



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document

D-97-08

**Distributed Cognitive Systems
Proceedings of the VKS'97 Workshop**

**Christoph G. Jung, Klaus Fischer, and Susanne
Schacht**

December 1997

**Deutsches Forschungszentrum für Künstliche Intelligenz
GmbH**

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341

**Distributed Cognitive Systems
Proceedings of the VKS'97 Workshop**

Christoph G. Jung, Klaus Fischer, and Susanne Schacht

DFKI-D-97-08

This work has been supported by a grant from The Federal Ministry of Education, Science, Research, and Technology (FKZ ITWM-95004).

© Deutsches Forschungszentrum für Künstliche Intelligenz 1998

This work may not be copied or reproduced in whole or part for any commercial purpose. Permission to copy in whole or part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the Deutsche Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to Deutsches Forschungszentrum für Künstliche Intelligenz.
ISSN 0946-0098

Distributed Cognitive Systems

Proceedings of the VKS'97 Workshop

Christoph G. Jung¹ , Klaus Fischer² , and Susanne Schacht³

¹GK Kogwiss. & MAS Group, FB Informatik, Univ. des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, Germany, jung@ags.uni-sb.de

²MAS Group, DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, kuf@dfki.de

³Computational Linguistics Research Group, Universität Freiburg, Friedrichstr. 50, 79098 Freiburg im Breisgau, sue@coling.uni-freiburg.de

ABSTRACT To intensify the contacts between *cognitive scientists* and researchers in *Distributed Artificial Intelligence*, we have organised the 1st German workshop on *Distributed Cognitive Systems*, VKS'97, in association with the 21st German Conference on *Artificial Intelligence*, KI'97. The workshop has been held in Freiburg on the 11th of September 1997. This document comprises the contributions of various research fields under the aspect of *social, intelligent agents* and gives an impression of future, inter-disciplinary collaboration.

Contents

1	Distributed Cognitive Systems	4
1.1	Introduction	4
1.2	Modalities of the Workshop	5
1.3	Results	6
2	Segmentation of Spoken Language: Aspects of Human and Machine Processing	7
2.1	References	8
3	Cooperative Rationality: A Cognitive Model of Social Conflicts	10
3.1	C-situations	10
3.2	Matrix games	11
3.3	Rationality	12
3.4	α -, β -, and γ -rationality	13
3.5	Purposes of the model	13
3.6	Model lay-out	14
3.7	Description of a matrix configuration	14
3.8	Goals and productions	15
3.9	References	17
4	Agentenmodelle zum Umgang mit vollständig erneuerbaren Gemeingütern	19
4.1	Einleitung	19
4.2	Eine formale Definition von Umweltgemeingütern und “Rationalität”	20
4.3	Das Saarbrücker Szenario	21
4.4	Agentenmodell	24
4.5	Indizien	25
4.6	Weitere Modelleigenschaften	26
4.7	Simulationsläufe	26
4.8	Anhang	28
5	Towards the Bounded Optimal Agent Society	32
5.1	Introduction	32
5.2	The Bounded Optimal Agent Society	33
5.2.1	Definition	33

	3
5.2.2 A Hierarchical Model of Resource Control	34
5.3 Engineering Constructs	35
5.3.1 Resource Control in Structured Agent Societies	35
5.3.2 Resource Control in a Layered Agent Architecture	37
5.4 Cognitive Concern	38
5.4.1 Macro-Level	38
5.4.2 Micro-Level	39
5.5 Conclusion and Future Work	39
5.6 References	39
6 Roboter und Computer spielen Fußball	41
6.1 Das Anliegen	41
6.2 RoboCup-97: Die erste Weltmeisterschaft im Roboter-Fußball	42
6.2.1 Der Wettbewerb	42
6.2.2 Unser Weg ins Finale	43
6.3 Die Realisierung	43
6.3.1 Die Simulationsumgebung	43
6.3.2 Die Implementierung unserer Mannschaft	46
6.4 BDI: Theorie unserer Architektur	47
6.4.1 Grundlagen für den Erfolg unserer Mannschaft	48
6.4.2 Weitere Vorhaben	49
6.5 Die Organisation des RoboCup	49
6.6 Bedeutung des RoboCup	49
6.7 Web-Seiten mit Informationen	50
6.8 References	50

Distributed Cognitive Systems

Christoph G. Jung¹, Klaus Fischer², and Susanne Schacht³

1.1 Introduction

The VKS'97 workshop on *Distributed Cognitive Systems* has been addressed to researchers from the areas of *Distributed Systems*, *Distributed Artificial Intelligence* and *Cognitive Science*. The goal of the event has been to regard the notions of an *intelligent agent* and a *society of agents* from different perspectives and to build up a platform of fruitful exchange between those three communities.

Distributed Systems focus on technical questions around interacting computations and mobile objects. On the other hand, cognitive scientists examine by empirical studies and cognitive modelling the structure and the interaction of intelligent processes and entities in nature. The implementation and the evaluation in formal systems count as major indicators for the adequacy of a cognitive theory. Between those complementary positions, topics of *Distributed Artificial Intelligence* can be located. DAI researchers particularly investigate multi-agent systems, i.e. societies of autonomous agents that exhibit a certain degree of local intelligence for the solution of global problems.

Let us describe the core of interest as *Distributed Cognitive Systems*. To integrate cognitive models with the latest technical concepts into applications is an engrossing topic of research that can be driven by all the three communities. By its realisation, it is possible to derive additional implications for one's specific area. To proceed with this respect, a tight teamwork is required. Whereas the engineers of Distributed Systems and DAI could recognise foundations of new and more efficient information processing by obtaining results from cognitive science, the goals of computer science also do not prevent their developments from counting as cognitive plausible models for certain achievements of nature.

The workshop and its proceedings should deliver an integrative contribution. Therefore, we have installed two focal points: an engineering one with questions to the technical implementation of multi-agent systems and a cognitive one that deals with cognitive aspects of distributed environments. The contributors were asked to emphasise the interests of the other research areas in their own work. Such a requirement of interdisciplinary thinking seems to be a good goad to strip off the often tight corset of one's own area. A plenum discussion at the very end of the workshop has comprehended the overall impression at hand of the given presentations to uncover the possibilities of an enforced contact between the researchers.

DAI with Cognitive Orientation

The engineering focus has been dedicated to Distributed Artificial Intelligence and Distributed Systems. The requested descriptions are located in the areas of agent theories, architectures, and languages, mobility, distributed and object-oriented technologies, decision making (such as planning), societies of agents, cooperation and conflict management, interlingua and ontolingua, and the application of multi-agent technology in practical domains. Because of the required bias, contributions

¹GK Kogwiss. & FB Inform., Univ. des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, Germany, jung@ags.uni-sb.de

²Multi-Agent Systems Group, DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, kuf@dfki.de

³Computational Linguistics Research Group, Universität Freiburg, Friedrichstr. 50, 79098 Freiburg im Breisgau, sue@coling.uni-freiburg.de

should discuss the aspects of cognitive orientation, in particular. We were especially interested in cognitive-plausible constraints.

Cognitive Aspects of Distributed Environments

Adequate, cognitive research for the VKS'97 workshop covers the agent-centered area of cognitive architectures and modelling including their structure, their computational foundation, the applied constraints, their models of memory, the role of attention, and their learning facilities. It also covers the society-centered approach of empirical and theoretical work on control theory, game theory, and decision theory. Finally, we were also interested in logical foundations of multi-agent systems, such as the concept of intention coupled with speech act theory. To satisfy the framework, we requested the demonstration of approaches to agentify, distribute, or integrate efficient problem solving.

Common areas of interest

The common topics that are relevant for all three communities solidify from the previous descriptions. On the agent level, these are questions of resource-adaptivity, concurrency and distribution, meta-reasoning, representation and computational languages, and decision and control theory. On the social level, these are especially structures as examined by game theory and cooperation models.

1.2 Modalities of the Workshop

The auspices of the workshop have been represented by the subdivisions 1.1.6, *Distributed Artificial Intelligence*, and 1.0.1, *Cognitive Science*, of the German Association for Computer Science (GI). Contributions are original works and have been presented to an extent of 30 minutes including 10 minutes of discussion on Thursday, the 11th of September 1997, in Freiburg.

Time	Event
15:00	Klaus Fischer, Christoph G. Jung, and Susanne Schacht Introduction
15:05	Martin Beckenkamp and Axel Ostmann Agentenmodelle zum Umgang mit vollständig erneuerbaren Gemeingütern
15:35	Break
16:00	Christian Gerber and Christoph G. Jung Towards the Bounded Optimal Agent Society
16:30	Markus Hannebauer RoboCup: Agents Playing Soccer
17:00	Plenum Discussion: Common Perspectives for Distributed Cognitive Systems ?

Unfortunately, two presentations, Susanne Jekat about "Segmentation of Spoken Language: Aspects of Human and Machine Processing" and Werner H. Tack about "Cooperative Rationality: A Cognitive Model of Social Conflicts" could not take place, but are included into these proceedings.

Organising and Program Committee

The organising committee has been constituted from the three editors of these proceedings. The program committee consisted of the following researchers:

Hans-Dieter Burkhardt
Institut für Informatik, Humboldt-Universität zu Berlin

Stefan Fünfroeken
FB Informatik, Technische Universität Darmstadt

Gerhard Kraetzschmar
Fakultät für Informatik, Universität Ulm

Ralf Kühnel
Institut für Informatik, Humboldt-Universität zu Berlin

Jörg P. Müller
Mitsubishi Electric Europe, London

Jürgen Müller
FB Informatik, Universität Bremen

Gerhard Strube
Abteilung Kognitionswissenschaft,
Institut für Informatik und Gesellschaft,
Albert-Ludwigs-Universität Freiburg

Werner H. Tack
FB Psychologie, Universität des Saarlandes

Gerhard Weiss
Institut für Informatik, Technische Universität München

1.3 Results

Our main impression from the workshop and the discussions around that event is that Distributed Cognitive Systems, although being a particular focus of three research areas, still represent a very heterogeneous field: only a few researchers could immediately identify with our setting and therefore decide to participate actively. Surprisingly, this is quite opposed to the size of the audience: around 50 people from very different backgrounds obviously enjoyed the presentations. This proved that there is indeed a common ground of discussion and understanding. Researchers are willing to adopt new perspectives for their own field and quite a few collaborations, such as with respect to the role of emotions and rationality as emphasised by Beckenkamp and Ostmann (Chapter 4) and Gerber and Jung (Chapter 5), have reportedly given rise by the workshop. Due to the variety of methodologies that are applied, this is a very promising starting point. This could be furthermore enhanced by emphasising the role of social sciences in Distributed Cognitive Systems because of its being currently underrepresented. Encouraged by the final success, we hope to set up a continuing series of events that elaborates on this aspect and extends the fruitful focus as set by the VKS'97 workshop.

Segmentation of Spoken Language: Aspects of Human and Machine Processing

Susanne J. Jekat¹

Extended Abstract

In human and machine speech processing, segmentation above word boundary is necessary for different reasons:

- limited memory capacity,²
- separation of relevant message(s) from irrelevant surface phenomena,
- separation of minor from main information,
- restructuring of ill-formed input,
- evaluation of contradictory information delivered within a single turn,
- recognition of topic hierarchy,
- generation of units fitting into and manageable by the processing system.

In written language there is a pre-segmentation by different conventionalized cues as for example punctuation and subdivision of the text into sentences, paragraphs etc.. This pre-segmentation is based on a structural analysis of the usually well- formed surface of written texts. As for spoken texts, performance phenomena such as false starts or repairs do not give access to structural analysis like that of written language. It is necessary then to identify meaningful functional units in order to recognize central information and to evaluate contradictive messages.³

Parting from the hypothesis that segmentation procedures and the resulting functional units principally depend on the actual task, we suppose here that human segmentation of spoken texts for the purpose of interpretation differs from segmentation within other tasks. Additionally, machine interpretation, although not dedicated to replace the human interpreter but rather designed for applications within limited domains, may profit from insight into these mechanisms of human processing.

Some examples for the necessity of identifying functional units within the translation process are listed in [1].

1. The turn contains important information and redundant performance phenomena, which have to be separated, because redundant phenomena need not to be translated: 'But uh ehm what's see. The Monday or Tuesday and Wednesday on the seventh or eighth of June I have free.'

¹FB Informatik, Universität Hamburg, Vogt-Kölln-Str. 30, D-22527 Hamburg, Phone: 040-54715-520, Fax: 040-54715-515, jekat@informatik.uni-hamburg.de

²Some of the items listed here are only valid for either human or machine processing. E.g. limited capacity for the storage of everyday conversations or single turns is not supposed to be a real problem of machine processing.

³Meaningful functional units are often corresponding to speech acts (cf. [3]), their description can be specified by the description of their linguistic realization and they can be adapted to a certain domain (for example dialogue acts in the VERBMOBIL project, cf. [9]).

2. The turn contains contradictory information, so information has to be weighted or modified: 'Well I, I'm certainly free Monday afternoon. If it's not too late.'
3. The turn contains two proposals that first are introduced as equivalent but at the end of the turn one of the proposals is presented as the preferred one which for adequate translation has to be identified: 'I could go any time between Wednesday and Friday eh so it could be either Wednesday and Thursday, that's May eighteenth and nineteenth, or eh Thursday and Friday, that's nineteenth and the twentieth. But I must say, I would prefer Wednesday and Thursday, because it's not directly before the holidays.'

At present, still very little is known about special processing mechanisms of human interpreters. [5]:

Although simultaneous interpreting⁴ has received increasing attention as an object of study (...), unfortunately little of it has been in the form of reliable experimental research (...). Consequently, precious little is known about the differences between experienced and novice interpreters' performance, and nothing at all is known about any possible differences in the way they go about carrying out the task.

Therefore our research in segmentation for interpretation starts with the investigation of structuring devices rendered by speech itself as there are

- prosodic cues,
- lexical cues,
- syntactic cues,
- pragmatic cues (cf. [7]).

Considering time pressure and default strategies (e.g. the strategy to expect a prototypical linguistic surface and to choose the prototypical meaning of a text as prominent meaning) as relevant factors within the segmentation process, we will try to verify the following hypotheses:

1. Cues for segmentation are ordered hierarchically according to saliency.
2. Prosodic cues are very important for speech segmentation and help to identify meaningful functional units.

In the first part of our paper we will examine structuring signals of speech under the aspect of perception and processing. In the second part we will compare the differences between monolingual processing and processing for interpretation. The existence of principal differences is already indicated by research in cerebral activity, see e.g. [4] and [6]. In the third part of our paper we will deduce relevant mechanisms for automatic speech segmentation from cue evaluation and the supposed cue hierarchy. Automatic segmentation proposed here will be dedicated to support transfer within an automatic interpreting system.

2.1 References

- [1] J.W. Amtrup and S.J. Jekat. *Segmentation of spoken language for NLP*. In L. Dreschler-Fischer and S. Pribbenow, (eds.), *KI-95 Activities: Workshops, Posters, Demos*,
- [2] Gesellschaft f-ur Informatik e.V., 1995, pages 298—299.

⁴From our point of view Dillinger's statement can be generalized to conference and dialogue interpreting if understood as referring to the state of the art in translation studies.

- [3] J.L. Austin. *How to do things with words*. Oxford University Press, 1962
- [4] V. Darò. *Non-linguistic factors influencing simultaneous interpretation*. In S. Lambert and B. Moser-Mercer, (eds.), *Bridging the Gap. Empirical research in simultaneous interpretation*. John Benjamins, Amsterdam, 1994, pages 249—269.
- [5] M. Dillinger. *Comprehension during interpreting: What do Interpreters know that Bilinguals don't?*. In S. Lambert and B. Moser-Mercer, (eds.), *Bridging the Gap. Empirical research in simultaneous interpretation*. John Benjamins, Amsterdam, 1994, pages 155—189.
- [6] F. Fabbro, L. Gran, B. Basso, A. Bava. *Cerebral lateralization during simultaneous interpreting*. volume 39 in *Brain and Language*. 1990, pages 69—89.
- [7] S.J. Jekat. *Segments for Interpretation*. In L. Dreschler-Fischer and S. Pribbenow, (eds.), *KI-95 Activities: Workshops, Posters, Demos*,
- [8] Gesellschaft f-ur Informatik e.V., 1995, pages 307—308.
- [9] Jekat, S.J., Klein, A., Maier, E., Maleck, I., Mast, M., and Quantz, J. *Dialogue Acts in VER-BMOBIL*. volume 65 in *VM-Report* April 1995.

Cooperative Rationality: A Cognitive Model of Social Conflicts

Werner H. Tack¹

ABSTRACT

Cognitive science in general and cognitive modelling in particular are sometimes confronted with the reproach of a much too narrow concentration on isolated single agents — be it artificial or human individuals — and a corresponding neglect of social phenomena emerging from the interaction between several agents. The true core of this criticism is that indeed the majority of cognitive models simulate cognitive functions and performances of single agents, the unjustified inference is that this would be the consequence of some kind of principal deficit of the cognitive approach. There is good hope that this prejudice will loose some of its strength in the future. This paper aims at speeding up this process by presenting an example of the usefulness of cognitive models in the field of social processes.

The kind of social processes we are trying to understand and to analyse are those which can lead to the resolution of a conflict of interests by finding an acceptable compromise. They presuppose a situation with several agents, each of whom can choose one out of a set of available options. Furthermore, there is mutual result dependency: The final result of decisions is not determined by the single choice of a single agent but the consequence of the choices of all the partners. These partner agents have individually different preferences on those final outcomes which they regard as possible and therefore take into account when deciding which option to choose.

3.1 C-situations

C-situations [11, 12, 13, 14] are formal descriptions of the relevant aspects of those social situations which can lead to conflicts of interest. Such situations require a set of different agents

- having several options at their disposal,
- taking into regard several possible outcomes resulting from individual decisions of all the agents involved, and
- pursuing different interests. The interests are interpreted as individual asymmetric preference relations on the set of possible outcomes taken into account.

Thus, a *C-situation* is a system

$$\langle N; A_1, A_2, \dots, A_n; B_1, B_2, \dots, B_n; P_1, P_2, \dots, P_n \rangle$$

with

1. a finite set N of agents with cardinality $|N| = n$,
2. for each agent $i \in N$ the set A_i of available options he (or she) can choose from, and — after introducing the set of resulting outcomes as complete combinations of options $A = \bigotimes_{i \in N} A_i$ —

¹AE Kognitive Modellierung, GB Psychologie, Universität des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, werner@cops.uni-sb.de

3. the sets $B_i \subseteq A_i$ of outcomes taken into account by $i \in N$,
4. the asymmetric preference relations $P_i \subset B_i \times B_i$.

A C-situation is *complete*, if every agent is able to always take into account all possible outcomes:

$$\forall i \in N : B_i = A$$

It is named *measurable* if for every agent there exists a real-valued utility function such that the agent prefers outcome a over b if and only if a has — on his utility function — a higher value than b . Formally:

$$\forall i \in N : \exists u_i : A \rightarrow \mathfrak{R} : a, b \in A : a P_i b \leftrightarrow u_i(a) \geq u_i(b)$$

A *complete* and *measurable* C-situation is what game theorists call a *game in normal form* or — a little bit shorter — a *normal-form game* (see e.g. [10]). As such, it can be represented by omitting the B_i (which are all equal to A) and by substituting the individual utility functions u_i for the original preference relations P_i as

$$\langle N; A_1, A_2, \dots, A_n; u_1, u_2, \dots, u_n \rangle$$

3.2 Matrix games

The ‘matrix game’ is an experimental three-person game which has been used in several experiments on cooperative conflict resolution [8, 13]. As a rather simple and transparent illustration of a complete and measurable C-situation, it will be used here as an example for introducing the concept of cooperative rationality and for developing a corresponding cognitive model.

Imagine that the agents taking part in a matrix game are three human subjects sitting at a roundtable such that each of them has a right-hand partner and a left-hand one. Each one is confronted with a display presenting three tilted 3×3 -matrices. We take one of the subjects and call him ‘center player (C)’. He can choose one and only one of the three matrices labelled u , v and w . His or her left-hand partner (L) has the choice between the three rows going through all three matrices from upper left to lower right and labelled r , s and t . The right-hand partner’s (R) options are x , y and z , the rows from upper right to lower left.

If — as for an example — L decides to take t , only the nine outcomes represented by the matrix-cells $t-u-x$, $t-u-y$, $t-u-z$, $t-v-x$, $t-v-y$, $t-v-z$, $t-w-x$, $t-w-y$, and $t-w-z$ are still available. If — additionally — R takes x , only three outcomes are still feasible, and if C decides on v , the final outcome is fixed at position $t-v-x$.

Each combination of three options — one from each agent’s option set — determines an outcome which is represented by one of the 27 matrix cells describing an individual game. Each cell representing an outcome contains three numbers in a triangular arrangement: One on top and the others bottom-left and bottom-right. The number on top is a number of points given to C if this outcome is the result determined by the final decisions. The bottom-left number shows the points for L and the bottom right number those for R . The preference relations of the agents are given by the assumption that everybody prefers more points over less. In experiments, this assumption is supported by the fact that experimental subjects get a final payment proportional to the total number of points they collect in a series of experiments.

