Resource Management for Boundedly Optimal Agent Societies

Christian Gerber[1] **and Christoph G. Jung**[2]

Abstract. In this paper, we propose a generalization of the bounded rationality paradigm to multi-agent systems (MAS) in which self-interested agents work on a common goal. A *hierarchical resource adaption scheme* is developed that links the *micro-level* of a hybrid and layered agent architecture to the *macro-level* of the corresponding society. The introduction of societal structures provides the key for breaking down the highly complex search for optimal solutions into a multi-staged process. Each of the social stages integrates the complementary benefits of *simple* and *complex decision making* based on the device of *abstract resources*. These are unified representations of quantitative environmental and architectural constraints or interdependencies between subordinate problem-solvers. The presented framework is integrated into an existing multi-agent system and thus applied to a series of applications. To clarify our approach, we refer to a transportation telematics domains.

1 Introduction

During the last decade, a new focus in Artificial Intelligence (AI) has emerged around the notion of *bounded rationality* [14]: intelligence is not anymore viewed as an in-principle capability of solving abstract problems, but is rather being measured through performance with respect to a dynamically changing environment. We consider *resource distribution* to be a key to achieve online adaption of a system to changes in the domain, where *resources* are regarded as mostly quantitative environmental and architectural conditions that constrain the behavior of the *situated agent*.

Still, the question how to generalize theories and their practical implications to the multi-agent case is an issue where a commonly agreed solution has not been found yet. In this paper, we extend the theory of *bounded optimality* [13, 12] to societies of agents that are, at least partially, benevolent to each other and that share a common goal. From this extension we derive a two-fold mechanism to control resources not only on every stage of a layered agent architecture, but also on every social stage in the society. We propose this mechanism as to meet the urgent demand of verifiably robust and *scalable* MAS, i.e., systems that are able to adapt to any problem size. This property is especially relevant for large applications such as in emerging global networks, flexible manufacturing systems (FMS), and transportation telematics.

The mechanism being presented consists of a *simple decision making unit* and of a *complex decision making unit*. *Complex* decisions denote long-term intentions out of primitive system operations whose composition produces an optimal answer. Generally, the corresponding decision procedures turn out to be complex too, in the sense that their amount of computation increases exponentially in problem size. *Simple* decisions are regarded as primitive measures that maximize the system's performance just for the next, single step. Often, they can be computed using myopic procedures that only take the immediate consequences of an action into account.

[1] German Research Center for Artificial Intelligence (DFKI), Im Stadtwald, B36, 66123 Saarbrücken, Germany; Part of this work has been supported by Siemens AG Germany

[2] GK Kogwiss., Univ. des Saarlandes, Im Stadtwald, Bau 36, D-66123 Saarbrücken, Germany

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

Structure of the Paper Throughout the paper, we frequently refer to a specific application, the TELETRUCK system [1], to illustrate our concepts. Hence for the sake of clarity, we briefly introduce the system in Section 2. In Section 3, we present our MAS-extension of the bounded optimality theory. To uphold the practical aspects of a tractable, robust, and scalable control regime, we propose in Section 4 our hierarchical resource management model that connects the *macro-level* of the agent society to the *micro-level* of *layered* agent architectures such as the INTERRAP model [11]. Section 5 then focuses on the representation of abstract resources and the two-fold distribution mechanism that is integrated into each social stage and that incorporates both simple, i.e., locally optimal, and complex, i.e., globally optimal methods for decision making. In Section 6, we discuss an interpretation of our approach under the perspective of *holonic* systems. Finally, we conclude and give an overview over ongoing and future work on realizing the present model.

Related Work Our theoretical and practical considerations draw heavily on the influencing ideas of [13, 12] and generalize the framework to the multi-agent case where self-interested agents work on a common goal. In contrast to, e.g., *social rationality* [6], we introduce explicit social control structures that are nevertheless open to individual decisions. The presented agent architecture bases on the hybrid agent architecture of [11] enhanced by variations of decision procedures found in [16] and [5]. While this paper focuses on the architectural, engineering aspect, [3] discusses the relevance of our approach to cognitive and social sciences.

