Tutorials

Developing CLARION-based Agents with the New CLARION Library

Michael Lynch (lynchm2@rpi.edu)
Department of Language, Literature and Communication Rensselaer Polytechnic Institute, 110 Eighth Street,
Troy, NY 12180 USA
Telephone: (518) 276-3243 | Fax: (518) 276-4092

Nicholas Wilson (wilson3@rpi.edu)
Ron Sun (rsun@rpi.edu)
Cognitive Science Department Rensselaer Polytechnic Institute, Troy, NY 12180 USA

Previous tutorials on CLARION have focused mainly on presenting detailed introductions to the core theoretical con- cepts underlying the CLARION cognitive architecture. For this tutorial, in addition to providing a detailed introduction to the theory, we will also focus on giving participants hands-on experience using the new implementation of CLARION -- the CLARION Library, version 6.1 (written in C#). To that end, we will introduce guidelines for setting up and using basic and intermediate aspects of the library (with detailed walk-throughs for several simulation examples) as well as present several significant new features and en- hancements.

As CLARION is implemented in C#, participants will learn how they can employ the CLARION library on differ- ent operating systems using either the Visual Studio or Mono development environments. By the conclusion of this tutorial, participants should be equipped with the necessary foundation to begin developing CLARION-based agents for their own applications.

Tutorial Outline

A General Overview of CLARION (15 min.)

In this section, an introduction to cognitive architectures in general, and CLARION in particular, will be presented. CLARION will be compared to various other architectures and a brief discussion of some past and current applications of CLARION will be presented along with cognitive justifi- cations and implications.

CLARION is a unified, comprehensive theory of the mind based on two basic theoretical assumptions: representational differences and learning differences of two different types of knowledge --- implicit vs. explicit, among other essential assumptions and hypotheses.

In addition to these theoretical assumptions, CLARION is a cognitive architecture composed of four main subsystems: the Action-Centered Subsystem, the Non-Action-Centered Subsystem, the Motivational Subsystem, and the Meta-Cognitive Subsystem.

Action-Centered Subsystem Basics (30 min.)

In this section, some basic concepts of the Action-Centered Subsystem (ACS) will be presented. The structure and design of various aspects of the ACS, along with the learning mechanisms and the properties of the model, will be presented.

The Action-Centered Subsystem is used mainly for action decision-making. In the ACS, the top level generally con- tains simple “State→Action” rules, while the bottom level uses multi-layer perceptrons to associate states and actions. Reinforcement learning algorithms (usually with backpropagation) are used in the bottom level while rule learning in the top level is mostly “one-shot” and can be performed bottom-up (via “explicitation”) or independently (e.g., through linguistic acquisition). This section will focus on the representation for the top and bottom levels, and will detail bottom level learning and bottom-up rule extraction and refinement (RER).

Setting up and Using the ACS (30 min.)

For the first hands-on section of the tutorial, participants will be instructed on how to set up and install the CLARI- ON Library and are walked through a simple simulation example. In addition, several core principles necessary for interacting with the library will be outlined.

Working Memory and Goals (15 min.)

In this section we will discuss the theoretical underpin- nings for the working memory (WM) and the role that goals play in the decision-making processes of the ACS.

The working memory is conceived as a requisite structure within the ACS, whereas goals are stored within a top-level construct of the Motivational Subsystem, referred to as the Goal Structure (GS).

Setting up and Using the WM and GS (15 min.)

For this hands-on section, participants will be shown both the manual and action-oriented methods for setting-up and using the working memory and goal structure. In addition, asimple simulation example will be presented that demonstrates the use of working memory.

Drives and Meta-Cognitive Modules (30 min.)

This section will focus on the structure and design of the motivational (MS) and meta-cognitive (MCS) subsystems. In particular, the drives within the MS and various meta- cognitive modules within the MCS, will be described.

The Motivational Subsystem contains both low-level (physiological) and high-level (social) primary drives that take into account both environmental and internal factors in determining drive strengths. These drive strengths are re- ported to the Meta-Cognitive Subsystem, which regulates not only goal structures but other cognitive processes as well (e.g., monitoring, parameter setting, etc).

Setting up and Using Drives and Modules (30 min.)

For this hands-on section, participants will be shown both the manual and action-oriented methods for setting up and using the working memory and goal structure. In addition, a simple simulation example will be presented that demonstrates the use of these mechanisms.