The game situation is a cooperative one. Human partners can communicate with each other. They can make proposals on who should choose which option, they can accept or reject an offer, they can give reasons for what they suggest or how they react on their partners’ proposals, they can come to agreements and they can finally make the choices they agreed upon — or they can break their promises.

3.3 Rationality

What can be called a ‘*rational decision*’ or a ‘*rationally acceptable compromise*’ in such a kind of situation? The term ‘*rationality*’ has a lot of different readings. There is the distinction between

- ‘*cognitive rationality*’, which deals with the relation between the outer world and our inner knowledge, beliefs, and assumptions, and there is
- ‘*instrumental rationality*’, dealing with the means-end-relations between our goals and intentions on one side and our actions or choices on the other [15].

In the field of instrumental rationality we find ‘*individual rationality*’ and ‘*strategic rationality*’.

Individual rationality is relevant when the outcome of an individual decision is a function of this decision (and — eventually — of some unknown states of nature resulting in an outcome which is a lottery with different final events and corresponding probabilities). This excludes all cases in which the outcome depends on the decisions of different agents, and in which the protagonistic person can have regards to goals, interests and intentions of others. Such situations are the stage for ‘*strategic rationality*’.

From the concept of individual instrumental rationality we are accustomed to the idea of a rationality which has something to do with optimising the consequences of one’s own behaviour. But that does not work in a C-situation requiring strategic rationality because consequences follow from the decisions of *all* partners involved in the given situation.

The more or less ‘*classical*’ answer to the question for rational decisions or rationally acceptable compromises in C-situations is

- first: make a rational guess of what the others will do, and
- then: behave individually rational under the assumption that this guess is correct.

This kind of strategic rationality is called ‘*conditional rationality*’ and the minimax principle or the concept of equilibrium are prominent examples. One of the disadvantages is that sometimes it leads to solutions which are far from looking ‘rational’ from a layman’s everyday perspective. A good example is the well known *prisoner’s dilemma situation* cited in [65a] in which all imaginable special forms of conditional rationality result in the same highly implausible solution.

Cooperative rationality presupposes that there are some means of communication between the agents which allow for agreements on who chooses which option. Because this is the case in matrix games, these can be used as a prototype for further analyses of this kind of rationality and its variants.

Before, it is helpful to introduce ‘*coalitions*’. This term is used for any subset of the agent set N . It includes the strange special case of a coalition consisting of one agent only as well as the other extreme given by the agent set N itself which is — in this context — called the ‘grand’ coalition. A coalition simply is any $S \subseteq N$.

The concept of an option can be generalised to cover coalitions’ options too. For each coalition, each combination of one and only one option from each set of member options is an option. Thus, we introduce *option sets*

$$A_S = \bigotimes_{i \in S} A_i$$

Furthermore, it makes sense to speak of *coalition preferences*. A coalition is said to prefer an outcome a over b if at least one coalition member prefers a over b and no one b over a . Thus,

$$a P_S b \quad \text{iff} \exists i \in S : a P_i b \quad \text{and} \quad \forall j \in S : \neg b P_j a$$

3.4 α -, β -, and γ -rationality

Cooperative rationality takes into account that any coalition — including the degenerate case of one agent only — has control over some means to restrict the set of feasible outcomes. It is cooperatively rational to accept an outcome a if no coalition S can enforce by an appropriate restriction that the finally resulting x is preferred over a by this coalition. Formally:

$$\text{rational}(a) \text{ iff } \text{not}(\exists S : \exists B \subset A : (S \text{ can restrict to } B) \wedge \forall x \in B : x P_S a)$$

If it is cooperatively rational to accept a then a itself is called a rational outcome; the set C of all cooperatively rational outcomes is the *core* of the game. The different modes of restricting the set of feasible outcomes lead to different variants of cooperative rationality and to different coretypes.

The simplest possibility for a coalition S is to choose an available option from A_S . This restricts the set of still possible outcomes to those containing exactly this option. Cooperative rationality based on this mechanism is called α -*rationality*. The corresponding α -core is what originally has been coined the ‘core’ and was introduced already in 1953 by [3].

A bit more complicated is a restriction which results from the choice of a function from the set of those option-combinations available to the non-members of a coalition into the coalition’s own option-set. This idea gives the β -*rational* outcomes as elements of the β -core.

A third possibility is to choose one option and — by announcing this choice — to restrict the set of feasible outcomes to the ‘best answers’ of the others. The best answers are simply the conditionally rational options of the countercoalition given that S chooses what it has announced to choose. This leads into the γ -core which is the set of γ -*rational* outcomes.

What makes all these considerations attractive for creating a cognitive model is that cooperative rationality can be achieved by an idealised negotiation process. Assume that an available outcome is proposed by some agent. If another agent or a coalition of several agents can reject this proposal by a ‘*valid argument*’ against it, the proposal is discarded and substituted by another one. This process comes to an end if a proposal is made against which no valid argument can be raised. But what is a valid argument?

We can regard any argument as valid in which some coalition shows that its members have an option which guarantees an outcome better for at least one of them and for the others at least equivalent to the current proposal. Accepting these arguments as valid leads to α -rationality. Second possibility: If a proposal is made, a coalition argues that the others have to choose certain options in order to reach this outcome and that the arguing coalition can react to these choices in such a way that an outcome better for at least one of them and at least indifferent for the other members results. Taking these arguments as valid leads to β -rationality. Third variant: A proposal is made, and some coalition announces a certain option-combination their members will choose. They point out that if the others react in their own interests, the result is guaranteed to be better for at least one of them and at least indifferent for the others. This is a „Zugzwang” strategy. Accepting such arguments leads to γ -rationality.

3.5 Purposes of the model

The matrix game is the playground for the development of a first small and simple cognitive model. Later on, it can and should be extended such that it handles any type of C-situations. In the context of matrix games, the system serves three purposes:

- analysis of given matrix configurations,
- simulation of experimental bargaining processes,
- decision aid for human players during the bargaining process.

β - and γ -core of a C-situation are always subsets of the α -core. If the α -core is empty, each form of cooperative rationality fails. If not, the relations between the differences of β - and γ -core and their intersection induce different strategic possibilities in negotiating an acceptable accord. Ostmann [84] developed a system of game types based on relations between the different cores. He could show that the agreements of subjects at the end of experimental bargaining processes depend on these types. The cognitive model of cooperative rationality should allow to analyse a given matrix configuration such that it finds the rationality properties of each possible outcome from which the relevant properties of the configuration as a whole easily can be deduced.

Former experimental research [13] has shown that agreements at the end of experimental negotiations are very often elements of the minimal core which is the intersection of all nonempty cores. Nevertheless, proposals, rejections, and counterproposals during bargaining do not reflect the idealised bargaining process leading into this set. The model should be able to handle alternative hypothetical rules describing bargaining behaviour and modelling experimental data on negotiations. It should help to answer the question whether and under which restrictions idealised and data-modelling rules lead to the same accord.

Whether game theoretic concepts are suitable for negotiation support is an often treated and mostly negatively answered question [7]. But all these treatments restrict the analysed concepts to the conditionally rational ones. On the other hand, there are some approaches in which production systems have been used for prenegotiation analyses of a given situation as well as for advice on how to react on the others' reactions during bargaining [4]. Up til now, it is far from being clear whether the model we are going to develop is of any use as a negotiation support tool, but it opens the way to new experimental research which can shed some light upon this question.

3.6 Model lay-out

The first approach towards a model of cooperative rationality in matrix games uses the framework of ACT-R, version 3.0 [2, 5]. ACT-R is a production system which deviates from 'neo-classical' production systems [6] insofar as it uses procedural as well as declarative formats for the representation of long-term knowledge. Additionally, ACT-R attaches to the representations of knowledge content a system of parameters representing such aspects as knowledge accessibility, usefulness, time demand, and so on. These parameters are relevant for a special kind of mechanism selecting an instantiation to fire called 'rational analysis' and for models which simulate the time behaviour of human cognitive processes. There are learning mechanisms for several of these parameters as well as for the 'compilation' of declarative representations into procedural ones. At any step of a model process, there is one current goal on top of a goal stack, and only those rules can become active which contain on the first position of their condition part a pattern matching this goal.

The model of cooperative rationality uses declarative representations for

- the information describing a given matrix-configuration with its values,
- the goals and subgoals pursued while working on a given task, and
- the new knowledge about options and outcomes inferred during such a process.

3.7 Description of a matrix configuration

The declarative representations used by ACT-R are typed chunks called 'working memory elements' or — abbreviated — 'WMEs' (pronounced wymeas). Each WME is a frame-like structure. It holds a name and a sequence of slots and fillers. The first slot is *isa* and its filler is the name of the WME-type to which this WME belongs. The corresponding type definition specifies the names of the other slots. If one of these slots is not mentioned in the description of the WME, it automatically gets a default filler — if such is specified by the type definition — or — if not — the filler *nil*. The filler of slot *isa* is always the name of a WME-type; the other fillers usually are WMEs themselves.

The description of a matrix configuration contains one WME of type `game-configuration` which specifies no slots at all. Under former versions of ACT-R we would probably have endowed this type with a slot containing the list of partners, another one holding the list of matrixcells, maybe a third one with the list of options, and so on. Since matching of list elements in the condition parts of productions is no longer possible in version 3 of ACT-R, this is of no use any more.

Next, there are seven WMEs of type `coalition` representing the three ‘players’ — that is the more technical term for the partners, subjects, or agents —, the three possible pair-coalitions and the grand coalition. Each of them has a slot pointing to the `game-configuration`. For each coalition representing a single player, there are three `single-options` pointing to the corresponding player. Twenty seven `pair-options` — nine for each of the three pair-coalitions — have two slots pointing to the two partners of the respective pair-coalition. The types `single-options` and `pair-options` are subtypes of a supertype `options`.

Each of the matrix cells is represented by a WME of type `cell` pointing to the three `single-options` and to the three `pair-options` describing the cell’s position and to three so-called values. One could argue that the specification of the `single-options` is sufficient and that the `pair-options` as their pairwise combinations can be directly inferred. That is true, but the model is intended to represent natural cognitive processes in human subjects, and there is no reason to assume, that e.g. the belongingness of a cell to a whole matrix is easier and faster to be recognised than its membership in a three-cell-block. Knowledge of both facts results from direct perception, and there is no sound reason to assume that one is directly known while the other would be the result of some implausible inference process.

A value is not the number given in the printed form of a matrix game. Instead, value is a type of WMEs with a slot for that number — called `amount` —, a slot pointing to the player who gets this amount, and a third one pointing to the cell. Most of the WME-types which represent a matrix configuration have some other slots relevant for certain tasks but not used for the initial description of a matrix configuration. Such an additional slot is `goal-done` which exists for coalitions as well as for options and for cells, another example are several `cell-slots` used to point to WMEs describing valid objections against an outcome given by this cell. All slots of this kind start with the default filler `empty`.

There are two tools which help to establish the WMEs describing a matrix game. The first one consists of LISP-functions which generate matrix games. There are two options:

- building a matrix configuration by entering the different values, and
- creating a random-configuration.

The output of these functions is a file containing the matrix game as a LISP-structure. The second tool is a function named ‘`WME-machine`’ which takes this structure and creates a new file containing a call to `SetWM`, an ACT-R function which establishes the WMEs describing a given matrix game in working memory.

In ACT-R, the time needed to find a match between a pattern given by a production and a WME depends on several factors, one of which is the WME’s activation. Activation can be raised by a match between a pattern and another WME with the WME in question as one of its slot fillers. But sometimes, it is desirable to activate WME *A* if another WME *B* has been matched which does not point to *A*. The reason simply is that there is a high probability that *A* is needed in a situation in which *B* already has been proven to be useful. This conditional probability of usefulness is handled in ACT-R by so-called ‘inter-associativity values’ which are also used in the descriptions of matrix game configurations.

3.8 Goals and productions

The tasks which the system has to work on are described by goals. There are two different WME-types for the description of a goal: `search-goal` and `test-goal`.

- A *search-goal* asks the agent to search for something (as an option which can be used for an α -objection against a proposal, or an outcome better than another one for everybody), and
- a *test-goal* asks to test whether an object (like a specific option or a cell of the matrix) is such a ‘something’.

Each WME describing a goal has a slot naming the

- kind of the goal, a second one describing the
- *current-player*, a third for the
- *current-option*, and a fourth pointing to the
- *current-cell*.

Current-player, *current-option* and *current-cell* have the default-value *empty*. Whether this makes sense or not depends on the kind of the goal.

An example will be used to explain the system of goals and subgoals and the productions

- establishing subgoals,
- performing tests, or
- giving up further trials to reach a subgoal.

The example task is to find an option of a single agent or a combination of options from two or more agents forming a coalition which can be used to formulate a valid α -objection against a given proposal. This task is described by a *search-goal* of the kind *startgoal*. There are four productions which can become instantiated if this goal is the current one:

- One which establishes one of the three agents as *current-player* and a *search-goal* to test whether this player can raise an α -objection,
- another which takes a pair of agents as *current-player* and establishes a *search-goal* to test whether there exists an α -objection useful for this pair-coalition,
- a third taking the grand coalition N as *current-player* and setting a *search-goal* to test whether all the agents together can raise an α -objection,
- a fourth which administrates the negative result if all the other goals have been tried out and found nothing.

If the production mentioned first has fired, the current goal is a *search-goal* of the kind ‘*individual-alpha-objection*’ specifying as the *current-cell* the outcome which has been proposed, and as *current-player* the agent which has been chosen to be tested first. The instantiation of a *current-player* is a step which leads from the part of the model which represents the group of agents as a whole to the part which models a single agent. The example task is part of the formal analysis of a given matrix configuration in which this difference is practically irrelevant. But modeling empirical data from negotiations requires to discriminate between general knowledge and private knowledge of the agents which depends on former inferences an agent has drawn or not. The system can handle this by restricting access to different parts of the totally available knowledge dependent on the current filler of the slot *current-player*.

Given the new current goal, two productions have a chance to fire:

- One establishing a *test-goal* of the kind ‘*individual-alpha-objection*’ for one of the options available to the *current-player*,

- and a second one stopping the test and creating a new WME which states that `current-player` is unable to raise an α -objection.

Such kind of a search-and-test-procedure needs a mechanism which avoids that an option (or a cell, or whatever is being tested) is tested twice for the same purpose. This is done by setting the `goal-done` value of an option equal to the name of a `search-goal` as soon as this goal has created a `test-goal` or a `sub-search-goal` establishing this option as ‘current’. Double testing is made impossible by inserting an appropriate test in the condition part of all productions which establish or re-instantiate a `test-goal` or a `sub-search-goal`.

Productions creating a `test-goal` or a `sub-search-goal` have conditions which imply the conditions of the corresponding GIVE-UP-production. ACT-R offers the possibility to prevent a premature stop of tests or subgoaling by giving them a higher production strength. A learning mechanism available in ACT-R is used to let the strength of the GIVE-UP-productions grow over time which leads to an increasing probability of a stop before all possibilities have been tested. The danger of making an error by not finding e.g. a possible α -objection can be minimised if good candidates for this objection are tested first.

Good candidates for some type of valid arguments are very often good candidates for other types of objections too. Therefore, we can use an ACT-R mechanism called ‘base level learning’ to increase the probability of probably useful alternatives to be tested first. The tuning of the combination of

- base-level learning (for the objects to be tested) and
- production-strength learning (for the GIVE-UP-production),

such that a combination of test-time and probability of missing a possible objection is minimised, has not yet been implemented. Doing this will be a contribution to answering the more general question: How can a system learn to adapt to time-restrictions such that the probability of an error of omission is minimised under the condition of a given distribution of the time used. Besides, such a mechanism models the fact that human subjects in a matrix game become familiar with the configuration of numbers and that this familiarity helps them to look into the right direction when searching for an objection or an acceptable compromise instead of starting a random search.

The current status of the project is that the most important aspects of cooperative rationality needed to analyse a given matrix configuration have been implemented. Tuning the interplay between base-level learning and production-strength learning is the next step, and maybe at the next European Workshop on Cognitive Modelling we can show that models of social systems can contribute to answering the question on how cognitive systems in general can adapt to time restrictions.

3.9 References

- [1] J. R. Anderson. *The adaptive character of thought*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.
- [2] J. R. Anderson. *Rules of the mind*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.
- [3] D. B. Gillies. *Some theorems on n-person games*. Ph.D. Dissertation, Department of Mathematics. Princeton University, Princeton, 1953.
- [4] G. E. Kersten. *Negotiation support: Development of representations and reasoning*. volume 34 in *Theory and Decision*. 1993, pages 293—311.
- [5] C. Lebière. *ACT-R 3.0: A users Manual*. Carnegie-Mellon-University, Pittsburgh, March 1996.
- [6] D. B. Lenat & G. Harris. *Designing a rule system that searches for scientific discoveries*. In D. A. Waterman & F. Hayes-Roth (eds.), *Pattern directed inference systems*. Wiley, New York, 1978, pages 25—51.

- [7] B. R. Munier & J.L. Rullière. *Are game theoretic concepts suitable negotiation support tools?* From Nash equilibrium refinements toward a cognitive concept of rationality. In volume 34 of *Theory and Decision*. 1993, pages 235—253
- [8] A. Ostmann. *Die Berücksichtigung externer Effekte und der Endlichkeit des Ergebnisraums bei kooperativ gespielten Normalformspielen*. volume 88 in *Arbeiten der Fachrichtung Psychologie*. Universität des Saarlandes, Saarbrücken, 1984.
- [9] A. Rapoport & A. Chammah. *Prisoner's dilemma*. University of Michigan Press, 1965.
- [10] A. Rapoport. *N-person game theory: Concepts and applications*. University of Michigan Press, 1970.
- [11] W. H. Tack. *Cooperative conflict resolution*. In E. D. Lantermann & H. Feger (eds.), *Similarity and choice: Papers in honour of Clyde Coombs*. Huber, Bern, 1980, pages 290—326.
- [12] W. H. Tack. *New approaches in research on cooperative conflict resolution by group decisions*. volume 68 in *Acta Psychologica* 1988, pages 113—136.
- [13] W. H. Tack. *Rationales Handeln in sozialen Situationen*. volume 22 in *Zeitschrift für Sozialpsychologie*. 1991, pages 151—165.
- [14] W. H. Tack. *Rationale Lösung von Interessenkonflikten*. volume 11 in *Zeitschrift für Psychologie*. 1991, pages 352—364.
- [15] B. Walliser. *Instrumental rationality and cognitive rationality*. volume 27 in *Theory and Decision*. 1989, pages 7—36.

Agentenmodelle zum Umgang mit vollständig erneuerbaren Gemeingütern

Martin Beckenkamp¹ und Axel Ostmann²

ABSTRACT

Zur experimentellen Untersuchung des konkurrierenden Zugriffs auf einen Ressourcenpool, der sich periodisch erneuert, stellt das Ostrom-Gardner-Walker Szenario (vgl. Ostrom, Walker & Gardner, 1994) ein flexibles Werkzeug zur Verfügung. Es erlaubt, die spieltheoretischen Analysen der Übernutzungs- und Krisenproblematik von Gemeingütern mit dem Handeln der Versuchspersonen zu konfrontieren. Bei der Interpretation der beobachteten Aktionen ergeben sich jedoch Probleme, Evaluationsprozesse und Handlungsintentionen, die zu diesen Handlungen geführt haben, ausreichend zu identifizieren. Gut identifizierbar sind die Verlaufsmuster ("Makro"-Effekte der Gruppe). Hinweise auf den Einfluß der Erwartungen und Intentionen, die zu den individuellen Aktionen und den Makro-Effekten führen, ergeben sich aus Begleitfragebögen.

In Regressionsanalysen kann der Einfluß des bisherigen Ressourcenmanagements der Gruppe und der Einfluß zukünftiger Erwartungen bezüglich des Gruppenverhaltens als Prädiktor für die jeweilige Aktion des Individuums untersucht werden. Unterschiede in den Bemühungen, sich Vorteile gegenüber anderen zu verschaffen, können sowohl aus dem bisherigen Verlauf, wie etwa angestautem Ärger, Gier oder Angst, wie auch aus Unterschieden in den Individuen selbst, wie etwa Wettbewerbs- oder Kooperationsorientierung begründet werden.

Aufbauend auf diesen abgeleiteten Konstrukten und den Regressionsanalysen werden Agentenmodelle spezifiziert, die die beobachteten individuellen und Gruppenverhaltensmuster generieren.

4.1 Einleitung

Güter sind Mittel, die zur Befriedigung menschlicher Bedürfnisse dienen. Gemeingüter sind — im Unterschied zu Wirtschaftsgütern (Sachen, Rechte und Dienstleistungen), die man sich an einem mehr oder weniger idealisierten Markt gehandelt vorstellt — nicht einfach privatisierbar. Der mögliche Zugriff dritter Personen kann nicht oder nur unter nichttrivialen Umständen verhindert werden. Gemeingüter können aus diesem Grund zu spezifischen Problemen führen, die in dieser Art für Wirtschaftsgüter nicht bekannt sind und von Hardin (1968) als die "Tragödie der Gemeingüter" bezeichnet wurde.