2 The TELETRUCK Application

The TELETRUCK system [1] has been developed as a multi-agent based fleet scheduling system in which a collection of geographically dispersed shipping companies carry out transportation tasks for various customers. The companies have a fleet of transportation units like drivers, trucks, or trailers at their disposal that can be combined to *means of transportation.* Units of various types may differ in many ways: trucks can be classified into pure tractors, those without loading space, and those without. The type and size of loading space of containers constrains the type of cargo that can be transported. Also human drivers differ in their supplied working time and the type of cargo he or she may transport depending on issues such as special training or certain licenses.

The job of a dispatching system is the assignment of transportation orders to those means of transportation which fulfill best the customers' requirements (e.g., time constraints). For each means of transportation a tour plan has to be generated for the assigned tasks. This plan specifies the order and the time schedule for the execution of the transportation tasks. Hence, the dispatchers are faced with a two-staged planning problem.

In common practice, traditional Operations Research (OR) methods are used to solve this problem. However, the underlying problem specification is not static, since new customer orders or modifications of already scheduled orders may drop in during the execution time. Furthermore, traffic jams or truck breakdowns can lead to the infeasibility of a plan and enforce online re-planning. OR-techniques can hardly cope with the dynamics arising during the execution time of the tour plans since they have to build a new schedule from scratch.

In TELETRUCK, means of transportation are represented through autonomous agents that compute an overall solution in a decentral manner. Furthermore, these agents are able to monitor the execution of their local schedules which they can re-plan locally in case of unforeseen events.

[2] showed that the quality of solutions obtained from a multi-agent implementation for the off-line, medium-scaled scheduling problem (50-100 orders) can compete with OR solutions. However, in addition the TELETRUCK approach is much more flexible, since it allows to vary the number of agents on-line and can cope with open, dynamic scheduling problems and with uncertainty in plan execution.

Two additional factors of the transportation domain go beyond OR techniques: First, there is the required scalability of the applied decision making. Most of the traditional minimum-cost flow algorithms scale proportionally to the number of *all possible* origin and destination locations, no matter if such a location is on schedule at all. In large-scaled real world applications such approaches become easily intractable. In TELETRUCK, such problems can naturally be overcome through structurization and organization of the agents representing the geographically distributed fleet, making the TELETRUCK approach well scalable.

Second, there are also factors of self-interest in the transportation domain. Human drivers' preferences could sometimes conflict with each other or collide with some goal of the shipping company. Taking those wishes into account plays an important role in developing practical fleet schedules. Furthermore, we could allow for shipping companies to cooperate with each other in order to process tasks that no company could process on its own. In each case, there is however some common goal involved which justifies coordination.

In the following, we shall elaborate theoretically and practically our approach of a balanced and tractable decision mechanism for distributing tasks and resources in the multi-agent case.

3 The Boundedly Optimal Agent Society

In [12], Russell and Subramanian define *bounded optimality* as a property defined on a set of agents $\mathcal{L}$ regarded as programs or states. The environment in which such an agent is situated can be represented by a given transition $T : W \times \mathcal{L} \to W \times \mathcal{L}$ of world states W. The boundedly optimal agent $l_{opt} \in \mathcal{L}$ with respect to T solves the constrained optimization problem $l_{opt} = \mathit{argmax}_{l \in \mathcal{L}} V(\lambda w.T(w, l))$. In terms of a utility assignment V, it thus maximizes the outcome of functions $\lambda w.T(w, l)$ that are enumerating the possibly infinite world evolution given an agent program l.[3]

[12] argues that optimal behavior cannot be reached in a domain-independent manner with only some *object-level* oriented decision making procedure. Additionally, the higher-order problem of approximating the optimal agent program has to be solved. This justifies the application of *meta-reasoning* in the proposed, practical architecture of [13]: *Internal* (*architectural*, *computational*) resources are *assigned* to and *monitored* from a set of optional courses (or programs) running on the object-level. Since the meta-reasoning component also applies for internal resources, it should be of neglectable complexity. In contrast to the *complex*, structured set of decisions on the object-level, the meta-level reasoning is a fast and *simple* decision procedure. Simple resource assignment is therefore only locally optimal, i.e., for a single time step. Thus it does not necessarily lead to the global optimum, but hopefully to a satisfying solution.