Hands-On Practice Session #1 (15 min.)

In the final section before lunch, participants will be given the opportunity to further explore the CLARION Library and the simulations that were presented to this point. Partic- ipants will also be encouraged to ask any questions they may have with regard to using the library at this time.

The Non-Action-Centered Subsystem (45 min.)

Similar to the section on the ACS, this section will detail the Non-Action-Centered Subsystem (NACS). The structure and design of the various aspects of the NACS, along with the learning mechanisms and the theorems describing the properties of the model, will be presented.

The Non-Action-Centered Subsystem stores declarative (“semantic”) and episodic knowledge and is responsible for reasoning in CLARION. In the NACS, the top level con- tains simple associations while the bottom level involves nonlinear neural networks. Associative learning algorithms (e.g., backpropagation or contrastive Hebbian) are generally used in the bottom level whereas associations in the top lev- el are mostly learned “one-shot” (similar to the ACS).

Performing Reasoning using the NACS (15 min.)

For this hands-on section, participants will be given a very brief introduction to using the reasoning mechanism in the NACS. However, as the NACS is currently in the devel- opment stage, this demonstration will necessarily be brief.

Intermediate Aspects of the ACS (30 min.)

In this section we will discuss several intermediate con- cepts for the ACS. In particular, we will review the theoreti- cal considerations that govern IRL and Fixed rules.

IRL and Fixed rules are the other two forms of procedural knowledge (besides RER rules) that can be found in the top level of the ACS.

Setting-up and Using IRL and Fixed Rules (30 minutes)

For this hands-on section, participants will be shown how to do some basic customization using the CLARION Li- brary. In particular, we will show participants how to use C#’s *delegate* concept in order to quickly and easily create their own customized rules. In addition, a simple simulation that uses IRL rules will be presented.

Pre-Training, Tuning and Parameter Setting (15 minutes)

For this hands-on section, participants will be shown sev- eral methods for performing simple tuning and parameter setting operations in the CLARION Library.

Features and Plugins (15 minutes)

For this hands-on section, participants will be shown some of the useful features and plugins that are currently available as part of the CLARION Library.

Hands-On Practice Session #2 (30 min.)

In the final section of the day, participants will be given the opportunity to further explore the CLARION Library and ask any additional questions they may have.

Relevance for Cognitive Science

The CLARION cognitive architecture is well established with over 100 scientific papers and several books. CLARI- ON is particularly relevant to cognitive scientists because of its strong psychological plausibility and the breadth of its application to cognitive modeling and simulation. In CLAR- ION, each structure corresponds to a psychological pro- cess/capacity. CLARION-based models have been used to explain data as diverse as implicit learning, cognitive skill acquisition, inductive and deductive reasoning, meta- cognition, motivation, personality, and social simulations.

Presentation Details

Descriptions and demonstrations during the presentation will be provided using PowerPoint and the Visual Studio and Mono development environments.

Participants in the tutorial are encouraged to ask questions throughout the presentation to clarify any ideas described.

Sample Materials

- Sample slides: https://sites.google.com/site/clarioncognitivearchitectur e/presentations
- A complete technical specification of CLARION: http://www.cogsci.rpi.edu/~rsun/sun.tutorial.pdf
- A list of CLARION-related publications: http://www.cogsci.rpi.edu/~rsun/clarion-pub.html
- The current (6.1.0.6, C#) and previous (6.0.5, Java) versions of the CLARION Library: https://sites.google.com/site/clarioncognitivearchitectur e/downloads
- Other demonstration materials: See the "Tutorials" folder within the current CLARION Library software package

Scaling models of cognition to the real world: Complexity-theoretic tools for dealing with intractability

Iris van Rooij (i.vanrooij@donders.ru.nl)
Donders Institute for Brain, Cognition, and Behaviour
Radboud University Nijmegen

Johan Kwisthout (kwisthou@liacs.nl)
Leiden Institute of Advanced Computer Science
Leiden University

Introduction

A common theoretical obstacle encountered by computational- or rational-level models of cognition is that the cognitive capacities that they postulate appear to be computationally intractable (e.g., NP-hard or worse). Formally, this means that the computations that these models postulate consume an exponential amount of time. Informally, this means that the postulated computations do not scale in any obvious way to explain how cognitive capacities can operate in the real world outside the lab. How can cognitive scientists overcome this undesirable property of models of cognition? Over the last decade, several sophisticated complexity-theoretic techniques have been developed in theoretical computer science that can be utilized by cognitive modelers to systematically generate hypotheses about model changes or constraints that yield computational tractability without loss of the general applicability of the models. With this workshop we aim to bring these complexity-theoretic techniques to the attention of a broad audience of cognitive modelers and illustrate how they can be used to make cognitive models that scale to situations of real-world complexity.