Hardin führt zur Illustration der Tragödie eine Gras- oder Steppenlandschaft bzw. eine "Weide" an, die jedem Hirten zur Nutzung offensteht (in Mitteleuropa entspricht dies in etwa einem Allmend). An diesem Beispiel erläutert er die Nutzenstruktur bzgl. der einzelnen Hirten (Individuen) einerseits und der Gemeinschaft andererseits, die zwangsläufig zur Tragödie führt. Jahrhundertlang mag die gemeinsame Nutzung gut funktionieren, doch der Tag der Abrechnung kommt nach Hardin, wenn etwa durch soziale Stabilität bedingt, die Bevölkerung zunimmt und somit auf einmal mehr Nutzer des Gemeingutes vorhanden sind. Dann resultiert aus der inhärenten Logik, mit der Gemeingüter funktionieren erbarmungslos die Tragödie:

Die Tragödie beruht - und das ist das Ärgertliche am Commons-Dilemma - auf der Rationalität

¹AE Kognitive Modellierung, FB Psychologie, Universität des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, beckenkamp@cops.uni-sb.de

²FB Finanzwissenschaften, Universität des Saarlandes, Im Stadtwald, D-66123 Saarbrücken & FB Volkswirtschaft, Universität Karlsruhe, axel@cops.uni-sb.de

der Akteure. Als rationales Wesen versucht jeder Hirt seinen Gewinn zu maximieren. Explizit oder implizit fragt jeder "Was bringt es mir, wenn ich ein Vieh mehr auf die gemeinsam genutzte Weide stelle?" Dieser Nutzen für den Einzelnen hat eine positive und eine negative Komponente. Hardin meint dazu:

1. Die positive Komponente besteht in der Hinzunahme eines weiteren Tiers. Da der Hirte fast alle Einnahmen aus diesem weiteren Tier als Gewinn verstreichen kann, ist der positive Nutzen fast +1.
2. Die negative Komponente ist das Übergrasen des Gemeingutes durch ein Tier. Diese Effekte werden aber verteilt auf alle Hirten zusammen, sodaß der individuell zu tragende Schaden nur ein Bruchteil von 1 ist.

Daraus schließt der entsprechende Hirte, daß er durchaus noch ein Tier dazustellen sollte... und noch eins... und noch eins ... Aber diese Schlußfolgerung wird von jedem rational denkenden Hirten getroffen und darin liegt die Tragödie.

4.2 Eine formale Definition von Umweltgemeingütern und "Rationalität"

Im Anschluß an Hardin's Arbeit wurde dieser Zusammenhang formal präziser mit spieltheoretischen Begriffen gekennzeichnet. Folgt man etwa dem Ansatz von Dasgupta und Heal (1979), kann jeder Nutzer eines Gemeingutes im Ausmaß x auf die gemeinsam genutzte Ressource zugreifen. Der Gewinn aus dem Zugriff im Ausmaß x auf die gemeinsam genutzte Ressource ist von zwei Parametern abhängig:

1. dem Gesamtzugriff s aller Benutzer auf die Ressource. Die Produktionsfunktion $f(s)$ gibt an, welcher Gesamtertrag $u(s)$ in Abhängigkeit vom Gesamtzugriff s aus der Ressource geschöpft werden kann. Der Gesamtbeitrag s setzt sich zusammen aus der individuellen Nutzung x und dem Zugriff der anderen Nutzer y . Damit ist $s = x + y$
2. der Quote q . Der individuelle Ertrag $u(x, y)$ ergibt sich dann aus einer Quote $q = \frac{x}{s}$, die mit dem Gesamtertrag aus dem Gemeingut $f(s)$ multipliziert wird $u(x) = qf(s)$

Rational ist es nun, auf einen vorgegebenen Gesamtzugriff der anderen (y) eine "beste Antwort" zu suchen, so daß $u(x + y, x)$ nach x maximiert wird. Dabei wird also ausschließlich nach dem Eigennutz des jeweiligen Agenten maximiert. Unterstellt man jedem der Nutzer des Gemeingutes diese Rationalität, läßt sich per Kalkül feststellen, ob es möglicherweise einen Gleichgewichtszustand oder Gleichgewichtszustände gibt, bei denen für keinen der Nutzer ein x' existiert, daß die Bedingung erfüllen würde $u(y + x, x) < u(y + x', x')$. In diesem Gleichgewichtszustand, dem "Nash-Gleichgewicht", ist die Auszahlung für jedes einzelne Individuum unter Zugrundelegung einer individuellen Nutzenmaximierung optimiert. Hardins oben dargestellte Illustration bzgl. der Hirten stellt dar, was diese formale Überlegung in Bezug auf die einzelnen Hirten bedeutet. Nun kann man dieses Nash-Gleichgewicht, das sich formal aus einer Strategie der individuellen Nutzenmaximierung ergibt — also mit der Vorannahme oder Unterstellung von vollständig rationalen Agenten — mit dem Wohlfahrtsoptimum vergleichen. Das Wohlfahrtsoptimum ist gegeben, wenn $u(s)$ maximiert ist. Es ist also durch den Punkt gekennzeichnet, an dem die gemeinsam genutzte Ressource optimal genutzt wird, wenn sowohl der Ertrag aus der Ressource, als auch die Kosten, die für die Nutzung der Ressource anfallen, im optimalen Verhältnis stehen: die Grenzkosten entsprechen hier dem Grenzertrag.

Das Wohlfahrtsoptimum und das Nashgleichgewicht können als Meßlatte dienen für das konkrete Verhalten von Personen: Würden sich Personen vollständig rational im Sinne einer individuellen Nutzenmaximierung verhalten, müßten Sie im konkreten Umgang mit Umweltgemeingütern beim Nash-Gleichgewicht liegen. Verhalten sie sich hingegen vollständig kooperativ, müßte das Wohlfahrtsoptimum realisiert werden. Hardins Ausführungen bzgl. der Tragödie der Gemeingüter können

so interpretiert werden, daß er meint, daß Personen letztlich immer beim Nash-Gleichgewicht landen. Tatsächlich verhalten sich Personen häufig weder vollständig rational noch vollständig kooperativ im oben dargestellten Sinn. Dies hat sich in unterschiedlichsten Szenarien gezeigt, häufig wurden dazu Versuchspersonen untersucht, die eine computersimulierte Ressource zu verwalten hatten. Die Auszahlungen an die Vpn richtete sich nach den individuell erwirtschafteten Ergebnissen im Umgang mit der Simulation.

4.3 Das Saarbrücker Szenario

Auch im Saarbrücker-Szenario, einer Computersimulation die eng an finanz- und wirtschaftswissenschaftliche Fragestellungen angelehnt ist und sich an konkret bestehenden Modellen für Ertrags- und Kostenfunktionen von Umweltgemeingütern orientiert, zeigten sich entsprechende Ergebnisse. Bevor wir diese im Detail anschauen, soll zunächst ein Blick auf wesentliche Eigenschaften und die formale Definition des Saarbrücker Szenarios geworfen werden.

ALLGEMEINE EIGENSCHAFTEN DES SAARBRÜCKER SZENARIOS

1. Es handelt sich um eine Ressource, die in jeder Periode vollständig regeneriert (Das Szenario ist angelehnt an die Modellierung der Produktions- und Kostenfunktion der Wasserversorgung der Kanarischen Inseln).
2. Jeder Versuch findet über zwanzig simulierte Perioden statt (die Vpn wissen nicht, wieviele Perioden durchgeführt werden).
3. Jede Vp i hat in jeder Periode die Möglichkeit, zwischen $x_{min} = 0$ und $x_{max} = 25$ mit jedem beliebigen Betrag x_i auf das Gemeingut zuzugreifen. Für jede Einheit, mit der nicht auf das Gemeingut zugegriffen wird, erhält sie einen Betrag von 5 ecu, die nach dem Versuch in Pfennige umgerechnet werden, ausgezahlt wird. Der Betrag x_i mit dem sie auf das Gemeingut zugreift, erbringt einen Ertrag, der sich nach der oben dargestellten Produktionsfunktion errechnet: $u(x_i) = qf(s)$, wobei $q = \frac{x}{s}$ und $f(s) = s(23 - 0.25s)$.

Damit lehnt das Saarbrücker Szenario an Ostrom, Gardner und Walker (1994) an, die in umfangreichen experimentellen Untersuchungsreihen mit einigen Variationen wie etwa bzgl. der Versuchspersonenzahl oder der Kommunikationsmöglichkeit den Zugriff auf Umweltgemeingüter untersucht haben. Für diese Szenario und den entsprechenden Variationen sind Nash-Gleichgewichte und Wohlfahrtsoptima errechnet worden. Auch hier zeigte es sich, daß die Vpn im allgemeinen nicht so schlecht sind, wie es das Nash-Gleichgewicht befürchten lassen müßte, aber auch weit entfernt vom Wohlfahrtsoptimum bleiben. Im Saarbrücker-Szenario kommt ein vierter Punkt zum OGW-Szenario hinzu: die Einführung von Kontrollen und Sanktionen.

4. Für das Szenario ohne Kontrolle ergibt die Produktionsfunktion einen maximalen Ertrag bei $s = 46$. Das Wohlfahrtsoptimum liegt bei $s = 36$. Die Versuchspersonen werden darüber informiert, daß ein Kontrolleur eingeschaltet wird, wenn der Gesamtzugriff s den Betrag von 46 überschreitet. Der Kontrolleur ist ein Zufallsmechanismus, der einen der acht Vpn nach Zufall (Gleichverteilungsmodell mit Unabhängigkeit der einzelnen Kontrollen) kontrolliert. Wenn diese Versuchsperson i mit mehr als 5 Einheiten ($x_i > 5$) auf das Gemeingut zugegriffen hat, muß sie Transferleistungen zahlen an die anderen Versuchspersonen, die linear abhängig vom eigenen Überziehungsbetrag über die 5 erlaubten Einheiten hinaus sind. Darüber hinaus kommt sie für die Kosten des Kontrolleurs in Höhe von 40 ecu auf. Hat die kontrollierte Person hingegen nicht überzogen, dann werden die Kosten für den Kontrolleur auf alle 8 Vpn verteilt.

Das Saarbrücker Szenario beschäftigt sich also mit der Frage, wie die Institutionen "Kontrolle" und "Sanktionen" im Hinblick auf den (selbstverwalteten) Umgang mit Umweltgemeingütern wirken. Hierzu wurden ebenfalls umfangreiche Experimentalserien gemacht, wobei ebenfalls unter anderem

die Versuchspersonenanzahl und die Kommunikationsmöglichkeit variiert wurden. In den meisten dieser Serien wurden auch statt einer zwei Vpn per Zufall kontrolliert, wenn das Limit überschritten wurde, im Experiment, das im folgenden beschrieben wird, war jedoch nur eine Kontrolle/Periode bei Überschreitung des Limits von 46 vorgesehen. Aus den vorherigen Serien ergab sich die Fragestellung, inwieweit die Höhe der angedrohten und ggf. zu zahlenden Sanktionen eine Rolle spielt im Umgang mit dem Umweltgemeingut. Dazu führten wir ein Experiment mit drei unterschiedlichen Versuchsbedingungen durch: Sanktionsfaktor $\sigma = 0$, $\sigma = 1$ und $\sigma = 2$.

Unter der Versuchsbedingung $\sigma = 0$ hatte die Vp, falls sie kontrolliert wurde und das individuelle Limit von 5 Einheiten überschritten hatte, den Kontrolleur zu zahlen. Die Kontrolle kostete, unabhängig von der Höhe des Verstoßes, 40 ecu. Hatte die kontrollierte Vp nicht verstoßen, dann wurden die Kosten von 40 ecu auf die acht Vpn aufgeteilt. Unter der Versuchsbedingung $\sigma = 1$ hatte die Vp, falls sie kontrolliert wurde und das individuelle Limit von 5 Einheiten überschritten hatte, den Kontrolleur und Ausgleichszahlungen in Höhe des Überziehungsbetrages $x_i - 5$ jeweils an die sieben anderen Vpn zu zahlen. Unter der Versuchsbedingung $\sigma = 2$ war im entsprechenden Fall das Doppelte des Überziehungsbetrages an Ausgleichszahlungen fällig. Wie unter der Versuchsbedingung $\sigma = 0$ wurden auch in diesen beiden Bedingungen für den Fall, daß die kontrollierte Vp nicht verstoßen hatte, die Kosten von 40 ecu auf die acht Vpn aufgeteilt.

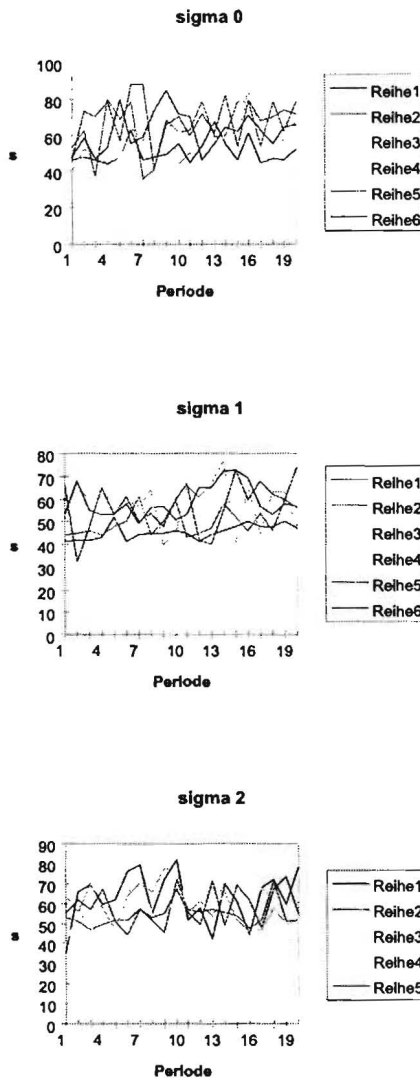
FORMALE EIGENSCHAFTEN DES SZENARIOS

Um die Leistungen der Vpn unter den verschiedenen Versuchsbedingungen beurteilen zu können, bietet sich neben dem direkten Vergleich untereinander die Berechnung der entsprechenden Nash-Gleichgewichte und des Wohlfahrtsoptimums an. Das Wohlfahrtsoptimum verändert sich in den drei Versuchsbedingungen nicht (da es unterhalb der Grenze liegt, wo die Kontrolle eingeschaltet wird). Es liegt in allen Versuchsbedingungen sowie beim Szenario ohne Kontrolle und Sanktionen bei $s = 36$. Für das Szenario ohne Kontrolle und Sanktionen in Anlehnung an Ostrom, Gardner und Walker (1994) errechnet sich das (symmetrische) Nash-Gleichgewicht bei $s = 64$. Die Einführung von Kontrollen und Sanktionen dient nun ja dazu, dieses Nash-Gleichgewicht näher zum Wohlfahrtsoptimum zu drücken. Das tut es auch, allerdings ist die Angabe nun etwas komplizierter, da durch die Zufallskontrollen ein Erwartungswertkalkül zur Berechnung der Nash-Gleichgewichte erforderlich ist. Dabei können für jede der Versuchsbedingungen mehrere (asymmetrische) Nash-Gleichgewichte errechnet werden:

Lösung	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	Szenario ohne Kontrolle
Wohlfahrtsoptimum	36	36	36	36
Nash-Gleichgewichte	60.25 61.6 62.5	57.63 58.8	55.67 55	53.33 64
Effizienzen in Promille	423 bis 335	578 bis 475	645 bis 598	395

Somit wirkt die Einführung von Kontrollen und Sanktionen in die gewünschte Richtung. Die Zugriffe auf das Umweltgemeingut werden reduziert, wenn man vollständig rationale Agenten voraussetzt, und zwar je größer die angedrohten Sanktionen für entsprechende Überziehungen sind, umso weniger rentiert es sich noch, übermäßig stark auf die Ressource zuzugreifen.

EMPIRISCHE ERGEBNISSE MIT DEM SAARBRÜCKER SZENARIO



Wir unterscheiden in unserer Datenanalyse, ähnlich wie etwa auch Flache (1996), die Analysen auf der Makroebene von den Ergebnissen auf der Mikroebene. Ergebnisse auf der Makroebene betreffen die gesamte Gruppe, Ergebnisse auf der Mikroebene hingegen das Verhalten der einzelnen Versuchspersonen in den entsprechenden Gruppen unter den einzelnen Versuchsbedingungen $\sigma = 0, 1$ und 2 . Ein wichtiger Indikator für die verschiedenen Versuchsbedingungen ist dabei der Gesamtzugriff s und seine Veränderung über die 20 Perioden des Versuchs. Dabei ist zunächst — über alle Gruppen betrachtet — ein leichter Zuwachs beobachtbar, der sich in der Korrelation zwischen Periode und Gesamtzugriff s ausdrückt ($r = 0.17$). In den Korrelationen für die einzelnen Versuchsbedingungen drückt sich eine gewünschte Effizienz der Sanktionen aus: die Zunahme an Zugriffen über die Runden wird mit wachsendem σ verringert ($r = 0.23, 0.20$ und 0.07). Die Oszillationsmuster selbst scheinen sich nicht über die Zeit wesentlich einzupendeln bzw. zu beruhigen. In der folgenden Tabelle sind die Perioden 5 – 12 als Teil 1 und 13 – 20 als Teil 2 des Versuchs bezeichnet und die jeweiligen Standardabweichungen unter den unterschiedlichen σ -Bedingungen errechnet. Unter $\sigma = 0$ nimmt die Schwankungsbreite (erwartungstreu geschätzte Standardabweichung) über alle Eingaben aus der den jeweiligen Versuchsteilen gesehen ab, bei $\sigma = 2$ bleibt sie etwa gleich und bei $\sigma = 1$ nimmt sie zu.

σ	Teil 1	Teil 2
0	12.92	9.69
1	7.61	9.64
2	10.01	9.01

Besonders interessant auf der Makroebene sind die Effekte der Sanktionen auf das Zugriffsverhalten. Die empirisch gefundenen Zugriffe und Effizienzen können mit den theoretisch errechneten Zugriffen und Effizienzen des Nash-Gleichgewichts und mit dem Wohlfahrtsoptimum verglichen werden:

	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$
s	60.79	53.49	59.33
Effizienzen in Promille	303	612	394

Das heißt, das unter den Bedingungen $\sigma = 0$ und $\sigma = 2$ sich die Vpn noch ineffizienter verhalten, als es das Nash-Gleichgewicht angibt, unter der Bedingung $\sigma = 1$ hingegen effizienter. Bzgl. des Gesamtzugriffs der Gruppe s zeigt sich ein deutlicher V-Effekt.

Zu den Daten und insbesondere zu den Oszillationsmustern, die wir in früheren Experimenten erhalten haben, hatten wir einfache Computersimulationen entworfen, die das Gruppenverhalten recht gut erklären konnten. Der hier vorgefundene V-Effekt war jedoch damit überhaupt nicht erklärbar, zumal in der damaligen Simulation eine Reaktion auf unterschiedliche Sanktionsbedingungen überhaupt nicht vorgesehen war. Wir haben zu diesem Befund intensive Berechnungen und Datenanalysen bzgl. der Individuen angefertigt und wollten eine möglichst sparsame Computersimulation erzeugen, die ähnliche Befunde hervorbringt, wie wir sie eben geschildert haben.

4.4 Agentenmodell

In der Social-Dilemma-Literatur wird häufig untersucht, inwieweit man den Versuchspersonen und damit dann generell den natürlichen Akteuren in Social- respektive Commons-Dilemma-Situationen Gier und/oder Furcht unterstellen kann. Beide Konstrukte eignen sich dazu, individuelle Unterschiede (egal ob als "state" oder "trait") zu erklären. Gier kann dabei verstanden werden als das Motiv der Vorteilsnahme, Furcht als Motiv des Selbstschutzes. Dabei wird in der Literatur weitgehend offen gelassen, inwieweit affektive und inwieweit kognitiv-rationale Prozesse unterstellt werden.

Für unser Agentenmodell haben wir affektive und kognitiv-evaluative Prozesse versucht zu trennen. Der Agent antwortet auf externe Reize (y, s , Kontrollergebnisse) entsprechend seines internen Zustands mit einem bestimmten Zugriff x .

Experimentelle Daten legen nahe, daß die Entscheidung über die Zugriffshöhe sequentieller Natur ist: zunächst erfolgt die Entscheidung, ob das Limit eingehalten wird, und danach jene über die Höhe. Im Experiment beobachtet wurden nur gewisse Zugriffswerte. Diese empirische Skala umfaßt "prominente" Werte. Für den Fall der Regeltreue ist x fast immer 5 oder 0. Dabei stellt 5, also das Ausschöpfen des Limits, den Normalfall dar. Im Falle des Verstoßes wurden starke Häufungen bei den 5er Schritten und bei den kleineren Verstößen beobachtet. Für unser Modell stilisieren wir die Skala $scale(.)$ für mögliche Zugriffe wie folgt:

$$scale(d) = 0, 5, 6, 8, 9, 10, 15, 20 \text{ für } d = 0, 1, \dots, 5 \quad \text{und} \quad scale(d) = 25 \text{ für } d > 5$$

Wir gehen davon aus, daß es für die Entscheidung zu verstoßen, zwei unterschiedliche Gründe geben kann: es mag sich einmal dabei um einen Affekt handeln, ein andermal um kühle Überlegung.