We are interested in the multi-agent case of bounded optimality: We extend the environment $T : W \times \mathcal{L}^* \to W \times \mathcal{L}^*$ to a simultaneous reduction of all the agent programs. The *boundedly optimal agent society* is then the selected tuple $\vec{l}_{opt} \in \mathcal{L}^*$ as a solution to the constrained optimization problem $\vec{l}_{opt} = \mathit{argmax}_{\vec{l} \in \mathcal{L}^*} \tilde{V}(\lambda w.T(w, \vec{l}))$.

Herein, the performance measure $\tilde{V}$ is an appropriate *game-theoretic* measure that reflects the global performance with regard to the common goal of the involved agents similar to the utility V in the single-agent case. In addition, it has to reflect the stability of trading off individual interests, i.e., the fewer arguments a participant agent might object against a particular corporate solution and the less social importance the agent has, the more stable the solution is. Such aspects can be integrated to V as weighted penalties. It is clear that such a theoretical perspective only makes sense in the presence of a common goal and the existence of stable solutions with respect to individual interests.

A practical multi-agent architecture to approximate the boundedly optimal society, however, does not only have to elaborate *social* macro-level aspects to ensure optimality. It also has to link micro-level agent models, (such as the hybrid model of [11]), that are suitable for both reactive and deliberative behavior. The social architecture that we develop in the following sections copes with complex MAS-typical inter-dependencies in assigning resources and leaves enough space for the autonomous decision making facilities to refine the formulated guidelines.

Relation to the Application Similar to the OR approach, applying a single-agent architecture to the complete transportation problem reveals the lacking scalability of Russell and Wefald's centralized approach. The object-level has to solve the distributed vehicle routing problem while the assignments of tasks to trucks might rather be realized on the meta-level. Both jobs are intractable for real-world problem sizes, let alone the incapability of the system to react to the dynamics of the whole, complex scenario. In addition, the existence of an explicit global utility function that gives reasonable predictions for all shipping companies cannot be assumed.

A simple employment of a meta-reasoning architecture according to [13] to every agents in a multi-agent-based approach is still not

[3] In the original formulation of [12], agent interpreter and environment are independent. Our presentation, however, unifies these into a single transition T which is motivated by our further elaborations. Furthermore, we do not model T under incomplete knowledge, at the moment. These simplifications should not affect the claims of our paper. Especially the replacement of utility with *expected utility* within all the presented concepts slightly complicates the picture, but is a standard technique.

sufficient. In our example domain we could model, for instance, each truck as a separate Russell-and-Wefald agent that independently applies for delivery tasks, plans and executes a tour. However, because the truck agents do only optimize from their local perspective, the quality of the resulting distributed schedule would be rather poor. Additionally, the restricted reactivity of the truck agents prohibits a flexible recoverage from failure. The modeling of shipping companies as agents behaves in a similar sub-optimal fashion and furthermore shares the high complexity of the single-agent case as described above. A more structured meta-reasoning procedure is required.