Morning Session

In the morning session the tutorial organizers, Van Rooij and Kwisthout, will give a conceptual primer on computational complexity analysis in the context of cognitive modeling. The session will include a conceptual introduction to tractable cognitive modeling. Subsequently, they will review complexity-theoretic concepts (e.g., NP-hard, fixed-parameter tractability) and techniques (e.g., polynomial-time and parameterized reduction). Participants will have opportunity to practice the techniques via hands-on exercises (these can be done using paper and pencil). Also more controversial issues will be topic of discussion, such as the question to what extent intractable computations can be efficiently approximated by randomized or heuristic methods. The organizers aim for an interactive style of discussion.
Reading material: van Rooij, I. (2008). The Tractable Cognition thesis. *Cognitive Science, 32*, 939-984.

Afternoon session

In the afternoon session, four speakers will illustrate several applications of the concepts and techniques introduced in the morning session. Each application talk will consider a different type of model in a different cognitive domain.

What does (and doesn't) make deriving analogies hard?
Todd Wareham (Memorial University of Newfoundland) will present complexity analyses of Structure-Mapping Theory (SMT), assessing several conjectures in the literature about conditions that make analogy derivation under SMT feasible in practice.

Does recipient design make intention recognition tractable?
Mark Blokpoel (Radboud University) will consider Bayesian models of intention recognition and recipient design in the context of communication. He will demonstrate these models are NP-hard but also identify model constraints that yield computational tractability.

A tractability border in natural language semantics
Jakub Szymanik (University of Groningen) will discuss how ambiguity in natural language may be related to computational complexity. He will focus on logic-based models of quantifier expressions (e.g. `some', `more than') and will outline a tractability border between quantifier sentences.

Is managing multiple goals an intractable balancing act?
Daniel Reichman (Weizmann Institute of Science) will put forth the idea that people find it difficult to achieve multiple goals simultaneously because doing so entails solving computational intractable problems. He will outline approaches that can aid people in solving hard problems related to the attainment of multiple interrelated goals.

For more information about this tutorial, full details of the schedule, and extra materials, please refer to our website: http://tcs.dcc.ru.nl/iccm2012/

Design principles revisited: The continued design of the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS)

Troy Dale Kelley (troy.d.kelley6.civ@mail.mil)
U.S. Army Research Laboratory Human Research and Engineering Directorate Aberdeen Proving Ground,
Aberdeen MD 21005

Abstract

The Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS) is a production system based robotics controller based largely on the cognitive architecture the Adaptive Character of Thought-Rational (ACT-R). At the beginning of the research program a set of design principles were developed to aid in the design of the robotics system. These principles are discussed and revisited here.

Keywords: cognitive architectures, cognitive modeling, robotics

Introduction

In the last several decades, cognitive architectures have been designed around psychological principles in an attempt to reproduce the thought patterns of the human mind (Anderson & Lebiere, 1998). These cognitive architectures have made progress in modeling the human mind by using the production system architecture as a basis; however, they traditionally have had little interaction with the outside world which gives them limited functionality as real-world robotics controllers. The Sub-symbolic Robotic Intelligence Control System (SS-RICS) was developed using a production system as the central executive, as with traditional cognitive architectures, while also using sub-symbolic algorithms for perceptual processing. This allows SS-RICS to interact with the outside world. Additionally, these perceptual sub-symbolic algorithms are run in parallel with the production system, and mimic the parallel perceptual processing seen in the humans and animals. Additionally, the production system within SS-RICS is capable of shutting down certain algorithms (i.e. face recognition) if the current goal does not require the specified algorithms, thereby freeing up computational resources.