Um zwischen Affekthandeln und Überlegungshandeln zu trennen, unterstellen wir eine internen Speicher für so etwas wie Gier oder Aggression, der entsprechend der Erstevaluation der Lage Ladung zugeführt bekommen kann. Ist danach der "Speicher voll", d.h. überschreitet die Ladung eine gewisse (eventuell individuelle) Hemmschwelle, so kommt es zu einem im wesentlichen affektiven Verarbeitungsprozeß. Im anderen Fall erfolgt eine Zweitevaluation; der Agent ist frei, die Lage mittels einer casuistische Heuristik zu beurteilen. In beiden Fällen erfolgt die Ausgabe des jeweils bestimmten Zugriffs. Bei der affektiven Verarbeitung erfolgt zusätzlich eine entsprechende Entladung des Speichers. Die aggressive Handlung ist mit interner Katharsis verbunden.

Die folgende Tabelle enthält eine Prozessübersicht und eine erste Spezifikation der Prozeduren.

Prozeß	Modul	Bezüge	Ausgaben	Nächster Prozeß
Affekt	“anstauen”	Auszahlung Kontrollergebnis	$A \leftarrow A + E$	Entscheidung
Entscheidung	“heiß” “kühl”	Krise	$x \leftarrow 0$	Katharsis
		Normalzustand	$x \leftarrow 5$	Warten
		Chance	$x \leftarrow 9$	Warten
				Warten
Katharsis	“abkühlen”	$K = A$ (bzw. $K = 2E$)	$A \leftarrow A - K$	
Warten	–	neue Runde		Affekt

Wir benennen die internen Variablen wie folgt:

- H Hemmschwelle
- A Aggression (Speicherinhalt)
- x Ausgabevariable
- E Aggressionszuwachs (Zuladung)
- K Katharsis (Entladung)

Die obige Systematik erlaubt, Fragen zu möglichen Präzisierungen zu stellen:

- inwieweit kann die Hemmschwelle als konstant betrachtet werden
- welche Faktoren führen zu einem Aggressionszuwachs
- in welchem Ausmaß kann Entladung erfolgen
- wie wird unter den Fällen “normal”, “Krise” und “Chance” unterschieden
- wieweit wird das Kontrollregime s berücksichtigt

Diese Fragen stellen sich nach zwei Seiten: empirische Indizien und Modelleigenschaften. Schließlich ist dann auch die Frage zu stellen, wieweit aggregierte und individuelle Verlaufsdaten mit den empirischen Befunden in Einklang gebracht werden können, und welche Variation durch die internen Parameter ausgelöst werden kann.

4.5 Indizien

An externen Variablen werden gemäß der Begleitbögen vor allem die x, s, u und Kontrollergebnisse der letzten Periode fokussiert. Das Kontrollregime wird nicht benannt. Sein Einfluß kann also als im wesentlichen indirekt, etwa über die Auszahlung, vorausgesetzt werden. Benannte Quellen des Ärgers sind zu geringe Auszahlung, zu hohe Zugriffe der Partner, bestrafte Zurückhaltung und ein entdeckter eigener Verstoß. Wir benutzen entsprechende Indikatoren zur additiven Bestimmung des Aggressionszuwachses. An zwei Stellen berücksichtigen wir das Kontrollregime direkt:

1. Der Aggressionszuwachs durch entdeckten eigenen Verstoß wird mit der Schärfe des Kontrollregimes gewichtet. Der entsprechende Ärger wird mit den Proportionalitätsfaktor gewichtet
2. Der affektiv-bestimmte Überziehungsbetrag wird um $1 - [\sigma = 0]$ gedämpft. Bei zugriffsproportionaler Strafe gilt eine gewisse Vorsicht.

Diese vorsichtige Berücksichtigung des Kontrollregimes steht im Einklang mit den Daten der Begleitbögen.

Die Begleitbögen geben wenig Aufschluß über mögliche "rationale" Heuristiken; deutlich wird jedoch trotz seltenem Auftreten, daß ein besonders hoher Gesamtzugriff in der Vorrunde, zum "Ausstieg" aus dem Gemeingut, also zur Entscheidung $x = 0$ führen kann. Häufiger findet sich der entgegengesetzte Ausnahmefall einer deklarierten Chance ("jetzt zugreifen").

Manche Versuchspersonen deklarieren sich als bemüht, ihr eigenes Zugriffsverhalten zu dämpfen. Dabei sind die Motive unterschiedlich: man trifft auf Bindung an Normen, aber auch auf die Absicht, ein Vorbild zu geben. Es wird auch deutlich, daß in der Regel, die Dämpfungsabsicht nicht gewisse Verstöße ausschließt, etwa weil "es nun zuviel wird" oder man nicht immer "der Dumme sein" will.

Die genannten interindividuellen Unterschiede kondensieren wir in der Variation der Hemmschwelle H . Wir setzen die individuelle Hemmschwelle als $H = H_0 + H_i$. Für den Zustand $A > H$ läuft dann der affektive "heiße" Prozeß, ansonsten der "kühle" Prozeß ab. Man beachte, daß eine höhere Hemmung nicht unbedingt mit einer gedämpfteren Zugriffsverhalten einhergehen muß. Geringere Hemmung bedeutet frühere Entladung und damit im Schnitt ein geringeres Aggressionsniveau.

Im folgenden werden wir zwei unterschiedliche Zugriffsregel für den "heißen" Prozeß untersuchen.

Typ A. die Zugriffshöhe ergibt sich als $x \leftarrow [\sigma = 0] + A - H_0$. Diese Regel führt zu tendenziell aggressiverem Zugriffsverhalten der Gehemmtten.

Typ B. die Zugriffshöhe ergibt sich als $x \leftarrow [\sigma = 0] + A - H$. Diese Regel führt zu tendenziell gedämpfteren Zugriffsverhalten der Gehemmtten.

4.6 Weitere Modelleigenschaften

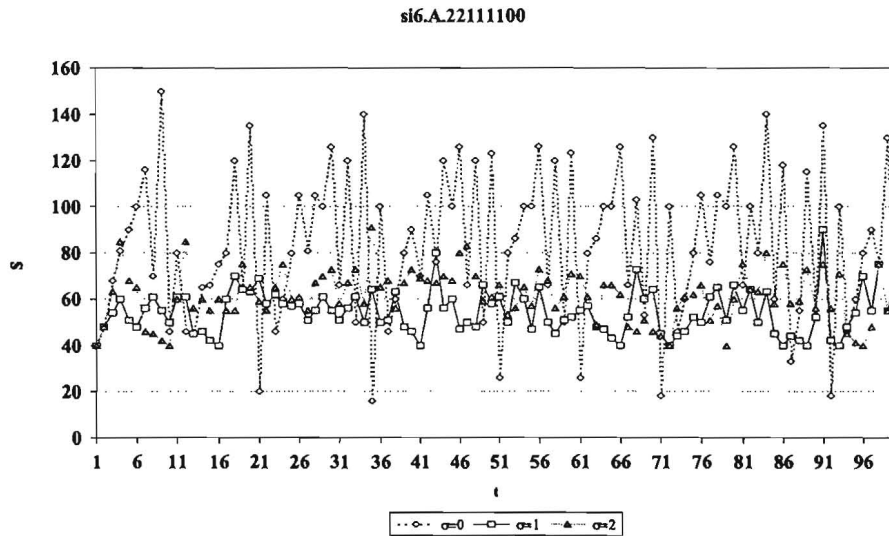
Die bislang diskutierten Spezifikationen sind weiterhin unvollständig. Sie wurden im Verlauf der Studie in gewissen Variationen als Programme implementiert (siehe Anhang 4.8). Zentral sind dabei die Unterprogramme "affekt" und "reak", wobei "affekt" den Prozeß "Affekt" und "reak" die Prozesse "Entscheidung" und "Katharsis" abbildet. Für das Verhalten des Systems ist wie bereits oben angedeutet auch der Vektor der individuellen Hemmschwellen zentral. Als Anfangswerte verlangt das System den x -Vektor, den A -Vektor, den Vektor $H = (H_0, H_1, \dots, H_n)$, sowie nicht zu vergessen die Initialisierung der Zufallszahlen, die für den Kontrollmechanismus benötigt werden. Im folgenden berichten wir nur über die Startwerte $x^0 = (5, \dots, 5)$ und $H_0^0 = 3$. Als Startwert für x sind nur Werte kleiner gleich x^0 kritisch, da hierfür kein Aggressionsaufbau mehr stattfindet. Der Einfluß der H_0 -Werte auf den mittleren Zugriff ist dagegen diskontinuierlich und schwerer vorherzusagen, da frühere Auslösung mit geringerem Zugriff einhergeht.

4.7 Simulationsläufe

Bei der gewählten Implementierung ist für die anfänglich benutzte defacto trimodalen Entscheidung "heiß", "kalt", "weder noch" (im Programm "reak" realisiert in der letzten Zeile, siehe Anhang) theoretisch schwer entscheidbar, ob die Oszillationen andauern. Durch Abänderung der letzten Zeile kann der "weder noch"-Fall mit seiner verstoßfreien Verhalten beseitigt werden. Letztere Versionen werden im folgenden durch das Superskript ² gekennzeichnet. Der "weder-noch" Bereich hat ähnlich wie die Vorbesetzungsabfrage $(s - x) > 100$ die Funktion, Extremfälle im Verhalten nur in seltenen Fällen durch "Sonderregeln" zu erzeugen. Sehen wir jedoch die "kalte" Reaktion nicht als Sonderfall, so dürfen wir sicherlichen den Bereich des "weder-nochs" einschränken, wenn nicht sogar verschwinden lassen. Wir wenden uns zunächst dem trimodalen Fall zu, um die Auswirkung der verschiedenen Eingangswerte zu beurteilen. Die dazu herangezogenen Simulationsläufe haben die Kennung si6.A.xxxxxxx (Lauf 11 – 41).

Lauf 11 ist weitgehend exemplarisch und soll hier etwas genauer besprochen werden. Die folgende

Abbildung zeigt die Oszillationen des Gesamtzugriffs für die drei verschiedenen Kontrollregime.



Der Lauf ist 100 Perioden lang und zeigt andauernde Oszillationen, deren Amplitude für $\sigma = 0$ auffällig groß ist. Die anderen beiden Kontrollregime besitzen 40, also die Summe des individuell Erlaubten als Untergrenze. Lediglich für $\sigma = 1$ wird das Basis-Gleichgewicht 64 (ohne Kontrollregime) in weniger als 10% der Fälle überschritten.

Im Makroverlauf können wir also den empirischen V-Effekt im Prinzip nachstellen, wenn auch das Ausmaß der Zugriffe im Falle $\sigma = 0$ zu extrem ist. Es ist nun zunächst zu prüfen, ob dieses der Wahl der konkreten Ausgangswerte geschuldet ist.

In einer ersten Serie (hier: Lauf 21 – 23) beschäftigen wir uns mit der Frage welchen Einfluß die Wahl der Zufallszahlen auf das Ergebnis hatte, in einer zweiten Serie (hier: Lauf 31 – 33) gingen wir der Frage nach, welcher Einfluß der konkreten Typenverteilung zukommt.

In der Tat ist auch bei hundert Perioden die zufallsinduzierte Variation des mittleren Zugriffs noch erheblich. Dabei ist uns nicht bekannt, ob bei längerer Laufzeit diese Variation schließlich, zumindest für gewisse Kontrollregime, verschwindet. Lauf 41 mit $T = 1000$ zeigt jedoch, daß für "verhältnismäßige" Bestrafung ($\sigma > 0$) der Gesamtzugriff mit der Zeit stark reduziert wird. Für $s = 2$ blieb er hier sogar konstant 40 (individuell jeweils 5). Es wird in letzterem Falle also schließlich doch "gelernt" dem erhöhten Druck der Strafen nachzugeben. Inwieweit dieser Befund auch realistisch ist, können wir nicht beurteilen, zumal u.E. entsprechend lange Erfahrungszeiten kaum zu Verfügung stehen dürften. Dennoch ist der Befund für uns Anlaß genug, nach stabileren Systemen zu suchen. Es bietet sich dazu entweder die Typ B Variation (Lauf 61) oder die jeweiligen bimodalen Varianten A² (Lauf 51) und B² (Lauf 62) an. Im Anhang sind unsere Auswahl der Simulationsläufe dokumentiert. Die folgende Tabelle gibt eine Übersicht:

Name	Perioden (T)	Variante	Mittlere Gesamtzugriffe ($\sigma = 0$ $\sigma = 1$ $\sigma = 2$)
Lauf 11	100	si6.A.221100	(83.11 54.59 61.38)
Lauf 21	100	si6.A.221100	(87.24 53.43 59.63)
Lauf 22	100	si6.A.221100	(90.43 44.74 63.77)
Lauf 23	100	si6.A.221100	(82.40 52.32 63.11)
Lauf 31	100	si6.A.000000	(67.64 44.85 52.24)
Lauf 32	100	si6.A.111111	(42.08 40.79 52.15)
Lauf 33	100	si6.A.222222	(63.08 42.21 49.43)
Lauf 41	1000	si6.A.221100	(83.478 42.229 46.742)
Lauf 51	1000	si6.A ² .221100	(98.482 57.770 64.073)
Lauf 61	100	si6.B.221100	(47.35 44.37 48.63)
Lauf 62	1000	si6.B ² .221100	(60.084 51.810 53.166)
Lauf 71	100	si7.A.221100	(63.70 64.16 58.33)

Zwar wird der V-Effekt in der B-Variation erhalten, jedoch entsteht ein gegenüber dem Experimental extrem braves Zugriffsverhalten. Dennoch lehnen wir diese Implementation zunächst nicht ab, da es möglich sein müsste, durch Wahl der Hemm-parameter bessere Anpassung an das Experiment zu erzeugen. Verworfen wurde allerdings die Implementation si7, bei der es uns trotz aller Variation nicht gelang, Makroeffekte des Experimentes nachzustellen. Die Implementation si7 unterscheidet sich von si6 dadurch, daß die Entladung des Aggressionsspeichers nicht vollständig, sondern im Verhältnis zum Ausmaß des Aggressionsszuwachses vorgenommen wird. Darüber hinaus wurden auch weitere Ausmaßbestimmungen für die Katharsis geprüft und verworfen.

Kehren wir also zu Variante A zurück, und zu der Frage, welcher Einfluß der konkreten Typenverteilung (H_1, \dots, H_n) zukommt. In obiger Tabelle wird deutlich daß, die Typenverteilung den mittleren Zugriff stark und zudem in nicht monotoner Weise moderieren. Das Ausmaß hat eine Größenordnung, die den Einfluß anderer Variationen für die mittlere Dauer ($T = 100$) übertrifft.

Die Dominanz der Typenverteilung macht uns auch aufmerksam auf das unterschiedliche Verhalten und die unerschiedlichen internen Zustände der Agenten. Im Anhang ist auch dieses für die verschiedenen Läufe dokumentiert. Auffällig ist vor allem, daß die Zugriffsverteilungen und die Zugriffsvariatio über die Agenten realistisch nachgebildet werden kann.

4.8 Anhang

```

si6sigma
)load si6sigma
saved 1997-08-24 12:55:21 []pw+110

LEV
3

X
5 5 5 5 5 5 5

Z
4 3 2 1 4 3 2 1
2 2 1 1 1 1 0 0

b+kaus a;ue;h;t
^ a EINGABEVEKTOR, b AUSZAHLUNG NACH KONTROLLE
'SIGMA = ', SIGMA ^ 'ZUGRIFFE UND gESAMT: ', x, +/x
'vor kontrolle' ^ s+/a
[]+b+a*(Pa)P18-s+4
v+nP0 ^ ENTDECKTE VERSTÖßE
ue+(a>5)*a-5 ^ zuviel
ll:-(46>+/a)/0
'zufallsauswahl' ^ []+h+?n
v[h]+ue[h] ^ ENTDECKTES zuviel
b[h]+b[h]-ue[h]*40 ^ entdeckte zahlen die kontrolle
b+b-5*ue[h]=0 ^ KEIN VERSTÖß
b+b+SIGMA*v[h]-n*v ^ transfers
'nach kontrolle' ^ b

p+si lauf bis;l
l+0 ^ p+(0,2*n)P0 ^ SIGMA+si ^ x+X ^ z+Z
ll:-(bis<l+1+1)/0
p+p, [1] x, ,z[1;] ^ runde ^ +11

runde
^ SCHREIBT AKTUELLE kontrolle, evaluation UND reaktion
u+kaus x ^ i+0
z+affekt
l0:x[i]+reak i+i+1
+(n>i)/10
'NEUES x: ' ^ x

```

FIGURE 4.1. Hauptprogramm

```

r+5*s<100
+(z[1;w]<3)/0 ^ aggression KLEIN

LEV=5 oder 6

+((x>5)*s>100)/(r+5)*0 ^ furcht

+((z[2;w]=2)*(v[w]>4)*(s<100))/(r+x[w])*0 ^ fehlversuche wiederholen
+((z[2;w]>1)*(v[w]>0)*(u>40))/(r+x[w])*0 ^ fehlversuche wiederholen

```

FIGURE 4.2. Früher getestete Alternativen

```

r+reak w;lev
r+5*(s-x[w])<100
lev+z[1;w]-z[2;w]
+(LEV>=lev)/10 # hemmung ÄRGER: anstauen
r+5+ / 1 2 2 5 5 5 *(z[1;w]-LEV)>=16
+(r=5)/0 # z[1;w]+f/0,z[1;w]-2*f[w] # ->0
10:r+r+(z[1;w]<LEV-1)*+/4*(36>s-x[w])*x[w]<=5 # gelegenheit

q+hfg p;min;max;r
min+l/,p # max+f/,p
q+min+0,1max-min # r+pq
q+q,[0.5]+/(r,1)pq)=.(1,p,p)p,p

drei
[0.5] 0p[ts[7]?1000
p0+0 lauf 100
p1+1 lauf 100
p2+2 lauf 100
+ / # 0 ^8+ 3 16 p+/[1] # +p0,p1,p2
hfg 0 ^8 +p0
hfg 0 ^8 +p1
hfg 0 ^8 +p2
    
```

FIGURE 4.3. Später getestete und verworfene Alternativen

1145	1140	1055	1084	1070	1040	894	883
735	760	665	696	677	672	628	626
809	829	731	776	738	754	749	752

FIGURE 4.4. Vp-Statistik: Zugriffe (Lauf 11, T=100) in Apliwin: $\sigma = 0, 1, 2$

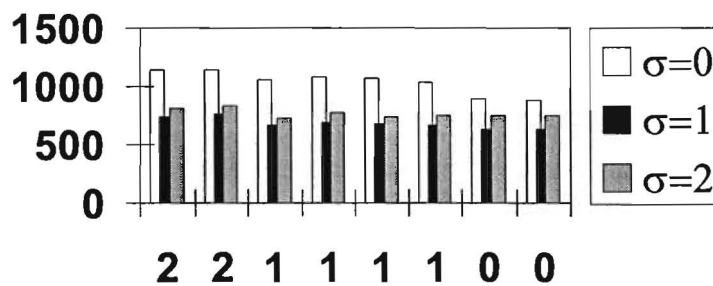


FIGURE 4.5. Vp-Zugriffe (si6, T = 100)

σ	0	1	2
in der Simulation	83.11	54.59	61.38
im Gleichgewicht	60.25-62.5	55.67-58.8	53.33-55
im Experiment	60.79	53.49	57.96 (59.33)

FIGURE 4.6. Zugriffe je Runde

203	202	178	187	188	181	66	65
201	189	158	128	157	110	90	86
193	175	143	122	122	142	86	88

FIGURE 4.7. Vp-Statistik: Aggressionen

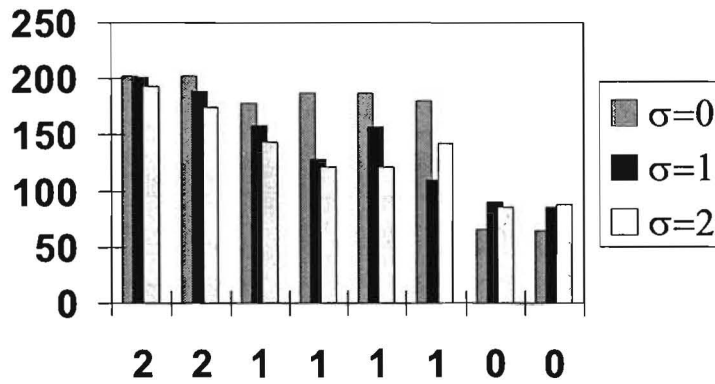


FIGURE 4.8. VP-Agressionsstau ($si6, T = 100$)

σ	0	5	6	8	9	10	15	20	25
0	80	332	0	66	2	42	70	113	95
1	0	516	24	87	21	99	40	13	0
2	0	473	22	88	18	102	39	56	2

FIGURE 4.9. Häufigkeiten

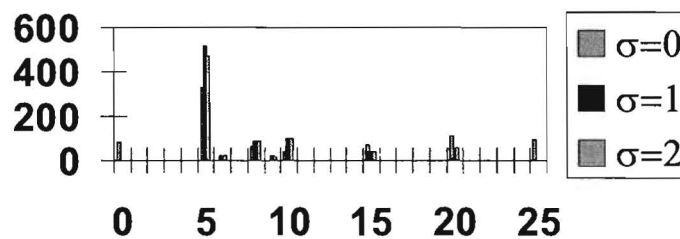


FIGURE 4.10. Zugriffsverteilungen

Towards the Bounded Optimal Agent Society

Christian Gerber¹ and Christoph G. Jung^{2 3}

ABSTRACT

The *bounded rational agent* is nowadays a well understood and agreed paradigm in *Cognitive Science*. Yet, the generalization of theories, such as *bounded optimality* [17], to *societies of cognitive entities* has not yet been investigated systematically in theory and practice. We address these aspects from the perspective of *Distributed Artificial Intelligence* (DAI) and develop a *hierarchical resource adaption* scheme that links the *micro-level* of a *hybrid* and *layered* agent architecture to the *macro-level* of their society. Our device of *virtual resources* as an abstraction of quantitative environmental and architectural constraints relates to considerations of cognitive psychology.