4 A Hierarchical Model of Resource Control

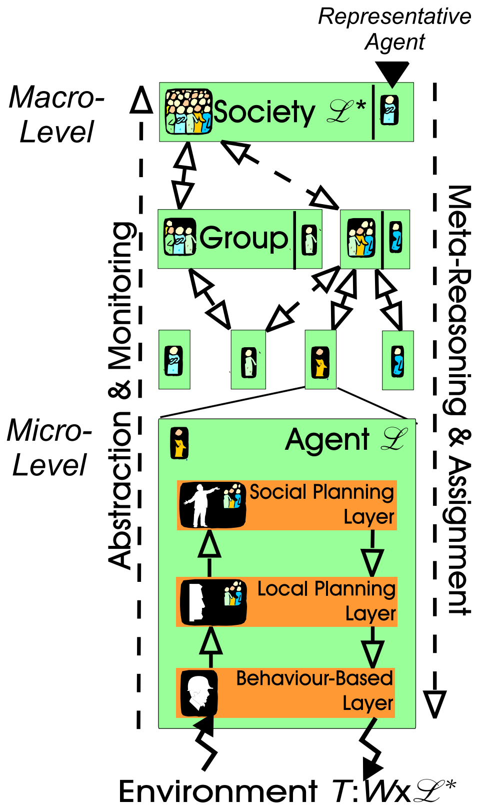

Figure 1. Stages and Decision Procedures in Resource Control

To approximate the boundedly optimal agent society, a straightforward though by no means trivial approach is to introduce a social hierarchy which consists of agent layers, individual agents, groups[4], and the society (Figure 1) as explicit *decision stages*. We follow the three-layered INTERRAP agent architecture of [11] to build up the agent stage. INTERRAP employs reactive, procedural knowledge in the *behavior-based layer* (BBL), deliberative planning in the *local planning layer* (LPL), and finally communication mechanisms in the *social planning layer* (SPL).

The content of any adaption decision on a particular macro-level stage, but also agent-internal stage, is the resource assignment to members of the subordinate stages based on their performances. On

[4] In our scheme, groups consist of arbitrary mixtures of agents and subgroups.

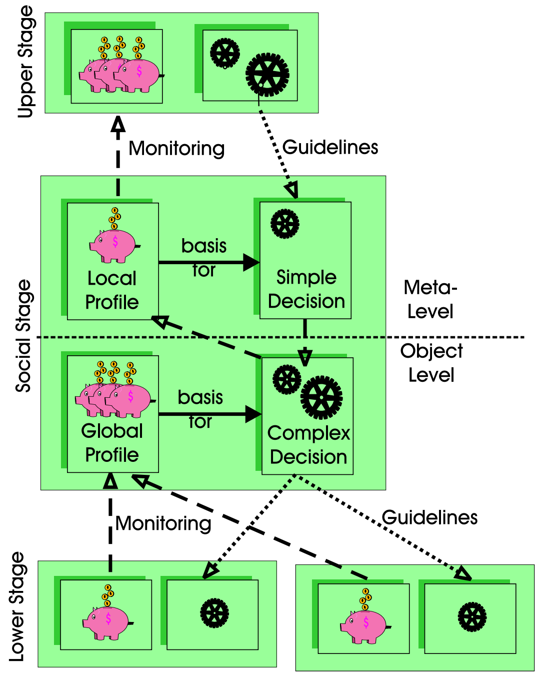

Figure 2. Stages and Decision Procedures in Resource Control

the macro-level stages, several options remain to actually implement the resource assignment procedure: Distributed approaches realize a common social stage purely by communication protocols and the inherent distributed decisions. More centralistic approaches either totally replace the former individual agents by a new agent or they introduce a so-called *representative agent* being a selected or fresh member of the groups.

We choose the latter approach of representative agents because one the one hand, it avoids the communication overhead of a fully distributed setting. Furthermore, it leaves open the possibility of a dynamic reconfiguration which is much harder to realize of one takes the replacement option. Finally, the representative straightforwardly inherits the resource-management facilities from the INTERRAP agent architecture: Similar to the micro-level case, the representative achieves an efficient structure of its social stage through performance monitoring and resource allocation; the same allocation mechanism can hence be employed for both the micro- and macro-level resource distribution. This concept supports the design of social structures that are dynamically created and modified where useful.

[13, 12] distinguish between between internal and external resources, between environmental and architectural constraints. Internal resources only affect the *interpretation* of the agent program, while external ones are the topic of the object-level reasoning. In this paper, we abstract from this distinction for the multi-agent case: the explicit representational device of *abstract resources* (Section 5.1) captures any general interdependency, be it internal or external, between the problem solvers constituting a social stage.

Based on abstract resources, we are able to develop a two-fold allocation procedure to be integrated on *every* (i.e., on each macro-level and each micro-level) stage that monitors the performance of lower stages and sets up guidelines for the still autonomous behavior of the lower stages. As Figure 2 illustrates, we integrate the complementary advantages of a complex and optimal search by a *decision-theoretic planning* approach (see Section 5.2) and of a fast and flexible mechanism, the *steepest ascent* method (see Section 5.3). At each social stage, the complex decision making is directed by the simple decision making as its meta-level. As a result, our social hierarchy is a dynamic trade-off between tractability and optimality which does not stick with the restrictions of Russell and Wefald's original architecture.