SS-RICS is part of an ongoing development within the U.S. Army Research Laboratory of a robotic control architecture that was inspired by computational cognitive architectures, primarily the Adaptive Control of Thought – Rational (ACT-R). SS-RICS combines symbolic and sub- symbolic representations of knowledge into a robotic control structure that allows robotic behaviors to be programmed in a production system format. The architecture is organized as a goal driven, serially executing, production system at the highest symbolic level; and a multiple algorithm, parallel executing, simple collection of algorithms at the lowest sub-symbolic level.

Five Development Principles

In order to guide the development of SS-RICS, five development principles were established in 2009 (Kelley et al. 2009).

1) The lowest level of perception includes algorithms running in a parallel fashion, while the highest levels of cognition are algorithms operating in serial fashion
2) At both the low levels and the high levels of cognition, the algorithms are relatively simple. It is the interaction, processing and results of simple algorithms which produce complex intelligent behavior.
3) Pre-programming SS-RICS is guided by the algorithms that are recognized as part of the human evolutionary process (for example, algorithms for edge detection, auto-focus of the eyes, pupil dilatation in different lighting environments). The pre-programming that is done should allow for the emergence of complex behavior, but not be the complex behavior itself.
4) Cognitive development within SS-RICS is principally about the reorganization of memory elements through increasing and decreasing their respective strengths.
5) Cognitive development and change can occur after allowing for specialized internal processing (i.e. dreaming) or after the necessary low level elements (i.e. features) are in place to allow for higher level symbolic extraction.

Developmental principles revisited

As defined in principle one, we have found an enormous value in running perceptual algorithms (motion tracking) in parallel with our other sub-symbolic algorithms (finding corners or gaps in a wall). This allows the higher levels of the system to turn off perceptual algorithms as the system becomes overloaded or runs out of memory; or allows us to pick and choose what functionality we are interested in, depending on the task. This can make the system very adaptive to certain tasks and make it able to use all of the available processing power for a given task. Additionally, we are currently running the cognitive process in serial but have found some utility in running multiple cognitive processes in parallel. In other words, the algorithm for the identification of an object is running in parallel with the algorithm for the identification of a specific face. The reader might ask – "what is the line between cognitive processes and perceptual processes" and it should be noted that this distinction can sometimes become blurred. It is not entirely clear that the identification of a face is, in fact, a cognitive process. We would rather reserve cognitive processes to strategy selection and problem solving so these lower levels processes should be, perhaps, pushed down to the parallel aspects of perception. This would make our goal stack relatively simple and would make the production system relatively simple to program.

As outlined in principle two, our algorithms, in general, remain relatively simple, except in some cases were we are using traditional AI techniques like Principle Component Analysis (PCA) or algorithms involved with Simultaneous Localization and Mapping (SLAM) (i.e. particle filters). While we strive to use cognitively plausible algorithms, traditional robotics algorithms can be seen as a means to an end for certain behaviors. For example, it is useful to use some SLAM algorithms to allow the robot to move from one room to the next, while more cognitively based algorithms like spreading activation can be used for object identification along the way.

Pre-programming algorithms based on evolutionary processes continues, and we feel we have adhered to principle three. However, when one considers the number algorithms humans are endowed with through evolution (i.e. color identification, sound localization, pupil dilatation based on light levels, object identification, object tracking, movement identification, contrast illumination.. and so forth), this can be a daunting task. Indeed, we have found this to be one of the more difficult and time consuming aspects of implementing an intelligent robotics system. It is important for any robotics engineer to realize that many of these low level algorithms need to be in place before any more complex behavior can emerge from an intelligent system. And while many of these algorithms seem intuitively simple (objectidentification) their implementation and interaction with other algorithms can create challenging developmental issues.

The reorganization of information as outlined in principles four and five continues to be an issue. We have not used proceduralization as implemented within ACT-R and would like to use this process to reduce the number and size of the goals developed by programmers. The struggle to write simple and powerful goals continues to be an issue, and we have looked at using subsumptive architectures to reduce the number and size of the goals. However, as I have pointed out in other articles, you cannot simulate extremely complex behaviors (i.e. playing chess) with a subsumptive architecture (Kelley and Long, 2010), and more powerful planning and strategy selection behaviors must still be written by hand or generated by some relatively complex process.