5.1 Introduction

During the last decade, a new, shared focus in *Cognitive Science* has emerged around the notion of *bounded rationality* [19]: researchers today widely agree on intelligence no more being viewed as an *in-principle capability* of solving abstract problems, but rather being measured by performance with respect to (dynamically changing) environmental conditions. We use the term *resources* to describe these, mostly quantitative, interactions that constrain the *situated agent*. *Resource adaption* thus turns out to be the key to either enrich simple formalisms for doing rational decisions [1] or to guide sophisticated and therefore complex computation towards the appropriate solution [17].

Encouraged by the success of *distributed computing* and the prospect of intelligent *divide-and-conquer* methods, *Distributed Artificial Intelligence* (DAI) [11] extends the theoretical and practical agent framework of Cognitive Science [4, 2, 16] to describe *societies of agents*, especially heterogeneous *multi-agent systems* (MAS). However, neither the connection to the *bounded rationality* concept nor the relevance of advanced, psychological theories to social structures has yet been addressed herein. From the complementary standpoints of *scalable*⁴, human-oriented MAS applications, e.g., in the emerging global network, in traffic telematics, and in manufacturing systems, and of the cognitive modeling of social phenomena, there is an urgent demand to investigate an integrating methodology.

Section 5.2.1 shows that the advanced theory of *bounded optimality* [17] can be extended to MAS. However, to uphold the practical aspects of tractable and scalable resource control (Section 5.2.2), we have to apply the DAI device of distribution to a generalized notion of resources (*virtual resources*). In contrast to, e.g., *social rationality* [10], our hierarchical approach furthermore connects the *macro-level* of the agent society (Section 5.3.1) with the *micro-level* of *layered* single-agent architectures such as the INTERRAP model [14] (Section 5.3.2). Finally, Section 5.4 relates our engineering constructs to mentalistic notions and respective research in cognitive psychology. We primarily discuss issues on adopting efficient human decision making and on providing for human-oriented design. The possibility

¹Multi-Agent Systems Group, DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, gerber@dfki.de

²GK Kogwiss. & FB Inform., Univ. des Saarlandes, Im Stadtwald, D-66123 Saarbrücken, Germany, jung@ags.uni-sb.de

³This work has been supported by Siemens AG Germany and the “Deutsche Forschungsgemeinschaft” (DFG).

⁴i.e., systems that are able to adapt to any application size

of using such systems as social simulation tools still accounts for a stronger interpretation of *cognitive adequacy*.

5.2 The Bounded Optimal Agent Society

5.2.1 DEFINITION

In [17], one of the most advanced accounts of bounded rational agents, Russell and Wefald define *bounded optimality* as a property defined on a set of agents \mathcal{L} regarded as programs or states. The environment in which these agents are situated is represented by a given transition $M : E \times \mathcal{L} \rightarrow E \times \mathcal{L}$ of environmental states E . The bounded optimal agent $l_{opt} \in \mathcal{L}$ with respect to M solves the constrained optimization problem $l_{opt} = \operatorname{argmax}_{l \in \mathcal{L}} \lambda x. M(x, l)$. Thus it maximizes the outcome of functions $\lambda x. M(x, l)$ that are containing the possible evolution of the world given an agent program.

Originally, agent architectures were supposed to approximate optimal behavior based on pure *object-level* decisions. Bounded optimality however reveals that this cannot be reached in a domain-independent manner without solving the higher-order problem of additionally approximating the optimal agent program. This justifies the application of *meta-reasoning* in the proposed, practical architecture of [17]: *Internal*, *architectural* or *computational*, resources are *assigned* to and *monitored* from a set of optional courses (or programs) on the object-level. Internal resources are those environmental constraints that only affect the interpretation of the agent program. They have to be distinguished from the object-level, *external* interactions.

As the meta-reasoning component also applies for internal resources, it should be of neglectable complexity and applies simplification:

- *Meta-greediness*: When going from simple (one-step) to sequential decision problems, there will be an exponential growth in complexity. Tractable resource assignment therefore only applies successive, simple decisions with locally optimal character and is called *meta-greedy*. It is assumed that this approximates the optimal solution of the overall sequential problem.
- *Independence*: Complex resource interactions on the object-level, e.g., caused by shared data structures, are another source of complexity explosion. It is assumed that the optional courses controlled by the meta-level are approximately *independent* with respect to computational resources.
- *Predictability*: To come to a decision, the meta-level must have an reasonable *a priori estimation* of the utility of the object-level's optional computations.

At first sight, the definition of [17] can be straightforwardly extended to the multi-agent case, provided that all agents of a certain society share a common social goal. Let the environment be a mapping $M : E \times \mathcal{L}^* \rightarrow E \times \mathcal{L}^*$ that determines the transition of the environment and the possibly simultaneous program reduction of agents belonging to one society. The *bounded optimal society* is a selected tuple of agent programs $\vec{l}_{opt} \in \mathcal{L}^*$ as a solution to the constrained optimization problem $\vec{l}_{opt} = \operatorname{argmax}_{\vec{l} \in \mathcal{L}^*} \lambda x. M(x, \vec{l})$.

This does not look to far away from the original, single-agent model of [17] by letting $\mathcal{L}_{new} ::= \mathcal{L}^*$. Unfortunately, the assumptions of a respective, centralized architecture cease to hold as it fails to exhibit scalability in general:

- *Object-level*: The complexity of the object-level (combining all perception histories and producing reasonable, simultaneous decisions) is intractable.
- *Meta-Level*: The same problem occurs in assigning and monitoring resources on the meta-level. Additionally, social laws and structures blur the distinction into external and internal resources. Thus, it is not clear how to reasonably combine the outcome estimations. Furthermore, the independence of the object-level's optional computations is no more plausible.

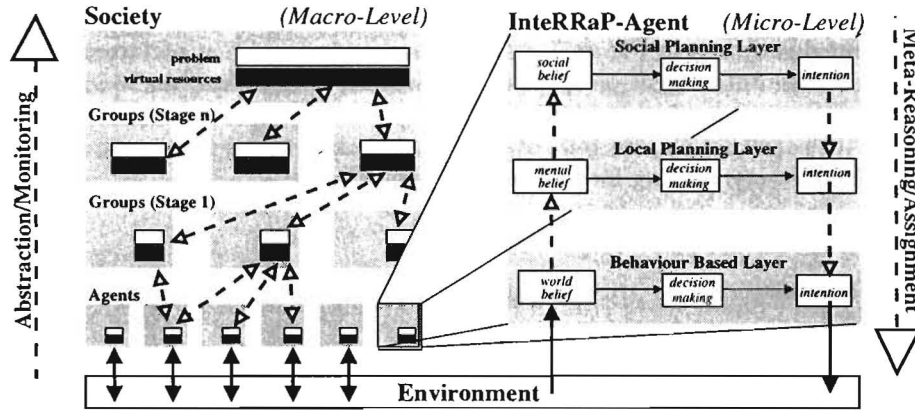


FIGURE 5.1. Hierarchical Resource Control & The InteRRaP Architecture

5.2.2 A HIERARCHICAL MODEL OF RESOURCE CONTROL

On the one hand, DAI research builds on the foundations of *Artificial Intelligence (AI)* by *constructing* locally intelligent agents, groups, and societies with a certain potential of (object-level) problem solving. On the other hand, a *reduction* strategy is employed to *distribute* a complex, monolithic problem into partial tasks to be carried out by societies, groups, and agents. The loss of interdependencies between the partial tasks hereby leads to sub-optimality and inconsistency. Therefore, either explicit (by coordination agents) or implicit (by interaction and negotiation) social decision mechanisms should guide and incorporate the partial problem solvers towards the optimal solution. In contrast to the rigid control in traditional distributed computing, the ability of self-organization and adaptivity to dynamic, inherently distributed problems renders the DAI paradigm an excellent tool providing for *robustness* and scalability. Additionally, it offers the natural, hierarchical modeling device of societies, groups, and agents.

While we have just discussed the intractability of optimal and centralized resource control in MAS, our investigation of the DAI paradigm delivers us the solution to the dilemma: a straightforward, though by no means trivial, approach is to make the approximation of bounded optimality (the meta-level problem) a part of the distributed (object-level) problem description (Figure 5.1). Each social stage in this description (society, group, individual agent, agent layer) is no more to only distribute and maintain its portion of the global task to its subordinate entities. It is also responsible for reorganizing the substructure.

As aforementioned, the clear distinction of both functionalities is however not possible: for instance, agents applying for computational resources, affect the group stage in a similar fashion, no matter if they interact internally by running on the same computing device, or if they interact externally by performing actions in the physical world. Thus we abstract away from this: The term and the representation of *virtual resources* captures any general interdependency, be it internal or external, between any problem solvers forming a social entity. Virtual resources are assigned and monitored which amounts to putting “guidelines” on the refinement of these decisions on the lower stages.

A very primitive, discrete example of such a virtual resource could be the well-known construct of a *semaphore*. It restricts certain activities applying for this abstract concept to happen in a sequential manner, as only one of them is able to get a hold on it and therefore allowed to compute (internal use) or act (external use). We will see that more elaborate, even continuous models of resources enable the connection to the cognitive restrictions of humans and human society and the suitability as a modeling tool (see Section 5.4).

Section 5.3.1 outlines a reasonable resource adaptive control mechanism for the various social stages. Similar to [17], its assignment should be fast. Thus it also relies on the assumptions of meta-greediness, independence, and predictability. However, where [17] resides to one ‘manager’, our multiple-staged, hierarchical model provides for more abstract and long-term resource reasoning including conflict resolution by obeying guidelines from above (e.g., the society).

The final connection to the concrete environmental and architectural constraints is done at the lowest levels of our description in Figure 5.1, the layers within the agent architecture. Our approach does not reside to the outer-agent perspective, but instead tries to link the *macro-level* of global, social activity to the *micro-level* of decision making within the agent's internals. We explain in Section 5.3.2 how to fit the construction principles of *layered* agent architectures, here the INTERRAP architecture [14], into our society heading for bounded optimality.

5.3 Engineering Constructs

5.3.1 RESOURCE CONTROL IN STRUCTURED AGENT SOCIETIES

Resource control on the macro-level focuses on distribution of virtual resources among the whole agent society. Virtual resources on that stage can be derived from structural principles of the society and from principles of agent communication within the society. Examples for structural principles are:

- *Number of agents* in the society,
- *organizational form* of the society,
- *migration*, i.e., distribution of agents over the net, etc.

Communication principals can be expressed through

- *introduction of communication channels* between subunits or even between agents belonging to a common subunit,
- *usage of different communication protocols* leading to different communication complexities,
- *assignment of communication roles to agents* leading to rankings inside subunits, etc.

A more detailed enumeration and description can be found in [7]. As such an agent society can be decomposed to smaller organizational forms, similar, but refined virtual resources as stated above have to be controlled on all subunit stages (see the left-hand-side of figure 5.1).

Considering groups containing single agents we additionally regard suggestible properties of agent members as virtual resources. In particular, agent *characteristics* contribute to the *diversity* of a group; they describe inhibitory or stimulating guidelines on resource allocation at the agent architecture stage described in section 5.3.2. Actually, this establishes the so-called *micro-macro link* (MML).

Obviously, expensive modification on the highest society-stage has the greatest impact on the overall performance. Hence modification on that stage should be performed rather rarely. On the other hand, parameter modifications of a single agent do not weigh so much concerning the overall performance of the system. Thus, these parameters may be modified more dynamically.

Since this paradigm requires explicit resource reasoning, at each stage we realize control by commissioning an agent to care about structural optimality, either in addition to its original tasks or as its sole functionality. Thus, control agents incorporate an optimization mechanism to organize an agent group: the goal is to find an optimal distribution of virtual resources among the agent group depending on a measurement of performance inherent to the overall task specification. This measurement bases on several factors such as operating time, quality of the result, etc. The application designer has to combine them, e.g., by building a weighted sum.

As ideally required from our definition of bounded optimal society in section 5.2.1 the optimization has to run over possibly infinite, unpredictable sequences of states. Even an off-line optimization routine cannot meet these requirements. Hence, we employ a tractable, on-line computation that follows Russell and Wefald's meta-greediness assumption. It applies subsequent steps of local optimization based on the performance at the actual time t .

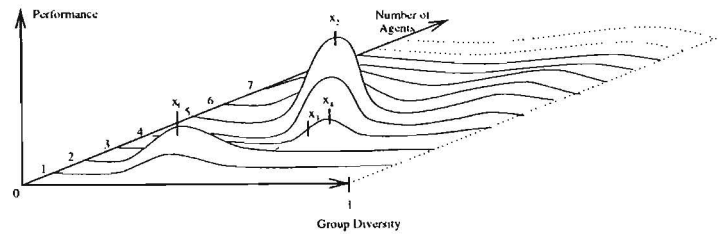


FIGURE 5.2. A Simple Example of a Performance Relation

Following the methodology of *Operations Research* (OR), the intended mechanism has to discover an optimum of an *objective function* in an multi-dimensional *search space*. The objective function coincides with the system's performance at time t while the search space must describe the set of possible configurations of the agent group: each modifiable quantity is represented by a dimension in the search space and each point in that search space corresponds to one particular system configuration.

Figure 5.2 shows a simple, two-dimensional example: here the objective function measures the efficiency of a group of agents. Performance depends only on the number of agents and the degree of diversity of the agent group. "Number of Agents" is represented by a discrete dimension, whose domain ranges from 0 to possibly infinity. "Group Diversity" is modeled as a continuous dimension, its domain ranging from 0 (i.e., all agents have identical characteristics) to 1 (i.e., their traits vary greatly).

Formally, the constrained optimization problem is characterized as follows:

$$\begin{array}{ll} \max & \text{Overall Performance} \\ \text{subject to} & \text{Distribution of resources on modifiable quantities} \\ & \text{Minimal and maximal boundaries for the use of each resource} \end{array}$$

A variation of Zoutendijk's steepest ascent method [21] for finding local optima is well suited for that task as it is able to react fast to situation changes in order to maintain high, but not necessarily optimal performance during the complete run of the application. Especially the loss of the temporal component of the objective function requires this permanent and dynamic control.

The point of concern is to find in each dimension a value with a high *marginal utility value*, i.e., a point where the investment for one more resource unit leads to a high performance *improvement*. A (local) optimal configuration can be found by moving from some arbitrary starting point in the search space iteratively to the optimal point. In every step, first the direction is determined in which to move. If the performance is representable by some differentiable function, the best direction can be determined using partial derivatives. In case of incomplete knowledge, the expected performance profile could be used instead.

In the above example a starting point may be x_3 . A maximal performance gain will be achieved by increasing the group's diversity: x_4 may be achieved. By increasing the number of agents to five, the optimal configuration (represented by x_2) will be found.

Basic assumptions of the steepest ascent algorithm can easily be approximated; some minor violation does not lead to a great loss of optimality in our domains:

- *The performance function is concave*: if this assumption is violated, the algorithm may miss a global optimum. For instance in the above example, by beginning from another starting point the algorithm might return x_1 as a result. This assumption corresponds to *meta-greediness*. As explained above, we do not require a mechanism to achieve global optima.
- *The search space is convex*: if this is not the case a "marginal step" denotes the smallest step possible to reach again a point in the search space. In the example the discrete dimension "Number of Agents" induces non-convexity. A marginal step along that dimension can be achieved by increasing or decreasing the agent number by one.
- *A total order is required for each dimension*. In any case, an order can be introduced artificially.

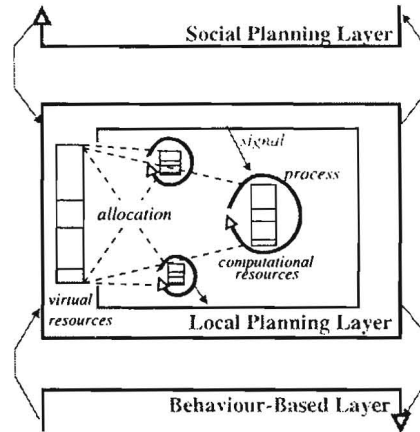


FIGURE 5.3. Processes, Layers, and Virtual Resource Assignment

- *During the search for optimization the performance function is stable*: The speed of our method guarantees only small discrepancy between expected and actual implication of a decision. This enables predictability and reactivity to unexpected dependencies between ideally orthogonal dimensions. A more detailed discussion on these assumptions can be found in [8].

5.3.2 RESOURCE CONTROL IN A LAYERED AGENT ARCHITECTURE

The micro-level of individual agents is finally responsible for transforming the abstract task and resource guidelines that are given by the just described social stages into computational and external actions. For this purpose we need to specify the further refinement and monitoring of these constraints in the light of the manifold capabilities that a practical, intelligent agent should incorporate:

- *reactivity/pro-activity* is the ability to install a short feedback loop with probably complex and dynamic environments.
- *deliberation* is the ability to solve abstract problems.
- *social competence* is the ability to adjust ones own decision making with the help or demand of other, heterogeneous agents by forms of communication.

Our *hybrid* approach, the INTERRAP architecture [14] (see right part of Figure 5.1), implements a smooth transition from sub-symbolic reactivity to symbolic deliberation and social capabilities in terms of *layers*. The *behavior-based layer* (BBL) applies procedural knowledge, the *patterns of behavior*, to obtain fast and routine decisions/actions (e.g., controlling sensors and actors). On top of this, the *local planning layer* (LPL) reasons (plans) a about how to reconfigure (in terms of a plan) the underlying reactive module to meet more long-term, complex goals (e.g., way-planning and task-planning). Finally, social decisions in negotiating/communicating (*social planning layer* — SPL) with other agents also use the device of planning: the commitment to adopt new goals or plans reconfigures the LPL.

By extending the original INTERRAP description to a formal specification, [12] develops a computational calculus that addresses the importance of concurrency for situated agents' acting while deliberating. Borrowing sophisticated designs from operating systems and modern programming languages, it employs a *process & signal* metaphor (Figure 5.3): Cognitive processes, e.g., perception, action, belief maintenance, reflexes, desires, behavior patterns, planning, and plans, are modeled as continuous, concurrent state transitions. These exchange information by sending *signals* to, e.g., deliver computation results or indicate exceptional situations, such as errors or environmental changes.

Layers are control structures built around processes of the respective stage, e.g., the BBL encapsulates the patterns of behavior and so-called reflexes. Originally, this control has focused on the

(de)activation of internal processes and the filtering of signals to influence information exchange, such as inhibition and excitation [3]. With respect to our enterprise, they can be as well used as additional stages in resource allocation: Each layer as a social stage reconfigures (maintains, monitors) the resource allocation of its direct sub-layer, but additionally retains some portion of virtual resources to its own processes. By mapping these virtual resources to the actual computational resource, i.e., time and space, processes get finally the possibility to compute and act.

The virtual resources within the agent include environmental aspects as *access to actors and sensors by semaphores* as well as architectural restrictions inspired by human cognition (*restricted knowledge access, potential of abstract problem solving, focus of attention*, etc., see Section 5.4). Again, we require the complexity of the resource mapping (Figure 5.3) to be neglectable in cost preserving a consistent and reactive system: the local adaption algorithm presented in Section 5.3.1 is applied again.

However, its subsequent steps are not solely determined by *decision-theoretic* considerations. Additionally, there are preferences that are reactively linked to coarse, environmental situations which can be interpreted as *motivational or emotional* tendencies. For example, the resource balance of competing processes that care about independent goals, like not to starve and not to be killed, could shift motivation in the presence of empty batteries or the recognition of some predator in the area. Inconsistent or randomized resource assignment preferences induces symptoms of paranoia. More *stereotypic, characteristic* constraints to the local adaption step even hardly change during the life of the agent as they represent his personality or characteristics. Introverted agents possibly do not assign much resources to their cooperative facilities in general; no resources available there even correspond to switching off the respective functionality of the layer. Passive agents forbear the execution of atomic actions.

In each case, the local adaption step is steered by the guidelines of its upper social stages and thus fits reasonably into our overall hierarchical model. We will come back to the relation of our constructs to human cognition in the next Section.

5.4 Cognitive Concern

5.4.1 MACRO-LEVEL

Group theorists have been studying behavior of human groups that have to solve a given task. In particular, scientists are interested in determining for a certain task the size and degree of diversity by which a group performs best. Experiments have shown that the optimal size of a cooperating group varies from five up to a dozen members (see [15], [18] or [5] for details), depending on the task and on properties of the test group. [13] compares characteristics of larger groups to those of smaller ones: Generally speaking, larger groups enjoy several advantages: They have more money, working time and expertise at their disposal, they tend to be more diverse. On the other hand, larger groups suffer several disadvantages: they often experience coordination difficulties which may decrease problem solving performance. Furthermore, there is more conflict among members of larger groups. In general, they are less willing to cooperate. These results show that there is no simple way to determine the perfect size of a human group. Hence, a better approach might be to study some of the correlates of group size: Relevant issues are *demographic characteristics* of group members (e.g., sex or race), their *abilities* (such as intelligence or expertise), *opinions* (for instance beliefs or values) and *personalities* (e.g., traits or needs) A great diversity (i.e., a great variance among these traits) of a group can improve its problem solving performance as specialists might be available to perform a certain task. The main risk of a great diversity is that it can produce conflicts among members.