Relation to the Application In TELETRUCK the lowest societal stage consists of truck driver, truck tractor, trailer, and cargo units, all of which are represented by agents. A means of transportation (i.e., a feasible combination of trailer, truck and driver) is represented by an additional agent, the so called *Planning'n'Execution Unit (PnEU)*. On a higher stage, the whole company is also agentified. Figure 3 shows the correlations. Higher stages in the society are company associations and inter-modal associations.

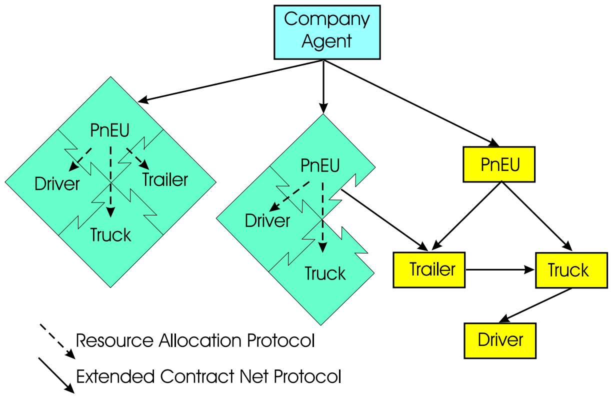

Figure 3. Stages and Decision Procedures in TELETRUCK

A truck agent, for instance, is realized in INTERRAP as follows: The agent executes and monitors the tours of a truck within the BBL. Route planning is done in the LPL while the SPL negotiates about tasks and coordination. A company agent monitors the performance of the company's fleet. If, for instance, all trucks are booked out frequently, the company agent induces the purchase of additional trucks.

The shortcoming of the distinction between external and internal resources can be seen at the following example: allowing two truck agents to plan and execute the same delivery task (wasting a computational resource), eventually leads to conflicting physical actions, such as the actual loading of the cargo (an external resource).

5 Decision Making Based on Abstract Resources

5.1 Abstract Resources and Profiling

As aforementioned, the clear distinction between architectural and environmental constraints does not seem to be reasonable in the multi-agent case: interacting agents affect the group stage in a similar fashion, no matter if this happens internally, e.g., by running on the same computing device, or if this happens externally by performing actions in the physical world. Similarly, controlling the computation of an agent also constrains its access to the outer world.

To represent the interdependencies between a number of problem solvers, we regard an abstract resource as a limited set of items for which these problem solvers apply. The task of each social stage of our hierarchy therefore amounts to decide about the distribution of the items. This happens by determining disjoint subsets of the abstract resource to be allocated to the subordinate problem solvers.

A rather primitive example of an abstract resource is a unary set which corresponds to the well-known construct of a *semaphore* in Computer Science. This restricts certain activities that apply for the semaphore 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). The semaphore also illustrates the higher-level character of resources since it does not restrict the detailed computation within the confirmed activity. Rather, it is a representation of a selected subspace of allocations for the concrete resources on the next deeper stage; assigning an abstract resource thus amounts to putting "guidelines" or constraints on the further refinement on the lower stages.

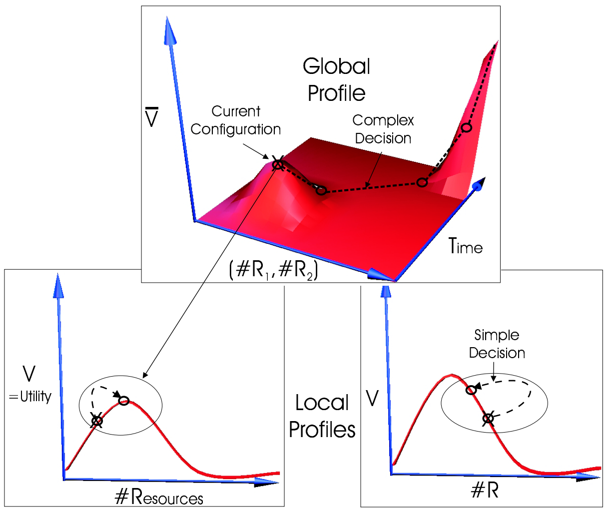

Figure 4. From Local to Global Profiles

The allocation of abstract resources has to happen based on decision-theoretic considerations about how useful a particular assignment is. Each subordinate problem solver therefore statistically monitors a so-called *local profile* which describes its current performance or utility (V) in relation to any possible resource configuration. A *global profile* can be generated by game-theoretic considerations ($\tilde{V}$ – see Section 3) and the use of higher-level knowledge about how resources are dynamically affected by the subordinate stages. An example is the knowledge whether a semaphore has been released and therefore is free for further use. The global profile integrates frequently gathered pieces of local information into a temporal projection or hypothesis of the system's development.