The abstraction and generalization of memories, as outlined in principle five, especially different types of memories (declarative, procedural and episodic) continues to be an area of continued research within SS-RICS. Interestingly, we have found some computational support for the concept of off-line processing or dreaming based on the speed of different memory retrievals. As part of our development of SS-RICS we found that real time retrieving memories for moving objects slows the system down too much, and it is better to try and remember everything that happens and consolidate these memories in order to speed retrievals. During consolidation, an off-line strategy to activate important memories is used and subsequent retrieval times can be greatly increased. Specifically, by increasing the strength of important memories using bottom-up activation, certain perceptions can then be selected by the cognitive system depending on their task relevance. This is more efficient than trying to identify everything that happens in real time. This would be an evolutionary argument for dreaming, which consolidates memories and speeds their retrieval times for the efficient execution of future recognitions.

Conclusions

SS-RICS continues to be undergo a complex and challenging development cycle, where new developments occur each day. We feel we have adhered to our original design guidelines and will continue to use these guidelines to further the development of the system.

References

Anderson, J. R., and Lebiere, C. The Atomic Components of Thought: Erlbaum, 1998.

Kelley, T.D., Avery, E., Long L.N., & Dimperio, E. (2009). A Hybrid Symbolic and Subsymbolic Intelligent System for Mobile Robots. Presented at the AAIA Infotech@Aerospace Conference, AAIA-2009-1949, Seattle, WA, April 6-9.

Kelley, T. D., & Long, L.N. (2010). Deep Blue cannot play checkers: The need for generalized intelligence for mobile robots. Journal of Robotics, 2010, 1-8.

Understanding cognitive processes through language use

Thora Tenbrink (tenbrink@uni-bremen.de)
SFB/TR 8 Spatial Cognition, University of Bremen, Germany

Introduction

How can cognitive processes be accessed and understood sufficiently to enable reliable computational models? One established way of addressing internal processes is to analyze their external representations, most prominently natural language produced along with cognitively complex tasks (Ericsson & Simon, 1993). The aim of this tutorial is to familiarize both young and experienced researchers with the systematic and linguistically informed analysis of language data collected in order to substantiate cognitive models. The method of *Cognitive Discourse Analysis* (CODA) (Tenbrink & Gralla, 2009; Tenbrink, 2010) will be introduced, which uses linguistic methods and insights to address research questions in cognitive science. One main aim is to identify particular types of linguistic patterns in the collected data that are likely to point to specific cognitive processes. The outcome of a CODA-based analysis is a validated account of systematic cognitive processes feeding directly into subsequent computational cognitive modelling.

Methods that employ language to address research questions in cognitive science range from psychological via psycholinguistic approaches to linguistic discourse analysis. In spite of their fundamental diversity, such methods share the basic view that patterns in language are systematically related to patterns of thought (Chafe, 1998). A prominent feature and aim of the CODA framework is to identify relevant types of linguistic patterns that are likely to point to specific cognitive processes in diverse scenarios. Systematic accounts of recurring patterns of thought and prominent conceptualizations provide a substantial prerequisite for cognitive modelling approaches of any kind.

CODA can be employed to enhance the analysis of think-aloud protocols and retrospective reports for the identification of (internal) *cognitive processes* (Ericsson and Simon, 1993; Tenbrink, 2008). Conventionally, the focus in this kind of analysis lies on the *content* of verbal data, addressing those aspects (e.g., particular thought processes or strategies) that the speakers are themselves aware of. The content-based inspection of verbal reports, particularly if carried out by experts in the problem domain and set against a substantial theoretical background (Krippendorff, 2004), often leads to well-founded specific hypotheses about the cognitive processes involved. The detailed systematic analysis of linguistic features and structures in CODA provides a particularly sound basis for using the language data as evidence (e.g., Hölscher et al., 2011; Tenbrink et al., 2011; Tenbrink & Seifert, 2011; Tenbrink & Wiener, 2009). CODA is used to gain insights into generalizable cognitive phenomena that go beyond conscious reflection by individual speakers, and that may not necessarily be directly observable in linguistic content. Speakers may not be aware of the cognitive structures that are reflected in particular ways of framing a representation linguistically. Further- more, they may not be consciously aware of the underlying network of options (Tenbrink & Freksa, 2009) that allows for a range of linguistic choices beside their own, which emerges more clearly by considering a larger data set collected under controlled circumstances. According to previous research in cognitive linguistics and discourse analysis (e.g., van Dijk, 2008), linguistic features such as the verbal representation of semantic domains reflected in ideational networks, lexical omissions and elaboration, presuppositions, hesitation and discourse markers, and the like all indicate certain conceptual circumstances; these are related to the current cognitive representations in ways that distinguish them from other options available in the network. In particular, the chosen linguistic options reflect what speakers perceive as sufficiently relevant to be verbalized, as well as the information status assigned to the diverse parts of the verbalization.

