswelt

Nr. 13 (1992) Esswein W., Locarek H.: Objektorientierte Programmierung mit dem Objekt-Rollenmodell

Nr. 14 (1992) Esswein W.: Das Rollenmodell der Organsiation: Die Berücksichtigung aufbauorganisatorische Regelungen in Unternehmensmodellen

Nr. 15 (1992) Schwab H. J.: EISREVU-Modellierungssystem. Benutzerhandbuch

Nr. 16 (1992) Schwab K.: Die Implementierung eines relationalen DBMS nach dem Client/Server-Prinzip

Nr. 17 (1993) Schwab K.: Konzeption, Entwicklung und Implementierung eines computergestützten Bürovorgangssystems zur Modellierung von Vorgangsklassen und Abwicklung und Überwachung von Vorgängen. Dissertation

Nr. 18 (1993) Ferstl O.K., Sinz E.J.: Der Modellierungsansatz des Semantischen Objektmodells

Nr. 19 (1994) Ferstl O.K., Sinz E.J., Amberg M., Hagemann U., Malischewski C.: Tool-Based Business Process Modeling Using the SOM Approach

Nr. 20 (1994) Ferstl O.K., Sinz E.J.: From Business Process Modeling to the Specification of Distributed Business Application Systems - An Object-Oriented Approach -. 1st edition, June 1994

Ferstl O.K., Sinz E.J. : Multi-Layered Development of Business Process Models and Distributed Business Application Systems - An Object-Oriented Approach -. 2nd edition, November 1994

Nr. 21 (1994) Ferstl O.K., Sinz E.J.: Der Ansatz des Semantischen Objektmodells zur Modellierung von Geschäftsprozessen

Nr. 22 (1994) Augsburger W., Schwab K.: Using Formalism and Semi-Formal Constructs for Modeling Information Systems

Nr. 23 (1994) Ferstl O.K., Hagemann U.: Simulation hierarischer objekt- und transaktionsorientierter Modelle

Nr. 24 (1994) Sinz E.J.: Das Informationssystem der Universität als Instrument zur zielgerichteten Lenkung von Universitätsprozessen

Nr. 25 (1994) Wittke M., Mekinic, G.: Kooperierende Informationsräume. Ein Ansatz für verteilte Führungsinformationssysteme

Nr. 26 (1995) Ferstl O.K., Sinz E.J.: Re-Engineering von Geschäftsprozessen auf der Grundlage des SOM-Ansatzes

Nr. 27 (1995) Ferstl, O.K., Mannmeusel, Th.: Dezentrale Produktionslenkung. Erscheint in CIM-Management 3/1995

Nr. 28 (1995) Ludwig, H., Schwab, K.: Integrating cooperation systems: an event-based approach

Nr. 30 (1995) Augsburger W., Ludwig H., Schwab K.: Koordinationsmethoden und -werkzeuge bei der computergestützten kooperativen Arbeit

Nr. 31 (1995) Ferstl O.K., Mannmeusel T.: Gestaltung industrieller Geschäftsprozesse

Nr. 32 (1995) Gunzenhäuser R., Duske A., Ferstl O.K., Ludwig H., Mekinic G., Rieder H., Schwab H.-J., Schwab K., Sinz E.J., Wittke M: Festschrift zum 60. Geburtstag von Walter Augsburger

Nr. 33 (1995) Sinz, E.J.: Kann das Geschäftsprozeßmodell der Unternehmung das unternehmensweite Datenschema ablösen?