Figure 4 illustrates the functional relationships encoded in profiles. The question of how they are actually represented is strongly coupled to the question of what constitutes the decision problem herein. Generally speaking, we regard the task of each stage as a search for an optimum of the *objective function* in the multi-dimensional *search space* represented by the global profile.

Relation to the Application In the transportation domain, truck agents compete for abstract resources such as fuel, working time of drivers, maintenance capacities, and of course the delivery tasks and planning/execution time. Shipping companies compete for even more abstract concepts such as pools of tasks, geographical or freight-type monopolies, and the capacities of other transportation systems. Arrangements with respect to these resources can be made, for instance, by a broker agent representing a company association.

5.2 Complex Decision Making

The performance of agents is usually not purely measured in terms of a real-valued function. Rather, agents are faced with a combination of symbolic characterizations of desired system states (*goals*) and numeric priorities between those (*preferences* or *utilities*). Originally, AI and Operations Research have developed rather isolated methods, namely *planning* systems and *optimization* routines to deal with these complementary representations. During recent years, there has been a deeper understanding of a unifying methodology which finds its practice in approaches to *decision-theoretic planning* [5].

Basically, decision-theoretic planning follows the deliberative AI paradigm in producing temporally structured sets (plans) of means (actions). Each action is annotated with preconditions, i.e., applicability conditions, and effects, i.e., changes imposed by the execution of the action. Instead of simply looking for a plan that is applicable in the current situation and that installs a symbolic goal by its effects, decision-theoretic planning additionally assigns a range of utility to each intermediate, incomplete plan. This assignment depends on the value of system states that the plan probably produces and to the cost of the actions performed within. Using these estimates as a guiding principle during search, the decision-theoretic planner returns the plan with maximal (expected) utility.

Such an approach is suitable to represent the (multi-agent) global profile and the involved reasoning: the plan whose application guarantees the optimal satisfaction of the goals, i.e., it establishes a state that maximizes the utility, corresponds a path from the current situation to the global maximum in the global profile (see Figure 4). In our scheme, actions model changes of the resource allocation, i.e., a restructuring of the subordinate problem solvers. Therefore, symbolic preconditions and effects of actions in AI planning can be enhanced with quantitative alterations of abstract resources.

Finding such optimal solutions comes at the cost of high computational complexity: the planning of a complex sequence of system reconfigurations consumes also computational resources. Indeed, the abstract resources available to the planner will be partly mapped to the time and space consumption of the planner's computation. The rest constitutes the resource representations that are the issue of the operators in the plan.

Relation to the Application Complex decisions in the transportation domain are, e.g., the planning of an optimal route for a truck incorporating fuel costs, road tolls, service time, and (un-)loading processes. For a shipping company, the reorganization of truck depots is a rare, but important measure to optimally adapt to the customers' needs. Useful long-term arrangements between companies of an association are another area, where complex decisions are required.

A shipping company will accept a number of tasks with a deadline. Afterwards, its representative agent has to decide about how long to deliberate the optimal distribution of tasks to trucks and will then distribute the remaining time to the truck agents which finally plan and execute their tours.

5.3 Simple Decision Making

As just discussed, complex decision making introduces a high complexity. This may lead to the consumption of the complete system time before actually triggering some external action to meet the goal. Inspired by [13], we argue that there has to be some meta-level guide that prunes the vast number of choices open to the deliberative module. On the one hand, this is given by the upper social stages and their decisions in the form of abstract guidelines. On the other hand, the refinement of these decisions can be done by a simple decision making mechanism operating on the local profile (see Figure 4 and Figure 2).