Besides building on established insights about the significance of particular linguistic choices, validating evidence for the relationship between patterns of language use and the associated cognitive processes can be gained by triangulation, i.e., the combination of linguistic analysis with other types of evidence such as behavioral performance data. In these combined ways, data collected in empirical studies serve as validated evidence for subsequent computational modelling of complex cognitive processes.

Format and schedule

This tutorial is designed to cover a half day (three hours). Rather than offering primarily theoretical insights, the tutorial will take the participants' current or intended projects as a starting point to address the following issues, supplemented wherever suitable by practical exercises.

Motivation: How can language data serve as empirical evidence for cognitive modelling?
Data collection: What kinds of issues need to be considered in the light of actual research purposes?
CODA based analysis (main part): Systematic data annotation and interpretation, substantiated by linguistic insights.
Triangulation and systematization: How can the insights gained from language be complemented by other types of empirical data and systematized for modeling purposes?
In contrast to previous offerings, this tutorial will focus on the systematic identification of the cognitive steps and principles that can be fed into computational models.

Target audience information

There are no particular prerequisites for attending this tutorial. It will be open for researchers in cognitive science at any point in their career, ranging from graduate students to established experts in cognitive modelling.
Linguistic knowledge or expertise is welcome but not a prerequisite for this tutorial. Participants are encouraged to bring examples of their own collected natural language data as handouts or on their computers. Sample data collected in relevant scenarios will be discussed, tailored to the participants' current focus of interest.

Acknowledgments

Funding by the DFG, SFB/TR 8 Spatial Cognition, I6- [NavTalk] is gratefully acknowledged. Further thanks go to my collaborators across diverse research projects that continue to support the development of the methodology.

References

Chafe, W. (1998). Language and the Flow of Thought. In M. Tomasello (ed.), *The New Psychology of Language. Cognitive and Functional Approaches to Language Structure*. Mahwah: Lawrence Erlbaum.

van Dijk, T.A. (2008). *Discourse and Context. A Sociocognitive Approach.* Cambridge: Cambridge University Press.

Ericsson, K.A., & Simon, H.A. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press.

Hölscher, C., Tenbrink, T., & Wiener, J. (2011). Would you follow your own route description? *Cognition 121,* 228- 247.

Krippendorff, K. (2004). *Content Analysis: An Introduction to its Methodology* (2nd ed.). London: Sage.

Tenbrink, T. (2008). The verbalization of cognitive processes: Thinking-aloud data and retrospective reports. In W. Ramm & C. 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.

Tenbrink, T. (2010). CODA: Kognitive Diskursanalyse. In: E. Ruigendijk, T. Stolz, & J. Trabant (Hrsg.), *Linguistik im Nordwesten: Beiträge zum 1. Nordwestdeutschen Linguistischen Kolloquium*. Bochum: Brockmeyer.

Tenbrink, T., Bergmann, E., Konieczny, L. (2011). Wayfinding and description strategies in an unfamiliar complex building. In L. Carlson, C. Hölscher, & T.F. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society. Tenbrink, T., & Freksa, C. (2009). Contrast sets in spatial and temporal language. *Cognitive Processing 10 Supplement 2*, S322-S324. Tenbrink, T., & Gralla, L. (2009). Accessing complex cognitive processes via linguistic protocol analysis. In U. Schmid, M. Ragni, M. Knauff (Eds.): *Proceedings of the KI 2009 Workshop Complex Cognition*, Paderborn, Germany, September 15, 2009.

Tenbrink, T., & Seifert, I. (2011). Conceptual Layers and Strategies in Tour Planning. *Cognitive Processing 12:1*, 109–125.

Tenbrink, T., & Wiener, J. (2009). The verbalization of multiple strategies in a variant of the traveling salesperson problem. *Cognitive Processing 10*:2, 143-161.