Nr. 34 (1995) Sinz E.J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme - Entwicklung, aktueller Stand und Trends -

Nr. 35 (1995) Sinz E.J.: Serviceorientierung der Hochschulverwaltung und ihre Unterstützung durch workflow-orientierte Anwendungssysteme

Nr. 36 (1996) Ferstl O.K., Sinz, E.J., Amberg M.: Stichwörter zum Fachgebiet Wirtschaftsinformatik. Erscheint in: Broy M., Spaniol O. (Hrsg.): Lexikon Informatik und Kommunikationstechnik, 2. Auflage, VDI-Verlag, Düsseldorf 1996

Nr. 37 (1996) Ferstl O.K., Sinz E.J.: Flexible Organizations Through Object-oriented and Transaction-oriented Information Systems, July 1996

Nr. 38 (1996) Ferstl O.K., Schäfer R.: Eine Lernumgebung für die betriebliche Aus- und Weiterbildung on demand, Juli 1996

Nr. 39 (1996) Hazebrouck J.-P.: Einsatzpotentiale von Fuzzy-Logic im Strategischen Management dargestellt an Fuzzy-System-Konzepten für Portfolio-Ansätze

Nr. 40 (1997) Sinz E.J.: Architektur betrieblicher Informationssysteme. In: Rechenberg P., Pomberger G. (Hrsg.): Handbuch der Informatik, Hanser-Verlag, München 1997

Nr. 41 (1997) Sinz E.J.: Analyse und Gestaltung universitärer Geschäftsprozesse und Anwendungssysteme. Angenommen für: Informatik ’97. Informatik als Innovationsmotor. 27. Jahrestagung der Gesellschaft für Informatik, Aachen 24.-26.9.1997

Nr. 42 (1997) Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf S.: Application Objects – fachliche Bausteine für die Entwicklung komponentenbasierter Anwendungssysteme. Angenommen für: HMD – Theorie und Praxis der Wirtschaftsinformatik. Schwerpunkheft ComponentWare, 1997

Nr. 43 (1997): Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework - . Accepted for: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1997

Ferstl O.K., Sinz E.J.: Modeling of Business Systems Using (SOM), 2nd Edition. Appears in: P. Bernus, K. Mertins, and G. Schmidt (ed.): Handbook on Architectures of Information Systems. International Handbook on Information Systems, edited by Bernus P., Blazewicz J., Schmidt G., and Shaw M., Volume I, Springer 1998

Nr. 44 (1997) Ferstl O.K., Schmitz K.: Zur Nutzung von Hypertextkonzepten in Lernumgebungen. In: Conradi H., Kreutz R., Spitzer K. (Hrsg.): CBT in der Medizin – Methoden, Techniken, Anwendungen -. Proceedings zum Workshop in Aachen 6. – 7. Juni 1997. 1. Auflage Aachen: Verlag der Augustinus Buchhandlung

Nr. 45 (1998) Ferstl O.K.: Datenkommunikation. In. Schulte Ch. (Hrsg.): Lexikon der Logistik, Oldenbourg-Verlag, München 1998

Nr. 46 (1998) Sinz E.J.: Prozeßgestaltung und Prozeßunterstützung im Prüfungswesen. Erschienen in: Proceedings Workshop „Informationssysteme für das Hochschulmanagement“. Aachen, September 1997

Nr. 47 (1998) Sinz, E.J.:, Wismans B.: Das „Elektronische Prüfungsamt“. Erscheint in: Wirtschaftswissenschaftliches Studium WiSt, 1998

Nr. 48 (1998) Haase, O., Henrich, A.: A Hybrid Respresentation of Vague Collections for Distributed Object Management Systems. Erscheint in: IEEE Transactions on Knowledge and Data Engineering

Nr. 49 (1998) Henrich, A.: Applying Document Retrieval Techniques in Software Engineering Environments. In: Proc. International Conference on Database and Expert Systems

Applications. (DEXA 98), Vienna, Austria, Aug. 98, pp. 240-249, Springer, Lecture Notes in Computer Sciences, No. 1460

Nr. 50 (1999) Henrich, A., Jamin, S.: On the Optimization of Queries containing Regular Path Expressions. Erscheint in: Proceedings of the Fourth Workshop on Next Generation Information Technologies and Systems (NGITS’99), Zikhron-Yaakov, Israel, July, 1999 (Springer, Lecture Notes)