A variation of the steepest ascent method [16] 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. By frequently *sampling and monitoring* the performance of the underlying complex decision according to the current resource configuration, it is able to estimate the surrounding area of the local profile (see Figure 4). 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. If the performance is representable by some differentiable function, the best direction can be determined using partial derivatives. Otherwise, a sufficient approximation based on the collected samples (e.g., interpolation) is used.

The basic assumptions of the steepest ascent method operating on the local profile coincide with Russell and Wefald's meta-greedy method which has been proven applicable. For example, *concavity* of the objective function is an important prerequisite to guarantee even global optimality. A simulated-annealing technique [9] helps not to run into local optima in case of non-convexity. However, since the environment may change rapidly, the achievement of only local optima is not a serious problem, since (global as well as local) optima may lose their optimality instantly. Other assumptions, such as convexity of the search space and ordinal dimensions are met by our abstract resource representation which is discussed in [3].

Relation to the Application Simple decisions in the transportation domain are, e.g., the truck agent's decision to accept or reject a particular delivery task. For a shipping company, marginal size variations of the active fleet, (de-)installations of particular depots, or assignments of a truck to one of those geographically distributed sites can be realized using this mechanism. Finally, an association broker agent adapts the possible negotiation schemes between shipping companies according to current necessities and arrangements.

5.4 Combining Complex and Simple Decision Making

As stated above, this combination of complex and simple decision making follows Russell and Wefald's meta-level reasoning architecture:all possible *object-level* actions for the *complex decision making unit* cover resource allocation for lower stages while the *simple decision making unit* on the *meta-level* reasons about the optimal adjustment of the complex decision maker. Obviously, the presented architecture cannot guarantee global optimality of the whole agent society, but in analogy to [13], globally optimal behavior is *approximated* in a tractable fashion.

6 Holonic Structures for Resource Management

As discussed in Section 4, we embody our resource allocation methods (the steepest ascent method and decision-theoretic planning) explicitly in an agent that is based on the INTERRAP architecture. The resulting system can be viewed as an implementation of the so-called *holonic* principle, as we point out now:

According to the Hungarian philosopher Arthur Köstler [10], a *holon* is an entity with self-similar characteristics: First, it appears as an inseparable unit (the Greek *holos* meaning *whole*); however, a closer look reveals that it can be subdivided into new structures (the suffix *on* meaning *particle*) which exhibit a similar appearance: Hence, the process of splitting a holon can (theoretically) be performed *ad infinitum*. On the other hand, holons can join in order to form a new entity, which again has holonic characteristics.

Such a description does initially not coincide with the common use of the term *agent* in DAI: In many applications, the model granularity is determined in advance, leading to serious drawbacks. Once defined, an agent has to represent an entity no matter if this task is too demanding or undemanding for it. However, our approach of a hierarchically moderated group of agents is much more flexible since the group structure can be made subject of the resource allocation mechanism. This leads to dynamic self-organization of the group and hence realizes the holonic paradigm.

[4] defines a collection of criteria to identify holonic structures in an application domain and also requirements and restrictions to the architecture of the agent society; all of which can be met by the approach presented in this paper. The two most important restrictions to an agent society are the following:

First, member agents of a holon must strive towards least one common goal (which might be represented explicitly or implicitly), leading to emergent overall goals of the holon. Obviously, these emerging goals may not conflict with the goals of the member agents, leading to the fact that agents can only be members of several holons with conflicting overall goals if the agents in question are indifferent towards such conflicting goals.

Second, agents forming a holon still remain autonomous problem solvers. However, they have to accept autonomy restrictions: agents commit report local profiles to higher social stages and to accept guidelines derived from these higher stages. In particular, the permission to communicate with agents outside of the holon is regarded as an abstract resource, given or denied from higher stages.

If we refer to a holon as a collection of agents in a society equipped with the presented resource allocation mechanism we can introduce an additional level of abstraction: we can identify a holonic entity and hence abstract away from the underlying agents, analogous to the way we previously had identified agents (and had abstracted from even lower-level concepts such as objects, processes or threads).

7 Conclusion and Future Work

We have presented a hierarchical resource adaption model for MAS that extends the theoretical framework and practical architecture of [12, 13] to gather a tractable, robust, and scalable approximation of bounded optimality. The main modeling tools are the concept of abstract resources and the holonic agent in order to smoothly bridge the micro-macro-gap often found in the design of complex systems.