The approach proposed in this paper enables us to characterize an artificial agent in similar terms used to describe humans. Hence, a group or society of agents is no longer characterized by only its number of agents, but also of its degree of diversity or organization and other parameters.

5.4.2 MICRO-LEVEL

We regard layered agent architectures as coinciding with some theories of the evolution of cognition [6] from stimulus-response systems to socially and even self-aware entities. Human ability to reflect and consciously guide their own decision making justifies a meta-reasoning framework. Similar to the human cognitive apparatus, simultaneous processes take place in our agents. Certain natural restrictions, like limited memory access, focus of attention, and the potential of abstract problem solving however seem to introduce a controlled form of concurrency thus avoiding the inherent computational complexity. By modeling these restrictions as virtual resources, we describe agents that exhibit a similar controlled inference. The connection of certain resource mapping parameters with motivation, emotion, or even characteristics is related to resource-oriented theories of motivation and emotion [20]. Section 5.3.2 has already discussed some examples of realizing human phenomena (hunger, fear, introversion, passivity. etc.) within our constructive architecture.

5.5 Conclusion and Future Work

We have presented a hierarchical resource adaption scheme for MAS that extends the theoretical framework and practical architecture of [17] to gather tractable and scalable bounded optimality. The concept of resources is generalized by introducing an abstract, virtual resource representation that is linked to the architectural and environmental situation of the MAS. It thus allows to smoothly bridge the macro-micro-gap found in complex systems. Due to their detachedness of the detailed implementation realities we could compare these constructs with well-established theories of the human mind on the one hand and of group behavior on the other hand. The mentalistic notions also help a human designer to design such a system and to interact with it. Cognitive scientists speak of *weak equivalence* to human society or mind.

An implementation and evaluation of our work will use the INTERRAP architecture which is currently under development both in OZ[9] and JAVATM. It will make heavy use of sophisticated resource control built into these languages. Furthermore, constraint-based methods, as integrated in OZ, seem to be a natural way of describing the steepest ascent algorithm. Future work will also investigate the benefits of a learning component to adjust the expected value of object-level computation and thus to support our resource adaption. It may be integrated into the resource-control components on both the agent architecture and the agent group/society level. With respect to cognitive science, our toolkit will provide the possibility of running simulations to explain empirical examinations pursued in the fields of cognitive and social psychology. The question remains whether our mentalistic notions of motivation, emotion, and characteristics could be seen as side-effects of nature's way to controlled inference.

5.6 References

- [1] J. R. Anderson. *The adaptive character of thought*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.
- [2] J. R. Anderson. *Rules of the mind*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.
- [3] R. A. Brooks. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, 1991.
- [4] J. G. Carbonell, C. A. Knoblock, and S Minton. PRODIGY: An integrated architecture for planning and learning. In *Architectures for Intelligence*. Lawrence Erlbaum Associate, 1991.
- [5] M. Cini, R. Moreland, and J. Levine. Group staffing levels and responses to prospective and new group members. *Journal of Personality and Social Psychology*, 65:723–734, 1993.
- [6] P. Gärdenfors. Cued and Detached Representations in Animal Cognition. *Behavioural Processes*, 35:263–273, 1996.

- [7] C. Gerber. An Artificial Agent Society is more than a Collection of “Social” Agents. Working Notes of the AAAI Fall 1997 Symposium on Socially Intelligent Agents, To Appear, 1997.
- [8] C. Gerber. Scalability of Multi-Agent Systems. PhD Thesis Proposal, Universität des Saarlandes, Saarbrücken, Germany, 1997.
- [9] Gert Smolka. *An Oz Primer*. DFKI Oz Documentation. Saarbrücken, Germany, April 1995.
- [10] L. M. Hogg and N. R. Jennings. Socially Rational Agents - Preliminary Thoughts. In J. Bell, Z. Huang, and S. Parsons, editors, *Proceedings of the 2nd Workshop on Practical Reasoning and Rationality*, April 1997.
- [11] M.N. Huhns, editor. *Distributed Artificial Intelligence*. Research Notes in AI. Morgan Kaufmann, 1987.
- [12] C. G. Jung and K. Fischer. A Layered Agent Calculus with Continuous, Concurrent Processes. In *Intelligent Agents*, volume 4 of *Lecture Notes in Artificial Intelligence*. Springer, 1997. to appear.
- [13] R. Moreland. Creating the ideal group: Composition effects at work. In E. Witte and J. Davis, editors, *Understanding Group Behavior*, volume 2. Lawrence Erlbaum Associates, Publishers, Mahway, NJ, 1996.
- [14] J. P. Müller. *The Design of Intelligent Agents: A Layered Approach*, volume 1177 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, December 1996.
- [15] D. Nasser. How to run a focus group. *Public Relations Journal*, 44:33–34, 1988.
- [16] A. Newell. *Unified theories of cognition*. Harvard University Press, Cambridge, London, 1990.
- [17] S J. Russell and E. Wefald. *Do the Right Thing*. MIT Press, 1991.
- [18] A. Scharf. How to change seven rowdy people. *Industrial Management*, 31:20–22, 1989.
- [19] H. A. Simon. *Models of Bounded Rationality*. MIT Press, 1982.
- [20] A. Sloman, L. Beaudoin, and I. Wright. Computational modelling of motive-management processes. In *Proceedings of the Conference of the International Society for Research in Emotions ISRE94*, 1994.
- [21] G. Zoutendijk. *Mathematical Programming Methods*. North-Holland, Amsterdam, 1976.

Roboter und Computer spielen Fußball

Hans-Dieter Burkhard¹, Markus Hannebauer und Jan Wendler

ABSTRACT

Anliegen des Roboter-Fußballs ist die Herausforderung neuer Ideen und technischer Lösungen im Bereich der Robotik und der Künstlichen Intelligenz. Im Rahmen der IJCAI-97 fanden vom 25.-28.8.97 in Nagoya die ersten Weltmeisterschaften im Roboter-Fußball in drei Wettbewerben (mittlere Roboter, kleine Roboter, Simulationsliga) statt. Der Beitrag gibt einen Bericht vom RoboCup. Er will gleichzeitig deutlich machen, daß sich hier ein weites Feld für reizvolle Aufgaben in Forschung und Lehre bietet.

6.1 Das Anliegen

Das Anliegen des Roboter-Fußballs [3] ist die Herausforderung neuer Ideen und technischer Lösungen im Bereich der Robotik und der Künstlichen Intelligenz. Das betrifft Hardware und Software gleichermaßen. Ein (maschineller) Fußballspieler muß sich in schnell wechselnden Situationen orientieren, er muß sich zielgerichtet bewegen, den Ball in geeigneter Weise abpassen, annehmen und abspielen können. Ein Mensch braucht Jahre, bis seine Wahrnehmung und seine motorischen Fähigkeiten dafür ausreichend entwickelt sind. Es bedarf weiterer Jahre, bis er das Zusammenspiel in einer Mannschaft und das Reagieren auf gegnerische Spieler beherrscht. Für Technik und Wissenschaft stellt der Roboter-Fußball vielfältige Aufgaben: Sensorik, Aktorik, Bildverarbeitung, autonome Agenten, Multi-Agenten-Systeme, Echtzeitverarbeitung, Bewegungs- und Aufgabenplanung, Lernen von Fähigkeiten, Erkennen und Anpassen an gegnerisches Verhalten, Umweltmodellierung, Navigation, Kooperation, Ihre Bearbeitung kann nur schrittweise gelingen und wird Jahrzehnte brauchen. Die Initiatoren verweisen darauf, daß es 40 Jahre gedauert hat, bis ein Computer den Schachweltmeister schlagen konnte. Sie möchten den "RoboCup" als die neue Herausforderung für die Entwicklung intelligenter Techniken sehen. Interessanterweise werden damit eher Fähigkeiten aus dem täglichen Leben anvisiert. Beim Schach gibt es klar umrissene Problemstellungen, und die Spieler sind zu jedem Zeitpunkt über die aktuelle Situation vollständig informiert. Sie haben mehrere Minuten Zeit zum Überlegen, eine Fehlentscheidung hat allerdings auch katastrophale Folgen für den Spielverlauf.

Gutes Schachspielen ist vergleichbar mit dem Lösen mathematischer Probleme. Beim Fußballspiel wechseln die Situationen dagegen sehr schnell. Die Spieler haben gerade in kritischen Situationen häufig nur unscharfe und unvollständige Informationen, müssen dabei aber schnell handeln. Dafür können Fehler aber oft noch korrigiert werden. Fußballspielen ist eher vergleichbar mit der Orientierung und mit körperlichen Fertigkeiten in der Alltagsumwelt. Es ist klar, daß die technologische Beherrschung entsprechender Aufgaben sowohl in der Robotik als auch in der Entwicklung intelligenter Software vielfältigen Nutzen bringen kann. Sie erfordert die Integration vieler Bereiche. Mit den Wettbewerben im Bereich des Roboter-Fußballs soll der Entwicklungsstand sichtbar werden, und die weitere Entwicklung soll vorangetrieben werden.

¹Lehrstuhl für Künstliche Intelligenz, Institut für Informatik, Humboldt Universität zu Berlin, {hdb,hannebauer,wendler@informatik.hu-berlin.de}

6.2 RoboCup-97: Die erste Weltmeisterschaft im Roboter-Fußball

6.2.1 DER WETTBEWERB

Im Rahmen der IJCAI-97 fanden vom 25.-28.8.97 in Nagoya die ersten Weltmeisterschaften im Roboter-Fußball statt. Gespielt wurde in drei Wettbewerben:

- Kleine Roboter bis 15 cm Durchmesser (bzw. 10 x 18 cm): Das Spielfeld ist hierbei eine Tischtennisplatte mit fester Begrenzung, und gespielt wird mit einem orangefarbenen Golfball. Gespielt wird zweimal zehn Minuten. Maximal 5 Spieler sind pro Mannschaft zugelassen. Die Spieler dürfen den Ball nicht festhalten, lediglich der Torwart darf innerhalb des Strafraums maximal 10 Sekunden den Ball halten. Es gibt eine Kamera über dem Spielfeld, die jeweils einen globalen Überblick vermittelt. Dieses Bild kann von externen Computern ausgewertet werden, die ihrerseits über Funk die Spieler steuern können. Dieses Vorgehen ist aber nicht notwendig, vielmehr wird die völlige Autonomie der Spieler angestrebt. Zumindest für die Wettbewerbe der mittleren Roboter wird bereits erwogen, die vollständige *on-board*-Verarbeitung zwingend vorzuschreiben.
- Mittlere Roboter bis 50 cm Durchmesser (bzw. 45 x 45 cm): Länge und Breite des Spielfeldes betragen das Dreifache des Spielfeldes der kleinen Roboter. Gespielt wird mit einem roten Fußball, die Spielfläche ist grün, und die Tore sind gelb bzw. blau. Im übrigen gelten die gleichen Regeln wie bei den kleinen Robotern.
- Simulationsliga: Hier wird in einer virtuellen Welt mit virtuellen Spielern gespielt. Grundlage ist der SoccerServer, ein frei verfügbares Simulationssystem, das von Itsuki Noda (Electro Technical Laboratory, Japan) nach den Vorgaben der RoboCup Federation entwickelt wurde. Die Visualisierung erfolgt durch den zugehörigen SoccerMonitor. Die Halbzeiten dauern jeweils fünf Minuten. Der SoccerServer simuliert alle Bewegungen des Balles und der Spieler auf einem Spielfeld von der Größe eines Original-Fußballfeldes. Die Klienten muß man sich als die Gehirne der Spieler vorstellen, während die materielle Welt durch den Server imitiert wird. Jede Mannschaft besteht aus 11 autonomen Programmen, die lediglich mit dem Server kommunizieren dürfen (näheres dazu siehe unten). Die Berechnungen folgen den physikalischen Gesetzen von Krafteinwirkungen und langsam abklingenden Bewegungen, dabei werden auch zufällige Abweichungen einkalkuliert.

In Nagoya gab es 5 Teilnehmer bei den mittleren Robotern, 4 Teilnehmer bei den kleinen Robotern und 29 Teilnehmer in der Simulationsliga. Bei den realen Robotern gab es noch viele grundlegende technische Probleme. Sie zeigen, welche Anstrengungen notwendig sind, um in einer komplexen Umgebung koordinierte Abläufe zu erzielen. Schon der Übergang vom Labor in eine Messehalle mit neuen Lichtverhältnissen und zahlreichen Störsignalen überforderte einige Teams. Im Extremfall standen die Roboter bewegungslos oder gleichmäßig kreisend auf dem Spielfeld. Auch die Tore fielen zuweilen eher zufällig. Probleme ergaben sich generell dann, wenn der Ball am Rand zu liegen kam: Hier mußte meistens ein erneuter Anstoß den Fortgang des Spieles ermöglichen. Herausragendes Team war die Mannschaft *CMUnited* von der Carnegie Mellon University (Manuela Veloso, Peter Stone et al.), die bei den kleinen Robotern angetreten waren. Sie konnten alle Spiele mit klarer Überlegenheit gewinnen. Bei den mittleren Robotern gab es ein torloses Finale zwischen dem *Dreamteam* von der University of Southern California (Wei Min Shen et al.) und den *Trakies* von der Universität Osaka (Minoru Asada et al.).

Frei von den Problemen der realen Welt konnten dagegen in der Simulationsliga unterschiedliche Ideen aufeinandertreffen. Herausragende Mannschaften waren hier das Simulationsteam von *CMUnited* (wiederum Manuela Veloso und Peter Stone), die beiden Mannschaften des Tokyo Institute of Technology (Tomohiro Andou bzw. Masayuki Ohta), *ISIS* von University of Southern California (Milind Tambe) und das Siegerteam *AT-Humboldt* der Autoren dieses Beitrags. Aber auch die schon früh im k.o.-System ausgeschiedene italienische Mannschaft aus Padua (Enrico Pagello) zeigte sehenswerten virtuellen Fußball. Bei den erfolgreichen Programmen spielten insbesondere agentenorientierte Ansätze und über Lernverfahren entwickelte *Fähigkeiten* eine Rolle. Als weitere Preise wurden vergeben:

- RoboCup Engineering Challenge Award für das australische Team *RMIT Raiders* (Andrew Price, Andrew Jennings et al., Melbourne) und für das japanische Team *Uttori United* (Kazutaka Yokota, Hajime Asama et al., Tokio) für die originellen Antriebskonstruktionen. In Umkehrung des Prinzips der Computer-Maus wird der australische Roboter durch eine motorgetriebene Kugel bewegt. Mehrere hundert Einzelteile wurden dazu selbst entworfen und hergestellt. Auch der japanische Roboter kann sich mit Hilfe von Rädern auf allen vier Seiten in jede beliebige Richtung bewegen.
- RoboCup Scientific Challenge Award für Sean Luke (University Maryland) für seine mit evolutionären Verfahren entwickelte Mannschaft. Die in mehreren Millionen simulierten Entwicklungsjahren entstandene Mannschaft gehörte zum Mittelfeld des Wettbewerbs.

6.2.2 UNSER WEG INS FINALE

Die Vorrundenspiele wurden klar gewonnen mit 25:0 gegen *Team Sicily* (Stanford University, Lyen Huang), 23:0 gegen Chuubu University (Japan, Tomoo Inden), und 7:4 gegen Tokyo Institute of Technology (Japan, Tomohiro Andou). Danach wurde im k.o.-System um den Einzug ins Finale weiter gespielt: 7:0 gegen Chukyo University (Japan, Susumu Takagi), 14:7 gegen die 2. Mannschaft des Tokyo Institute of Technology (Japan, Masayuki Ohta) und 6:0 gegen *CMUnited* (Carnegie Mellon University, Peter Stone und Manuela Veloso). Im Finale trafen wir wieder auf die 1. Mannschaft des Tokyo Institute of Technology (Japan, Tomohiro Andou). Durch eine Umstellung und verändertes Rollenverhalten in der Verteidigung gelang es uns, die Angriffskraft des Gegners einzuschränken und so im Finale einen klaren 11:2-Sieg zu erringen².

6.3 Die Realisierung

6.3.1 DIE SIMULATIONSUMGEBUNG

Um ein gemeinsames Arbeiten der Forschungsgemeinschaft auf dem Gebiet der virtuellen Fußballspieler zu ermöglichen, wird eine standardisierte Simulationsumgebung benötigt. Das im Moment verwendete System stammt aus Japan und wird von Itsuki Noda betreut. Es besteht aus mehreren Programmen, von denen zwei eine zentrale Rolle spielen. Der SoccerServer stellt das eigentliche Simulationsprogramm dar, während der SoccerMonitor die Spielsituation visualisiert.

Abbildung 6.1 macht den Zusammenhang zwischen Spielerprogrammen, SoccerServer und SoccerMonitor klar. Im Client-Server-Prinzip melden sich die zweiundzwanzig Spielerprogramme beim SoccerServer an. Dafür werden entsprechend viele unsichere UDP/IP-Sockets verwendet. Der SoccerMonitor kommuniziert ebenfalls über einen Socket mit dem SoccerServer, so daß auch andere Visualisierungsmöglichkeiten genutzt werden können (auf dem RoboCup97 wurde zum Beispiel auch ein Pseudo-3D-Monitorprogramm eingesetzt). Haben sich alle Spielerprogramme angemeldet, kann das Spiel vom SoccerMonitor aus gestartet werden, und der SoccerServer beginnt mit der Simulation. Zwei Schnittstellen ergeben sich bei der Kommunikation zwischen dem SoccerServer und den Spielerprogrammen: Der Server sendet textuelle Sicht- und Hörinformationen in festgelegten Intervallen an die Spielerprogramme, und diese reagieren darauf mit der Übermittlung von Kommandos aus einem festen Handlungsrepertoire.

Die Sichtinformationen enthalten zum jeweiligen Spieler relative Angaben über Entfernung und Abweichung von der Sichtrichtung in Grad. Bei dynamischen Objekten, wie Spielern und dem Ball, werden zusätzlich Werte übermittelt, die einen Eindruck vom relativen Geschwindigkeitsvektor dieser Objekte vermitteln. Dabei hat der Spieler nur einen eingeschränkten Sichtsektor und weit entfernte Objekte werden zudem nur ungenau gesehen, z. B. geht bei Spielern zunächst die Spielernummer und schließlich sogar die Teamzugehörigkeit verloren. Objekte in unmittelbarer Nähe, die sich außerhalb

²Die dafür notwendige Programmänderung erfolgte zwei Stunden vor dem Finale, und die Tests wurden zwecks Geheimhaltung am verdeckten Bildschirm ausgeführt.

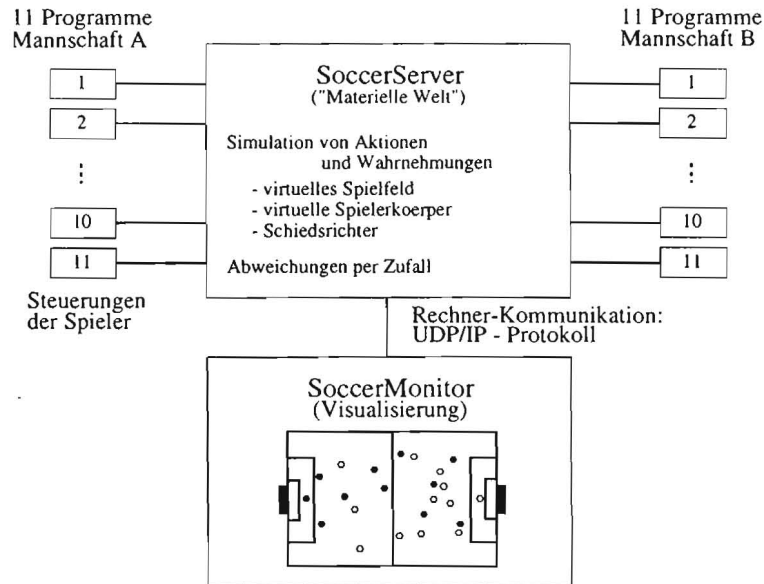


FIGURE 6.1. Zweiundzwanzig Spieleragenten und der SoccerMonitor melden sich per UDP/IP beim Soccer-Server an

dieses Sichtsektors befinden, werden zwar bemerkt, ihre Beschreibung ist aber ebenfalls unvollständig.