Nr. 51 (1999) Haase O., Henrich, A.: A Closed Approach to Vague Collections in Partly Inaccessible Distributed Databases. Erscheint in: Proceedings of the Third East-European Conference on Advances in Databases and Information Systems – ADBIS’99, Maribor, Slovenia, September 1999 (Springer, Lecture Notes in Computer Science)

Nr. 52 (1999) Sinz E.J., Böhnlein M., Ulbrich-vom Ende A.: Konzeption eines Data Warehouse-Systems für Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule“ im Rahmen der 29. Jahrestagung der Gesellschaft für Informatik, Paderborn, 6. Oktober 1999

Nr. 53 (1999) Sinz E.J.: Konstruktion von Informationssystemen. Der Beitrag wurde in geringfügig modifizierter Fassung angenommen für: Rechenberg P., Pomberger G. (Hrsg.): Informatik-Handbuch. 2., aktualisierte und erweiterte Auflage, Hanser, München 1999

Nr. 54 (1999) Herda N., Janson A., Reif M., Schindler T., Augsburger W.: Entwicklung des Intranets SPICE: Erfahrungsbericht einer Praxiskooperation.

Nr. 55 (2000) Böhnlein M., Ulbrich-vom Ende A.: Grundlagen des Data Warehousing. Modellierung und Architektur

Nr. 56 (2000) Freitag B, Sinz E.J., Wismans B.: Die informationstechnische Infrastruktur der Virtuellen Hochschule Bayern (vhb). Angenommen für Workshop "Unternehmen Hochschule 2000" im Rahmen der Jahrestagung der Gesellschaft f. Informatik, Berlin 19. - 22. September 2000

Nr. 57 (2000) Böhnlein M., Ulbrich-vom Ende A.: Developing Data Warehouse Structures from Business Process Models.

Nr. 58 (2000) Knobloch B.: Der Data-Mining-Ansatz zur Analyse betriebswirtschaftlicher Daten.

Nr. 59 (2001) Sinz E.J., Böhnlein M., Plaha M., Ulbrich-vom Ende A.: Architekturkonzept eines verteilten Data-Warehouse-Systems für das Hochschulwesen. Angenommen für: WI-IF 2001, Augsburg, 19.-21. September 2001

Nr. 60 (2001) Sinz E.J., Wismans B.: Anforderungen an die IV-Infrastruktur von Hochschulen. Angenommen für: Workshop „Unternehmen Hochschule 2001“ im Rahmen der Jahrestagung der Gesellschaft für Informatik, Wien 25. – 28. September 2001

Änderung des Titels der Schriftenreihe *Bamberger Beiträge zur Wirtschaftsinformatik* in *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* ab Nr. 61

Note: The title of our technical report series has been changed from *Bamberger Beiträge zur Wirtschaftsinformatik* to *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik* starting with TR No. 61

Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik

Nr. 61 (2002) Goré R., Mendler M., de Paiva V. (Hrsg.): Proceedings of the International Workshop on Intuitionistic Modal Logic and Applications (IMLA 2002), Copenhagen, July 2002.

Nr. 62 (2002) Sinz E.J., Plaha M., Ulbrich-vom Ende A.: Datenschutz und Datensicherheit in einem landesweiten Data-Warehouse-System für das Hochschulwesen. Erscheint in: Beiträge zur Hochschulforschung, Heft 4-2002, Bayerisches Staatsinstitut für Hochschulforschung und Hochschulplanung, München 2002

Nr. 63 (2005) Aguado, J., Mendler, M.: Constructive Semantics for Instantaneous Reactions

Nr. 64 (2005) Ferstl, O.K.: Lebenslanges Lernen und virtuelle Lehre: globale und lokale Verbesserungspotenziale. Erschienen in: Kerres, Michael; Keil-Slawik, Reinhard (Hrsg.); Hochschulen im digitalen Zeitalter: Innovationspotenziale und Strukturwandel, S. 247 – 263; Reihe education quality forum, herausgegeben durch das Centrum für eCompetence in Hochschulen NRW, Band 2, Münster/New York/München/Berlin: Waxmann 2005