[12, 13] describe a model like ours as *meta-level rational*. They furthermore propose that in order to obtain the best agent program, or the best set of agent programs in our case, *compilation* techniques have to be applied. Compilation merges the heuristic information gathered by the meta-level into the object-level data structures, thus making the meta-level redundant. With respect to boundedly optimal societies, such a compilation corresponds to the establishment of *social laws* which is a highly interesting perspective in an interdisciplinary setting of social sciences.

An implementation and evaluation of our work relies on the INTERRAP architecture [11] which is realized both in JAVA^{TM} and OZ[15]. Especially constraint-based methods, as provided in OZ, are a natural way of describing both the simple and the complex decision making algorithms, such as the logic-based planning module of INTERRAP [8]. Besides the mentioned transportation domain, current investigations also cover flexible manufacturing systems and the soccer simulation of RoboCup [7].

Acknowledgments We would like to thank Gero Vierke for fruitful discussions, in particular concerning the TELETRUCK application and the modeling of holonic structures.

REFERENCES

[1] H.-J. Bürckert, K. Fischer, and G. Vierke, 'Transportation Scheduling with Holonic MAS – The TeleTruck Approach', in *Proceedings of the Third International Conference on Practical Applications of Intelligent Agents and Multiagents (PAAM'98)*, (1998).

[2] K. Fischer, J. P. Müller, and M. Pischel, 'A model for cooperative transportation scheduling', in *Proceedings of the 1st International Conference on Multiagent Systems (ICMAS'95)*, pp. 109–116, San Francisco, (June 1995).

[3] C. Gerber and C. G. Jung, 'Towards the bounded optimal agent society', in *Distributed Cognitive Systems*, eds., C. G. Jung, K. Fischer, and S. Schacht, number D-97-8 in DFKI Document, Saarbrücken, (1997). DFKI GmbH.

[4] C. Gerber and G. Vierke, 'Holonic Multi-Agent Systems', Research Report, German Research Center for Artificial Intelligence, (1998). To Appear.

[5] P. Haddaway and S.Hanks, 'Issues in decision-theoretic planning: Symbolic goals and numeric utilities', in *Proc. of the Workshop on Innovative Approaches to Planning*, pp. 48–58, (1990).

[6] L. M. Hogg and N. R. Jennings, 'Socially Rational Agents - Preliminary Thoughts', in *Proceedings of the 2nd Workshop on Practical Reasoning and Rationality*, eds., J. Bell, Z. Huang, and S. Parsons, Manchester, (April 1997).

[7] C. G. Jung, 'Experimenting with layered, resource-adapting agents in the robocup simulation', in *Proc. of the ROBOCUP'98 Workshop*, (1998). to appear.

[8] C. G. Jung, 'Situated abstraction planning by abductive temporal reasoning', in *Proc. of the 13th European Conference on Artificial Intelligence ECAI'98*, pp. 383–387. Wiley, (1998). to appear.

[9] S. Kirkpatrick, C. Gelatt, and M. Vecchi, 'Optimalization by simulated annealing', *Science*, **220**, 671, (1983).

[10] A. Koestler, *The Ghost in the Machine*, Arkana Books, 1989.

[11] J. P. Müller, *The Design of Intelligent Agents : A Layered Approach*, volume 1177 of *Lecture Notes in AI*, Springer, 1996.

[12] S J. Russell and D. Subramanian, 'Provably Bounded Optimal Agents', *Journal of Artificial Intelligence Research*, **2**, (1995).

[13] S J. Russell and E. Wefald, *Do the Right Thing*, MIT Press, 1991.

[14] H. A. Simon, *Models of Bounded Rationality*, MIT Press, Cambridge, 1982.

[15] G. Smolka, 'The Oz Programming Model', in *Computer Science Today*, ed., Jan van Leeuwen, Lecture Notes in Computer Science, vol. 1000, 324–343, Springer-Verlag, Berlin, (1995).

[16] G. Zoutendijk, *Mathematical Programming Methods*, North-Holland, Amsterdam, 1976.