Informationsaustausch zwischen Spieler und Server

In der dargestellten Spielsituation erhält der rote Torwart zur Zeit 1979 (Maßeinheit: Zehntel-Sekunden) vom Soccerserver die folgende Nachricht:

```
Receive: (see 1979 ((flag c t) 60.3 -19)
((flag l t) 107.8 -36) ((flag r t) 35.2 40)
((flag p l t) 87.4 -41) ((flag p r t) 24.8 3)
((ball) 8.2 17 -0.164 5.4) ((player) 81.5 -42)
((player) 60.3 -44) ((player GELB) 36.6 -17)
((player ROT 2) 10 -18 0 0)
((player ROT 3) 22.2 -24 0 0)
((player GELB 11) 14.9 -23 -0.298 0.2)
((player) 73.7 -38) ((line t) 42.5 -55))
```

Er kann daraus unter anderem folgende Informationen entnehmen:

Die linke obere Eckfahne ('flag l t') ist in 107,8 m Entfernung im Winkel von -36 Grad zu sehen. Die erste Zahl hinter einem Objekt bezeichnet stets die Entfernung, die zweite Zahl den Winkel im Uhrzeigersinn (!) relativ zur eigenen Blickrichtung. Der Ball befindet sich 8,2 m vor ihm im Winkel von 17 Grad. Er befindet sich in Bewegung und wäre 100 Millisekunden später in einer um 0,164 m geringeren Distanz und einem um 5,4 Grad vergrößerten Winkel zu sehen. Die dritte und vierte Zahl geben bei bewegten Objekten an, wie sich Entfernung und Winkel ändern (dabei wird auch die Eigenbewegung des Spielers berücksichtigt).

Der (gegnerische) gelbe Spieler mit der Nummer 11 ('player GELB 11') ist 14,9 m entfernt und kommt näher. Bei weiter entfernten Spielern fehlen die Angaben zur Bewegung. Sie sind auch nicht mehr genau zu identifizieren (es fehlt die Spielernummer und später auch die Zuordnung zu einer Mannschaft).

Der Torwart entschließt sich dann, den Ball abzufangen. Dazu dreht er sich nach rechts um 54,724716 Grad und läuft mit maximaler Kraft ('dash 100.000000'). Er sendet dazu folgende Nachrichten im Abstand von jeweils 100 Millisekunden an den Soccerserver:

```
Send: (turn 54.724716)
Send: (dash 100.000000)
Send: (dash 100.000000)
```

Nach Ausführung dieser Aktionen hat sich sein Blickfeld geändert, und er erhält nach 300 Millisekunden die folgende Nachricht vom SoccerServer:

```
Receive: (see 1982 ((flag r t) 34.5 -10)
((ball) 6.7 -3 -0.402 5.4) ((line r) 12.3 -15))
```

Die Spieler sind jetzt außerhalb seines Blickfeldes. Die Entfernung zum Ball ist geringer geworden, und der Ball kommt noch weiter auf ihn zu (Entfernungswänderung -0,402 m).

Er korrigiert seine Laufrichtung und läuft weiter:

```
Send: (turn 25.853629)
Send: (dash 100.000000)
Send: (dash 100.000000)
```

Da er sich aus der Bewegung heraus drehen wollte, ist die Drehung nur unvollkommen gelungen (vergleiche die Winkelangaben zur Eckfahne 'flag r t'):

```
Receive: (see 1986 ((flag r t) 30.6 -26)
((ball) 3 -3 0.12 5.8) ((line r) 4.3 -30))
```

Er versucht, den Ball noch zu erreichen und wegzuschlagen:

```
Send: (dash 100.000000)
Send: (dash 100.000000)
Send: (kick 100.000000, -134.373864)
```

Jedoch vergeblich:

```
Receive: (hear 1989 referee goal_1_4)
```

der Ball ist schon im Tor.

Handlungsmöglichkeiten und Herausforderung

Zu den Handlungsmöglichkeiten des Spielers gehören die ebenfalls in relativen Angaben formulierten Kommandos, die direkt ins Spielgeschehen eingreifen, wie Turn, Dash, Kick und Say, sowie das Kommando ChangeView, mit dem bestimmt werden kann, welchen Sichtsektor der Spieler und welche Qualität die Sichtinformation hat. Diese Einstellung wirkt sich auf die Frequenz der Sichtinformation aus. Bei der von uns verwendeten Einstellung hat jeder Spieler einen Sichtsektor von 90 Grad und eine hohe Qualität; daraus resultiert ein Update-Intervall von 300 ms. Kommandos werden alle 100 ms vom SoccerServer akzeptiert. Diese getaktete Abarbeitung in festen Zeitintervallen macht den Echtzeitcharakter des Simulationssystems aus.

Folgende Grundprinzipien machen die beschriebene Umgebung so interessant:

- **Eingeschränkte Sensorik:** Wie beschrieben nimmt die Qualität der Informationen mit der Entfernung ab und ist außerdem durch den Sichtsektor eingeschränkt.
- **Ungenauere Kommandos:** Die Ausführung der abgesetzten Befehle ist verrauscht, so daß sich der Spieleragent nicht auf die erwartete Ausführung verlassen kann. Objekte bewegen sich zwar mit einer berechenbaren Geschwindigkeit und werden mit einem bekannten Faktor abgebremst, aber der SoccerServer rechnet mehrere zufällige Fehlerwerte in die Bewegung und die

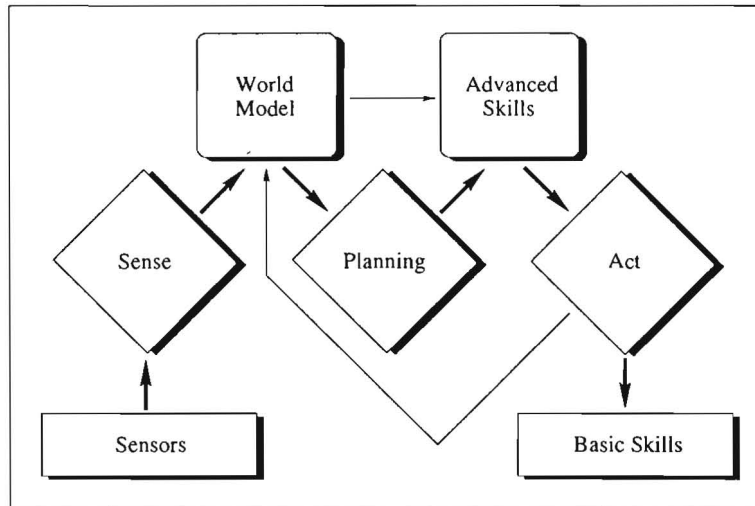


FIGURE 6.2. Aufbau des Agenten, schwarze Pfeile kennzeichnen den zentralen Steuer- und Datenfluß

Kommandoausführung ein, deren Betrag auch von der Stärke bzw. vom Winkel des abgesetzten Kommandos abhängt. Der Einfluß von Wind kann ebenfalls vom SoccerServer einbezogen werden, beim RoboCup97 wurde jedoch darauf verzichtet.

- **Eingeschränkte Kommunikation:** Mit dem Say-Kommando kann ein Spieler nur nach dem Broadcast-Prinzip Nachrichten an andere Spieler in einem bestimmten Umkreis schicken. Die gesendeten Mitteilungen können dafür beliebig kodiert sein.
- **Begrenzte Ressourcen:** Jeder Spieler besitzt nur ein gewisses Maß an Kondition. Durch Aktionen, wie Dash und Kick, wird je nach Stärke dieser Konditionswert verringert. Die Kondition baut sich nur dann langsam wieder auf, wenn der Spieler auf kraftraubende Aktionen verzichtet. Die Konditionsgrenze ist bei der gegenwärtigen offiziellen Einstellung bereits nach einem Lauf über die Spielfeldhälfte mit maximaler Geschwindigkeit erreicht.
- **Autonomie:** Durch das Client-Server-Prinzip des Simulators wird erzwungen, daß für jeden Spieler auch ein eigener Prozeß gestartet wird. Ein steuernder zentraler Agent und Interprozeßkommunikation sind nicht erlaubt. Ein Eingriff der menschlichen Beobachter bei der Simulation ist nur beim Starten des Spiels und bei der Beurteilung von unfairem Spiel oder Deadlocks nötig. Abseits und Handspiel gibt es nicht. Alle anderen typischen Fußballregeln werden vom SoccerServer beachtet und reguliert.

6.3.2 DIE IMPLEMENTIERUNG UNSERER MANNSCHAFT

Für jeden Spieler wird im Wettkampf ein autonomes Programm gestartet. Vom Aufbau sind die Programme gleich, anhand der zugewiesenen Spieler-Nummer erkennt das Programm aber die Rolle (Torwart, Verteidiger, Stürmer, ...), in der es agieren soll, und es verhält sich im Spiel dementsprechend unterschiedlich. Das Programm des Spieleragenten wurde mit C++ in einer Unix-Umgebung realisiert. Abbildung 6.2 erläutert den Hauptinformationsfluß im Agenten.

Sensorik und Aktorik

Die Komponente Sensoren parst und transformiert die in Textzeilen kodierten Sensorinformationen, die der Spieler vom Server bekommt. Die Komponente Elementare Fähigkeiten stellt dem Spieler Methoden zum Absenden der Befehle an den Server zur Verfügung. Die Komponenten Sensorik und Aktorik laufen quasi-parallel zu der Hauptroutine des Agenten, um den Reasoning-Vorgang nicht zu blockieren (Darstellung paralleler Abläufe in Abbildung 6.2 in Rautenform). Realisiert wurde die Parallelität durch UNIX-Systemsignale. Diese Technik erleichtert außerdem die Konsistenz der

eigenen Zeitvorstellung des Agenten und die Synchronität mit dem SoccerServer, da der Signaltakt sehr genau ist.

Weltmodell

In einer dynamischen und unsicheren Umgebung wie dem virtuellen Fußball kommt es darauf an, eine konsistente Vorstellung über die Welt zu modellieren. Kurzfristige Falschinformationen müssen *glattgebügelt*, ungenaue Informationen abgestimmt und fehlende Informationen implizit ermittelt werden. Im Idealfall führt dies zu einer Stabilisierung der Vorstellung des Agenten. Damit die relativen Angaben in Bezug zur eigenen Situation gesetzt werden können, muß zunächst die eigene absolute Position auf dem Spielfeld berechnet werden. Dazu werden Sichtinformationen über Linien, Flaggen und Tore benutzt und unter Beachtung aller möglichen Fälle trianguliert. Die eigene Geschwindigkeit kann aus den abgesetzten Kommandos abgeschätzt werden. Auch über die eigene Kondition wird Buch geführt. Sobald die eigene absolute Position bekannt ist, können auch die absoluten Angaben über die gesehenen Objekte errechnet werden. Das Weltmodell kann durch die Geschwindigkeitsangaben die zeitlichen Lücken zwischen den Sensorinformationen schließen und zukünftige Situationen simulieren. Die Simulation wird von den Komponenten Planung und Höhere Fähigkeiten genutzt, um die möglichen Konsequenzen von Kommandos durchzuspielen.

Planung

Die Komponente Planung realisiert einen Planungs- und Entscheidungsprozeß, der mehrstufig von groben Plänen zu konkreten Kommandofolgen führt. Trifft eine neue Sensorinformation ein, so wird ein neuer Planungsprozeß initiiert. Wenn sich der Ball in der Reichweite des Spielers befindet, kann dieser abspielen oder dribbeln. Hat der Spieleragent den Ball nicht unter Kontrolle, kann er sich dafür entscheiden, den Ball abzufangen, sich umzusehen oder eine bestimmte Position aufzusuchen. Mittels eines Entscheidungsbaums entscheidet sich der Spieler für ein solches Ziel, und er muß dann die beste Möglichkeit finden, dieses Ziel umzusetzen. Diese Phase des Planungsprozesses führt zu einem groben Langzeitplan mit einigen Parametern, z. B. der Abfangposition oder der Abspielrichtung. Die Konkretisierung dieses Plans wird den höheren Fähigkeiten überlassen, die weiter unten beschrieben werden. Kooperation zwischen den Teampartnern entsteht dadurch, daß sich der Spieler auf die Verhaltensweisen seiner Mitspieler verläßt, also ihr Verhalten modelliert. Er erwartet z. B., daß der nächste Spieler seines Teams zum Ball laufen wird, wenn er den Ball abspielt. Auf diese Weise kann darauf verzichtet werden, den Ball direkt auf einen Mitspieler zu spielen, und die Weitergabe wird beschleunigt.

Höhere Fähigkeiten

Die Komponente Höhere Fähigkeiten ist eine Klasse von Fähigkeiten, die es dem Spieler ermöglichen, den Ball effizient zu handhaben und sich optimal zu bewegen. Die technische Aufgabe der höheren Fähigkeiten ist es, die parametrisierten Langzeitpläne der Komponente Planung in Kurzzeitpläne, also konkrete Kommandofolgen, aufzubrechen. Bisher sind folgende Methoden implementiert: Kick (führt mehrere elementare kick-Schritte aus zum SZurechtlegen und effizienten Treten des Balles), Go to Position (drehen und laufen zum Erreichen einer gegebenen Position auf dem Spielfeld), Dribbeln (kicken und laufen ohne den Ball aus der Kontrolle zu verlieren) und Ausweichen vor Hindernissen (drehen und laufen).

6.4 BDI: Theorie unserer Architektur

Die oben behandelte Realisierung des Planungsprozesses kann theoretisch mit der in [1] auf die KI angewandten Belief-Desire-Intention-Architektur von Bratman ([2]) gefaßt werden. Rao und Georgeff führen dabei typische Charakteristika von Problemklassen an, die mit BDI erfolgreich bearbeitet werden können. Sie treffen voll auf die Fußballdomäne zu:

- Der SoccerServer und die Gegner sorgen für eine nicht-deterministische Umgebung.

- Der Agent selbst reagiert nicht-deterministisch, da Teile der Planung randomisiert sind.
- Der Spieleragent kann verschiedene Ziele gleichzeitig verfolgen, z. B. den Ball erreichen und dabei einen Gegner abdecken.
- Der Erfolg der eigenen Aktionen hängt stark von der simulierten Umwelt und den Gegnern ab.
- Jegliche Information ist lokal und für jeden Spieler etwas anders.
- Die Umgebung forciert bounded rationality, da zu tiefes Reasoning keinen Nutzen in dieser dynamischen Umgebung hat.

Dem Begriff *Belief* ist unsere Modellierung der Umgebung durch das Weltmodell gleichzusetzen. Die strikte Relativität der Angaben des SoccerServers führt in jedem Agenten zu einer individuellen Vorstellung von der Welt. Das Konstrukt *Desire* wird in unserer Implementierung durch Ziele aus einer Zielbibliothek erfüllt. In der gegenwärtigen Realisierung können zwar unterschiedliche, z. T. sogar konkurrierende Wünsche existieren, der Agent wählt jedoch immer nur exakt ein Ziel, daß dann auch umgesetzt wird. Die Selektionsfunktion zwischen den Zielen muß schnell sein, damit während dieses Vorgangs keine entscheidungsrelevanten Änderungen eintreten können. Wir standen vor dem gleichen Trade-off, wie Rao und Georgeff ihn beschreiben. Während man bei vollständiger Ausführung eines Langzeitplans riskiert, nicht mehr angemessen auf das Eintreten eines unvorhergesehenen Geschehnisses reagieren zu können, ist die ständige Überprüfung jedes Planungsschrittes zu teuer. Die führt zu den sogenannten Intentions, die in unserem konkreten Fall in zwei Stufen der Planung zerlegt sind. Zunächst wird die beste Möglichkeit, das aktuelle Ziel zu erreichen, ausgewählt und als Absicht gefaßt. Dies entspricht den teilparametrisierten Langzeitplänen. Die tatsächliche Umsetzung der Absicht wird dann in kleinere Abschnitte aufgespalten, die als Kurzzeitpläne in den Höheren Fähigkeiten realisiert sind. Die Persistenz unserer Zielverfolgung beruht nun darauf, daß selbst bei einer Neuintiierung des Planungsprozesse meist die gleichen Schritte genommen und damit die gleichen Ziele und Absichten erzeugt werden. Dazu muß der Spieler verhindern, daß nur geringfügig bessere andere Absichten das bisher geplante Verhalten zunichte machen, z. B. verzichtet der Agent auf kleinere Drehungen auf dem Weg zum Ball, die nur Zeit kosten würden. Erst wenn sich die Situation zu radikal geändert hat, z. B. durch den Verlust des Ballbesitzes, wird der Planungsprozeß einen völlig neuen Weg nehmen.

6.4.1 GRUNDLAGEN FÜR DEN ERFOLG UNSERER MANNSCHAFT

Als Grundlagen für den Erfolg unserer Mannschaft können wir mehrere Faktoren anführen:

- Die Integration von Verfahren aus mehreren Bereichen der Mathematik und der Informatik: Allein für die Positionsbestimmung sind einige hundert Fallunterscheidungen notwendig. Die Schnelligkeit bei der Behandlung und Weiterleitung des Balles beruht ebenfalls auf mathematischer Modellierung. Aus der Informatik spielen neben den Verfahren der KI vor allem die Beherrschung softwaretechnologischer Fragen im Bereich der objekt-orientierten Programmierung sowie der Kommunikation und Synchronisation von Prozessen eine wichtige Rolle.
- Die Anwendung agenten-orientierter Techniken: Sie ermöglichten eine klare und transparente Architektur der Programme. Trotz der relativ kurzen Entwicklungszeit und der Beteiligung unterschiedlicher Entwickler waren noch kurzfristige Verbesserungen und taktische Modifikationen möglich.
- Das offensive und kooperative Spiel der Mannschaft: Das Zusammenwirken der Spieler erfolgte (noch) ohne Kommunikation, allein durch Beobachtung und Modellieren des erwarteten Verhaltens der Mitspieler bei der Entscheidungsfindung. Das Spiel der Mannschaft zeigte die hohe Wirksamkeit bereits solcher einfacher Konzepte aus dem Bereich der Multi-Agenten-Systeme.
- Der Einsatz taktischer Varianten im Wettbewerb: Die Beobachtung unseres Finalgegners bewog uns zu einer (geheim gehaltenen) Umstellung in der Verteidigung, die sowohl die Aufstellung als auch das Spielverhalten betraf. Dazu waren Änderungen am Programm noch zwei Stunden vor dem Finale erforderlich.

- Die ausgezeichneten Fähigkeiten und die Motivation der beteiligten Studenten: Die Integration der unterschiedlichen Verfahren war nur durch entsprechend befähigte Studenten möglich. Insbesondere die erfolgreiche Weiterentwicklung und Modifikation der Programme unter Wettkampfbedingungen kann auch als herausragende sportliche Leistung gesehen werden.

Eine Erfahrung scheint uns wichtig zu sein: Reine KI allein reicht nicht aus. Wir hatten während der Entwicklung zuweilen die Situation, daß eine aus KI-Sicht verbesserte Variante plötzlich schlechtere Ergebnisse brachte. Die Ursache lag in Fehlern an der Basis, die durch das empfindliche KI-Verfahren in ihrer Auswirkung verstärkt wurden. In diesem Sinn erwarten wir Erfolg von lernenden Verfahren nur dann, wenn bereits eine bestimmte vorbereitende Modellierung erfolgt ist.

6.4.2 WEITERE VORHABEN

Die nächsten Schritte zur Verbesserung unserer Spieler betreffen unterschiedliche Formen des Trainierens und Lernens sowie weitere Methoden für das kooperative Spiel:

- Einzelne Entscheidungen und Fähigkeiten benutzen individuell festgelegte Parameter, deren Werte wir bisher durch Tests evaluierten. Diese Parameter sollen maschinell angepaßt werden.
- Mit Methoden des Fallbasierten Schließens können frühere *gute* Aktionen in ähnlichen Situationen wieder nutzbringend eingesetzt werden.
- Unter Lernen verstehen wir dabei sowohl das Trainieren von Verhaltensweisen beim *off-line-learning* als auch die Anpassung an das Verhalten des Gegners während des Spiels (*on-line-learning*).
- Das Teamverhalten unserer Mannschaft soll erweitert werden durch spezielle Verhaltensweisen in *Standard-Situationen*, explizite kooperative Fähigkeiten und Ausnutzung der bereitgestellten Kommunikationsmöglichkeiten auf dem Spielfeld.

6.5 Die Organisation des RoboCup

Die Organisation des RoboCup liegt in der Hand der RoboCup Federation. Sie trifft zum Beispiel die Entscheidungen über die Regeln bei den Wettbewerben und legt die Austragungsorte der Weltmeisterschaften fest. Die nächste Weltmeisterschaft findet vom 2.-8.7.98 (während der FIFA-Weltmeisterschaften) in Paris im Rahmen der Konferenzserie *Agents' World* statt.

Die RoboCup Federation arbeitet mit den Nationalen Komitees zusammen, derartige Komitees existieren gegenwärtig in Japan, USA, Frankreich, Italien und Skandinavien. Die Gründung eines deutschen Nationalkomitees wird im Zusammenhang mit der Gründung des Arbeitskreises *RoboCup* im Fachausschuß 1.4/4.3 *Robotik* vorbereitet.

Für die Mitarbeit in diesem Arbeitskreis sind alle Interessierten eingeladen. Unter anderem ist ein Workshop im Frühjahr in Ulm geplant. Aufgrund unseres Eindrucks von den Wettkämpfen in Nagoya können wir nur dazu ermutigen, sich mit eigenen Teams zu beteiligen: Noch läßt der allgemeine Entwicklungsstand auch Neueinsteigern eine reale Chance, sowohl bei den realen Robotern als auch in der Simulationsliga. Damit nicht jeder von vorn anfangen muß, haben wir die Quellen unseres Programmes frei verfügbar gemacht (wir haben allerdings noch nicht die Kapazitäten gehabt, die Programme gezielt aufzubereiten und bitten dafür um Verständnis).