Nr. 65 (2006) Schönberger, Andreas: Modelling and Validating Business Collaborations: A Case Study on RosettaNet

Nr. 66 (2006) Markus Dorsch, Martin Grote, Knut Hildebrandt, Maximilian Röglinger, Matthias Sehr, Christian Wilms, Karsten Loesing, and Guido Wirtz: Concealing Presence Information in Instant Messaging Systems, April 2006

Nr. 67 (2006) Marco Fischer, Andreas Grünert, Sebastian Hudert, Stefan König, Kira Lenskaya, Gregor Scheithauer, Sven Kaffille, and Guido Wirtz: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems, April 2006

Nr. 68 (2006) Michael Mendler, Thomas R. Shiple, Gérard Berry: Constructive Circuits and the Exactness of Ternary Simulation

Nr. 69 (2007) Sebastian Hudert: A Proposal for a Web Services Agreement Negotiation Protocol Framework . February 2007

Nr. 70 (2007) Thomas Meins: Integration eines allgemeinen Service-Centers für PC-und Medientechnik an der Universität Bamberg – Analyse und Realisierungs-Szenarien. Februar 2007

Nr. 71 (2007) Andreas Grünert: Life-cycle assistance capabilities of cooperating Software Agents for Virtual Enterprises. März 2007

Nr. 72 (2007) Michael Mendler, Gerald Lüttgen: Is Observational Congruence on μ-Expressions Axiomatisable in Equational Horn Logic?

Nr. 73 (2007) Martin Schissler: to be announced

Nr. 74 (2007) Sven Kaffille, Karsten Loesing: Open chord version 1.0.4 User's Manual. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 74, Bamberg University, October 2007. ISSN 0937-3349.

Nr. 75 (2008) Karsten Loesing (Hrsg.): Extended Abstracts of the Second *Privacy Enhancing Technologies Convention* (PET-CON 2008.1). Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 75, Bamberg University, April 2008. ISSN 0937-3349.

Nr. 76 (2008) G. Scheithauer and G. Wirtz: Applying Business Process Management Systems? A Case Study. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 76, Bamberg University, May 2008. ISSN 0937-3349.

Nr. 77 (2008) Michael Mendler, Stephan Scheele: Towards Constructive Description Logics for Abstraction and Refinement. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 77, Bamberg University, September 2008. ISSN 0937-3349.

Nr. 78 (2008) Gregor Scheithauer and Matthias Winkler: A Service Description Framework for Service Ecosystems. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78, Bamberg University, October 2008. ISSN 0937-3349.

Nr. 79 (2008) Christian Wilms: Improving the Tor Hidden Service Protocol Aiming at Better Performancs. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 79, Bamberg University, November 2008. ISSN 0937-3349.

Nr. 80 (2009) Thomas Benker, Stefan Fritzemeier, Matthias Geiger, Simon Harrer, Tristan Kessner, Johannes Schwalb, Andreas Schönberger, Guido Wirtz: QoS Enabled B2B Integration. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 80, Bamberg University, May 2009. ISSN 0937-3349.

Nr. 81 (2009) Ute Schmid, Emanuel Kitzelmann, Rinus Plasmeijer (Eds.): Proceedings of the ACM SIGPLAN Workshop on *Approaches and Applications of Inductive Programming* (AAIP'09), affiliated with ICFP 2009, Edinburgh, Scotland, September 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 81, Bamberg University, September 2009. ISSN 0937-3349.

Nr. 82 (2009) Ute Schmid, Marco Ragni, Markus Knauff (Eds.): Proceedings of the KI 2009 Workshop *Complex Cognition*, Paderborn, Germany, September 15, 2009. Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 82, Bamberg University, October 2009. ISSN 0937-3349.