6.6 Bedeutung des RoboCup

Die Bedeutung des RoboCup sehen wir darin, daß er Langzeitziele setzt, die die Entwicklung vielfältiger Techniken, Verfahren und Theorien sowohl bei der Hardware als auch bei der Software stimulieren kann. Für diese Beförderung sorgt sicher auch das öffentliche Interesse, wobei die Wettbewerbe

zugleich eine Evaluierung des Erreichten ermöglichen, also eine Art von Benchmark-Tests liefern. Dabei wird nur der erfolgreich sein können, der es versteht, unterschiedliche Ansätze erfolgreich zu integrieren.

Der Wettbewerb ist zugleich ein Motor für die Kreativität und für den Austausch der Ideen. Dabei soll noch einmal betont werden, daß die Meisterschaften in Nagoya auch eine sehr lustige Angelegenheit waren, die allen Beteiligten viel Spaß bereitet haben.

Spaß und Kreativität liegen nahe beisammen, und nicht zuletzt deswegen eignet sich der RoboCup auch sehr gut für die Lehre. Unterschiedliche Techniken lassen sich erklären und in studentischen Praktika üben. Ohne weitere Erklärung wird evident, daß gute Systeme nur durch Integration unterschiedlicher Verfahren entstehen können (und daß KI dabei immer nur ein Teil ist, der allerdings bei einer guten Basis den entscheidenden Vorteil bringen kann).

Der RoboCup ist in das Bewußtsein der Öffentlichkeit gedrungen, und wir werden *ob wir es wollen oder nicht* an unseren Erfolgen gemessen werden. Das gibt uns aber auch die Möglichkeit, zu zeigen, wo wir stehen, wobei wir uns nicht zu verstecken brauchen. Das Erfreuliche dabei ist, daß die zu entwickelnden Techniken und Verfahren uns auch praktisch bei der Entwicklung intelligenter Systeme weiterhelfen können.

6.7 Web-Seiten mit Informationen

Auf den folgenden Web-Seiten gibt es Informationen:

- <http://www.RoboCup.org/RoboCup/RoboCup.html>
- <http://www.ki.informatik.hu-berlin.de/RoboCup97/>

Die Programme SoccerServer und SoccerMonitor gibt es unter:

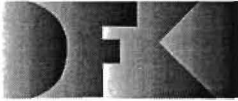
- <http://ci.etl.go.jp/~noda/soccer/server.html>

Dazu gehört ein einfaches Spielerprogramm. Weitere Beispiele für Spielerprogramme findet man auf den genannten Web-Seiten. Den logplayer zum Betrachten von Spielen bekommt man zusammen mit dem SoccerServer und SoccerMonitor. Die log-files der Weltmeisterschaftsspiele stehen unter:

- <ftp://ci.etl.go.jp/pub/soccer/log/RoboCup97/>

6.8 References

- [1] A.S. Rao and M.P. Georgeff. *BDI Agents: From Theory to Practice*. V. Lesser (Ed.) Proc. of the First Int. Conf. on Multi-Agent Systems (ICMAS-95). MIT-Press, 1995, 312-319
- [2] M.E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.
- [3] Hiroaki Kitano, Milind Tambe, Peter Stone, Manuela Veloso, Silvia Coradeschi, Eiichi Osawa, Hitoshi Matsubara, Itsuki Noda, Minoru Asada. *The RoboCup Synthetic Agent Challenge 97*. M.E. Pollack (Ed.), Proc. IJCAI-97. Morgan Kaufmann, 1997, 24-29.



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

-Bibliothek, Information
und Dokumentation (BID)-
PF 2080
67608 Kaiserslautern
FRG

Telefon (0631) 205-3506
Telefax (0631) 205-3210
e-mail
dfkibib@dfki.uni-kl.de
WWW
http://www.dfki.uni-
sb.de/dfkibib

Veröffentlichungen des DFKI

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder (so sie als per ftp erhaeltlich angemerkt sind) per anonymous ftp von ftp.dfki.uni-kl.de (131.246.241.100) im Verzeichnis pub/Publications bezogen werden. Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or (if they are marked as obtainable by ftp) by anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) in the directory pub/Publications.

The reports are distributed free of charge except where otherwise noted.

DFKI Research Reports

1997

RR-97-08

Stefan Müller

Complement Extraction Lexical Rules and Argument
Attraction
14 pages

RR-97-07

Stefan Müller

Yet Another Paper about Partial Verb Phrase Fronting
in German
26 pages

RR-97-06

Stefan Müller

Scrambling in German – Extraction into the *Mittelfeld*
24 pages

RR-97-05

Harald Meyer auf'm Hofe

Finding Regions of Local Repair in Hierarchical Con-
straint Satisfaction
33 pages

RR-97-04

Serge Autexier, Dieter Hutter

Parameterized Abstractions used for Proof-Planning
13 pages

RR-97-03

Dieter Hutter

Using Rippling to Prove the Termination of Algorithms
15 pages

RR-97-02

*Stephan Busemann, Thierry Declerck, Abdel Kader
Diagne, Luca Dini,
Judith Klein, Sven Schmeier*

Natural Language Dialogue Service for Appointment
Scheduling Agents
15 pages

RR-97-01

Erica Melis, Claus Sengler

Analogy in Verification of State-Based Specifications:
First Results
12 pages

1996

RR-96-06

Claus Sengler

Case Studies of Non-Freely Generated Data Types
200 pages

RR-96-05

Stephan Busemann

Best-First Surface Realization
11 pages

RR-96-04

Christoph G. Jung, Klaus Fischer, Alastair Burt
Multi-Agent Planning
Using an *Abductive*
EVENT CALCULUS
114 pages

RR-96-03

Günter Neumann
Interleaving
Natural Language Parsing and Generation
Through Uniform Processing
51 pages

RR-96-02

E.André, J. Müller, T.Rist:
PPP-Persona: Ein objektorientierter Multimedia-Prä-
sentationsagent
14 Seiten

RR-96-01

Claus Sengler
Induction on Non-Freely Generated Data Types
188 pages

1995**RR-95-20**

Hans-Ulrich Krieger
Typed Feature Structures, Definite Equivalences,
Greatest Model Semantics, and Nonmonotonicity
27 pages

RR-95-19

*Abdel Kader Diagne, Walter Kasper, Hans-Ulrich Krie-
ger*
Distributed Parsing With HPSG Grammar
20 pages

RR-95-18

Hans-Ulrich Krieger, Ulrich Schäfer
Efficient Parameterizable Type Expansion for Typed
Feature Formalisms
19 pages

RR-95-17

Hans-Ulrich Krieger
Classification and Representation of Types in TDL
17 pages

RR-95-16

Martin Müller, Tobias Van Roy
Title not set
0 pages

Note: The author(s) were unable to deliver this docu-
ment for printing before the end of the year. It
will be printed next year.

RR-95-15

Joachim Niehren, Tobias Van Roy
Title not set
0 pages

Note: The author(s) were unable to deliver this docu-
ment for printing before the end of the year. It
will be printed next year.

RR-95-14

Joachim Niehren
Functional Computation as Concurrent Computation
50 pages

RR-95-13

Werner Stephan, Susanne Biundo
Deduction-based Refinement Planning
14 pages

RR-95-12

Walter Hower, Winfried H. Graf
Research in Constraint-Based Layout, Visualization,
CAD, and Related Topics: A Bibliographical Survey
33 pages

RR-95-11

Anne Kilger, Wolfgang Finkler
Incremental Generation for Real-Time Applications
47 pages

RR-95-10

Gert Smolka
The Oz Programming Model
23 pages

RR-95-09

M. Buchheit, F. M. Donini, W. Nutt, A. Schaerf
A Refined Architecture for Terminological Systems:
Terminology = Schema + Views
71 pages

RR-95-08

Michael Mehl, Ralf Scheidhauer, Christian Schulte
An Abstract Machine for Oz
23 pages

RR-95-07

*Francesco M. Donini, Maurizio Lenzerini, Daniele Nar-
di, Werner Nutt*
The Complexity of Concept Languages
57 pages

RR-95-06

Bernd Kiefer, Thomas Fettig
FEGRAMED
An interactive Graphics Editor for Feature Structures
37 pages

RR-95-05

Rolf Backofen, James Rogers, K. Vijay-Shanker
A First-Order Axiomatization of the Theory of Finite
Trees
35 pages

RR-95-04

M. Buchheit, H.-J. Bürckert, B. Hollunder, A. Laux, W. Nutt, M. Wójcik
Task Acquisition with a Description Logic Reasoner
17 pages

RR-95-03

Stephan Baumann, Michael Malburg, Hans-Guenther Hein, Rainer Hoch, Thomas Kieninger, Norbert Kuhn
Document Analysis at DFKI
Part 2: Information Extraction
40 pages

RR-95-02

Majdi Ben Hadj Ali, Frank Fein, Frank Hoenes, Thorsten Jaeger, Achim Weigel
Document Analysis at DFKI
Part 1: Image Analysis and Text Recognition
69 pages

RR-95-01

Klaus Fischer, Jörg P. Müller, Markus Pischel
Cooperative Transportation Scheduling
an application Domain for DAI
31 pages

1994**RR-94-39**

Hans-Ulrich Krieger
Typed Feature Formalisms as a Common Basis for Linguistic Specification.
21 pages

RR-94-38

Hans Uszkoreit, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth A. Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus Netter, Günter Neumann, Stephan Oepen, Stephen P. Spackman.
DISCO—An HPSG-based NLP System and its Application for Appointment Scheduling.
13 pages

RR-94-37

Hans-Ulrich Krieger, Ulrich Schäfer
TDL - A Type Description Language for HPSG, Part 1: Overview.
54 pages

RR-94-36

Manfred Meyer
Issues in Concurrent Knowledge Engineering. Knowledge Base and Knowledge Share Evolution.
17 pages

RR-94-35

Rolf Backofen
A Complete Axiomatization of a Theory with Feature and Arity Constraints
49 pages

RR-94-34

Stephan Busemann, Stephan Oepen, Elizabeth A. Hinkelman, Günter Neumann, Hans Uszkoreit
COSMA – Multi-Participant NL Interaction for Appointment Scheduling
80 pages

RR-94-33

Franz Baader, Armin Laux
Terminological Logics with Modal Operators
29 pages

RR-94-31

Otto Kühn, Volker Becker, Georg Lohse, Philipp Neumann
Integrated Knowledge Utilization and Evolution for the Conservation of Corporate Know-How
17 pages

RR-94-23

Gert Smolka
The Definition of Kernel Oz
53 pages

RR-94-20

Christian Schulte, Gert Smolka, Jörg Würtz
Encapsulated Search and Constraint Programming in Oz
21 pages

RR-94-19

Rainer Hoch
Using IR Techniques for Text Classification in Document Analysis
16 pages

RR-94-18

Rolf Backofen, Ralf Treinen
How to Win a Game with Features
18 pages

RR-94-17

Georg Struth
Philosophical Logics—A Survey and a Bibliography
58 pages

RR-94-16

Gert Smolka
A Foundation for Higher-order Concurrent Constraint Programming
26 pages

- RR-94-15**
Winfried H. Graf, Stefan Neurohr
 Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
 20 pages
- RR-94-14**
Harold Boley, Ulrich Buhrmann, Christof Kremer
 Towards a Sharable Knowledge Base on Recyclable Plastics
 14 pages
- RR-94-13**
Jana Koehler
 Planning from Second Principles—A Logic-based Approach
 49 pages
- RR-94-12**
Hubert Comon, Ralf Treinen
 Ordering Constraints on Trees
 34 pages
- RR-94-11**
Knut Hinkelmann
 A Consequence Finding Approach for Feature Recognition in CAPP
 18 pages
- RR-94-10**
Knut Hinkelmann, Helge Hintze
 Computing Cost Estimates for Proof Strategies
 22 pages
- RR-94-08**
Otto Kühn, Björn Höfling
 Conserving Corporate Knowledge for Crankshaft Design
 17 pages
- RR-94-07**
Harold Boley
 Finite Domains and Exclusions as First-Class Citizens
 25 pages
- RR-94-06**
Dietmar Dengler
 An Adaptive Deductive Planning System
 17 pages
- RR-94-05**
Franz Schmalhofer, J. Stuart Aitken, Lyle E. Bourne jr.
 Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
 81 pages
- RR-94-03**
Gert Smolka
 A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
 34 pages
- RR-94-02**
Elisabeth André, Thomas Rist
 Von Textgeneratoren zu Intellimedia-Präsentationssystemen
 22 Seiten
- RR-94-01**
Elisabeth André, Thomas Rist
 Multimedia Presentations: The Support of Passive and Active Viewing
 15 pages

DFKI Technical Memos

1996

1997

- TM-97-03**
Hans-Jürgen Bürckert, Klaus Fischer, Gero Vierke
 TeleTruck: A Holonic Fleet Management System
 10 pages
- TM-97-02**
Christian Gerber
 Scalability of Multi-Agent Systems - Proposal for a Dissertation
 49 pages
- TM-97-01**
Markus Perling
 GeneTS: A Relational-Functional Genetic Algorithm for the Traveling Salesman Problem
 26 pages

- TM-96-02**
Harold Boley
 Knowledge Bases in the World Wide Web: A Challenge for Logic Programming (Second, Revised Edition)
 10 pages
- TM-96-01**
Gerd Kamp, Holger Wache
 CTL — a description Logic with expressive concrete domains
 19 pages

1995

TM-95-04

Klaus Schmid
Creative Problem Solving
and
Automated Discovery
— An Analysis of Psychological and AI Research —
152 pages

TM-95-03

Andreas Abecker, Harold Boley, Knut Hinkelmann, Holger Wache, Franz Schmalhofer
An Environment for Exploring and Validating Declarative Knowledge
11 pages

TM-95-02

Michael Sintek
FLIP: Functional-plus-Logic Programming
on an Integrated Platform
106 pages

TM-95-01

Martin Buchheit, Rüdiger Klein, Werner Nutt
Constructive Problem Solving: A Model Construction
Approach towards Configuration
34 pages

1994

TM-94-05

Klaus Fischer, Jörg P. Müller, Markus Fischel
Unifying Control in a Layered Agent Architecture
27 pages

TM-94-04

Cornelia Fischer
PAntUDE – An Anti-Unification Algorithm for Expressing Refined Generalizations
22 pages

TM-94-03

Victoria Hall
Uncertainty-Valued Horn Clauses
31 pages

TM-94-02

Rainer Bleisinger, Berthold Kröll
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

TM-94-01

Rainer Bleisinger, Klaus-Peter Gores
Text Skimming as a Part in Paper Document Understanding
14 pages

DFKI Documents

1997

D-97-08

Christoph G. Jung, Klaus Fischer, Susanne Schacht
Distributed Cognitive Systems
Proceedings of the VKS'97 Workshop
50 pages

D-97-07

Harold Boley, Bernd Bachmann, Christian Blum, Christian Embacher, Andreas Lorenz, Jamel Zakraoui
PIMaS:
Ein objektorientiert-regelbasiertes System zur Produkt-Prozeß-Transformation
45 Seiten

D-97-06

Tilman Becker, Stephan Busemann, Wolfgang Finkler
DFKI Workshop on Natural Language Generation
67 pages

D-97-05

Stephan Baumann, Majdi Ben Hadj Ali, Jürgen Lichter, Michael Malburg, Harald Meyer auf'm Hofe, Claudia Wenzel
Anforderungen an ein System zur Dokumentanalyse im Unternehmenskontext
— Integration von Datenbeständen, Aufbau- und Ablauforganisation
42 Seiten

D-97-04

Claudia Wenzel, Markus Junker
Entwurf einer Patternbeschreibungssprache für die Informationsextraktion in der Dokumentanalyse
24 Seiten

D-97-03

Andreas Abecker, Stefan Decker, Knut Hinkelmann, Ulrich Reimer
Proceedings of the Workshop „Knowledge-Based Systems for Knowledge Management in Enterprises“ 97
167 pages

D-97-02*Tilman Becker, Hans-Ulrich Krieger*

Proceedings of the Fifth Meeting on Mathematics of Language (MOL5)

168 pages

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).**D-97-01***Thomas Malik*

NetGLTool Benutzeranleitung

40 Seiten

1996**D-96-07***Technical Staff*

DFKI Jahresbericht 1995

55 Seiten

Note: This document is no longer available in printed form.**D-96-06***Klaus Fischer (Ed.)*

Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems

63 pages

D-96-05*Martin Schaaf*

Ein Framework zur Erstellung verteilter Anwendungen

94 pages

D-96-04*Franz Baader, Hans-Jürgen Bürckert, Andreas Günter, Werner Nutt (Hrsg.)*

Proceedings of the Workshop on Knowledge Representation and Configuration WRKP'96

83 pages

D-96-03*Winfried Tautges*

Der DESIGN-ANALYZER - Decision Support im Designprozess

75 Seiten

D-96-01*Klaus Fischer, Darius Schier*

Ein Multiagentenansatz zum Lösen von Fleet-Scheduling-Problemen

72 Seiten

1995**D-95-12***F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.)*

Working Notes of the KI'95 Workshop:

KRDB-95 - Reasoning about Structured Objects:

Knowledge Representation Meets Databases

61 pages

D-95-11*Stephan Busemann, Iris Merget*

Eine Untersuchung kommerzieller Terminverwaltungssoftware im Hinblick auf die Kopplung mit natürlichsprachlichen Systemen

32 Seiten

D-95-10*Volker Ehresmann*

Integration ressourcen-orientierter Techniken in das wissensbasierte Konfigurierungssystem TOOCON

108 Seiten

D-95-09*Antonio Krüger*

PROXIMA: Ein System zur Generierung graphischer Abstraktionen

120 Seiten

D-95-08*Technical Staff*

DFKI Jahresbericht 1994

63 Seiten

Note: This document is no longer available in printed form.**D-95-07***Ottmar Lutzy*

Morphic - Plus

Ein morphologisches Analyseprogramm für die deutsche Flexionsmorphologie und Komposita-Analyse

74 Seiten

D-95-06*Markus Steffens, Ansgar Bernardi*

Integriertes Produktmodell für Behälter aus Faserverbundwerkstoffen

48 Seiten

D-95-05*Georg Schneider*

Eine Werkbank zur Erzeugung von 3D-Illustrationen

157 Seiten

D-95-04*Victoria Hall*

Integration von Sorten als ausgezeichnete taxonomische Prädikate in eine relational-funktionale Sprache

56 Seiten

D-95-03*Christoph Endres, Lars Klein, Markus Meyer*

Implementierung und Erweiterung der Sprache ACCP

110 Seiten

D-95-02*Andreas Butz*

BETTY

Ein System zur Planung und Generierung informativer Animationssequenzen

95 Seiten

D-95-01

Susanne Biundo, Wolfgang Tank (Hrsg.)
 PuK-95, Beiträge zum 9. Workshop „Planen und Konfigurieren“, Februar 1995
 169 Seiten

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

1994**D-94-15**

Stephan Oepen
 German Nominal Syntax in HPSG
 — On Syntactic Categories and Syntagmatic Relations —
 80 pages

D-94-14

Hans-Ulrich Krieger, Ulrich Schäfer
 TDL - A Type Description Language for HPSG, Part 2: User Guide.
 72 pages

D-94-12

Arthur Sehn, Serge Autexier (Hrsg.)
 Proceedings des Studentenprogramms der 18. Deutschen Jahrestagung für Künstliche Intelligenz KI-94
 69 Seiten

D-94-11

F. Baader, M. Buchheit, M. A. Jeusfeld, W. Nutt (Eds.)
 Working Notes of the KI'94 Workshop: KRDB'94 - Reasoning about Structured Objects: Knowledge Representation Meets Databases
 65 pages

Note: This document is no longer available in printed form.

D-94-10

F. Baader, M. Lenzerini, W. Nutt, P. F. Patel-Schneider (Eds.)
 Working Notes of the 1994 International Workshop on Description Logics
 118 pages

Note: This document is available for a nominal charge of 25 DM (or 15 US-\$).

D-94-09

Technical Staff
 DFKI Wissenschaftlich-Technischer Jahresbericht 1993
 145 Seiten

D-94-08

Harald Feibel
 IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
 58 Seiten

D-94-07

Claudia Wenzel, Rainer Hoch
 Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
 25 Seiten

Note: This document is no longer available in printed form.

D-94-06

Ulrich Buhrmann
 Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
 117 Seiten

D-94-04

Franz Schmalhofer, Ludger van Elst
 Entwicklung von Expertensystemen: Prototypen, Tiefenmodellierung und kooperative Wissensentwicklung
 22 Seiten

D-94-03

Franz Schmalhofer
 Maschinelles Lernen: Eine kognitionswissenschaftliche Betrachtung
 54 Seiten

Note: This document is no longer available in printed form.

D-94-02

Markus Steffens
 Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
 90 pages

D-94-01

Josua Boon (Ed.)
 DFKI-Publications: The First Four Years 1990 - 1993
 75 pages

**Distributed Cognitive Systems
Proceedings of the VKS'97 Workshop**

Christoph G. Jung, Klaus Fischer, and Susanne Schacht

D-97-08
Document