Self-Configuring Resource Management in Cooperative and Uncooperative Autonomous Systems

Philipp Obreiter   Michael Klein

{obreiter,kleinm}@ipd.uni-karlsruhe.de

July 31, 2003

Technical Report Nr. 2003-15

University of Karlsruhe
Faculty of Informatics
Institute for Program Structures and Data Organization
D-76128 Karlsruhe, Germany

Abstract

In several circumstances, cooperation among autonomous agents is a prerequisite for effective or efficient task processing. This stems from the inherent or transient asymmetry of the agents' resources and capabilities. Since agents process tasks autonomously, they have to decide by themselves under which conditions and to which means they cooperate. Such local decision making renders dedicated entities for resource management dispensable and, thus, may increase the system's fault-tolerance. In this report, we propose a framework for self-configuring resource management in cooperative and uncooperative environments. Therefore, we recognize the relatedness of autonomous agents and economic entities. Both are REMMs [1] and assess their resources and demands locally which leads to more efficiency of the overall system. We identify and discuss two dimensions of autonomy, i.e. principal-agent and inter-agent autonomy. If all agents are part of the same organization, inter-agent autonomy gives way to cooperating agents. We elaborate distributed resource management for such cooperative systems. In case of inter-agent autonomy, uncooperative agent behavior is encouraged by their REMM nature. Yet, we show that the presented concept of resource management is seamlessly applicable to such an environment. Lastly, we identify discovery overlays as an efficient means of disseminating resource asymmetry awareness.

Contents

1 Introduction

An autonomous agent is designed, deployed, and maintained in order to execute a set of specific tasks. The conception of the autonomous agent has to take into account varying requirements and general conditions and, thus, is likely to be a compromise. Therefore, in a network of autonomous agents, it is probable that an agent A discovers another agent B in its vicinity which is more specialized in a specific task. In such a case, it is highly desirable that agent B executes the task on behalf of agent A since he is able to do so more efficiently. For instance, a small gripper robot might have to process a complex computation by himself if no other device is in its vicinity, whereas, in the presence of a powerful computing device, it delegates such a computation. Apart from ad hoc availability of such resources and services, the inter-agent dependency may be volitional in conception, e.g., by deploying a specialized agent for each task. In any case, cooperation among autonomous agents is a prerequisite for effective or efficient task processing.

The major challenge for an autonomous agent consists in estimating under which conditions cooperation becomes necessary and more resource efficient than isolated task processing. Since the agents are autonomous, they have to make the decision themselves according to their view of the environment. Such local decision renders dedicated entities for resource management dispensable. Furthermore, it may increase the system's fault-tolerance to agent failure and compromise.

Autonomous agents maximize their utility while minimizing resource consumption. Apparently, the behavior of autonomous agents is closely related to the one of economic entities. In economics, there are models that capture such behavior, notably the Resourceful Evaluative Maximizer Model *(REMM)* [1]. According to this model, each entity is resourceful and evaluates means of maximizing its utility, i.e. satisfying its demands. It is conjectured that overall resource consumption and service provision is efficient if each entity acts according to REMM and cooperate on markets [2]. For a set of such entities, resource management is performed locally and, thus, is self-configuring.

It seems promising to apply REMM to autonomous agents. Therefore, agents must be capable of assessing resources and require a notion of utility. Apparently, these requirements are contingent upon the agents' autonomy.

The outline of this report is as follows: In Section 2, we examine different types of agent autonomy. For the special case of an isolated agent, Section 3 describes the key concepts of resource management. The concept is generalized in Section 4 by considering the potentials of cooperation. Still, it is assumed that the agents are collectively subordinate to a common goal. Therefore, in Section 5, we discuss resource management for agents that tend to uncooperative behavior by maximizing their selfish utility. The practical issue of efficiently brokering cooperation is approached in Section 6. We survey related work in Section 7 and, finally, in Section 8, we conclude the report.

2 Dimensions of Autonomy

By referring to agents as autonomous, two different concepts are addressed. On the one hand, each agent autonomously makes decisions and processes tasks on behalf of its principal. On the other hand, agents might not depend on other agents in order to fulfill their tasks. In this section, we take a closer look at both dimensions of autonomy.

2.1 Principal-Agent Autonomy

Each agent is designed, deployed, and maintained by its principal, e.g., by an organization. The principal defines the agent's tasks and, in addition, the principal makes its utility and general conditions explicit for the agent. In this context, the agent's autonomy ensues from making independent decisions without consulting the principal. Yet, the agent's decision is made within the framework of the principal's specification. Obviously, the specification's accuracy is crucial for achieving the goals the principal aims at.

In contrast to human agents, an automated agent is designed by the principal so that the utility functions of the principal and the agent do not conflict. If the principal specifies its utility accurately enough, the agent may even become a perfect agent [3]. In any case, the application of automated agents eliminates agency costs. This insight is especially important for autonomous agents because autonomy generally entails information asymmetry. Take for instance a soldier *(agent)* the utility function of whom differs from the one of his commander *(principal)*, e.g., because of his risk aversion. Without additional mechanisms that compensate for information asymmetry (the commander has no complete knowledge of the soldier's actions), the agents are subject to adverse selection [4]. Yet, from an agency cost perspective, information asymmetry ceases to be detrimental if the agent is automated.

2.2 Inter-Agent Autonomy

The other dimension of autonomy becomes apparent for an agent that is able to execute its tasks without relying on the help of other agents. Nevertheless, this kind autonomy does not rule out inter-agent cooperation in order to execute tasks more efficiently.

Regarding cooperation, two behavioral patterns are identifiable. On the one hand, agents may be inherently motivated to cooperate, e.g., because they act on behalf of the same principal. On the other hand, agents may perceive cooperation solely as a means of maximizing their selfish utility [5].

In such a case, they resemble economic entities and tend to be uncooperative unless there is an added value of cooperation like remuneration. An agent chooses its behavior with regard to the other agents. Take for instance the agents A_1 and A_2 that are deployed by the same principal. They might exhibit cooperative behavior, whereas, at the arrival of an agent B_1 that is deployed by another principal, A_1 and A_2 are uncooperative with regard to B_1 and vice versa.

3 Resource Management of an Isolated Agent

If an agent is isolated from its principal and other agents, it has to manage its resources efficiently by itself. In this section, we discuss such isolated resource management which builds the foundation for distributed resource management in a network of agents.

In principle, it suffices that the principal specifies its utility function and a set of general conditions. Then, the agent makes decisions by maximizing the utility while considering these conditions. However, an agent is generally composed of several entities that process subtasks, so that global maximization becomes complex or even infeasible. It seems more promising to subdivide utility optimization to the respective entities, which would diminish the complexity of conceiving agents by rendering them more modular. For example, communicating agents require protocol entities in order to run the protocol stack [5]. The very nature of such protocol entities consists in abstracting from other entities' tasks. Therefore, the conception of an inter-entity utility maximization seems daunting. Nevertheless, the interdependencies of consuming the device's shared resources have to be made explicit, as it is shown in [6].

We suggest resource assessment as the glue of entity level decision making. If each entity is aware of the scarceness/preciousness of the respective resources, it is able to adapt its resource consumption patterns in order to execute its subtask accordingly. In this regard, the evaluative nature of the entities is revealed, which is conform to them REMM model. During decision making, each entity is aware of several resource tradeoffs [7] so that, the assessment of the respective resources given, it is able to determine the most efficient way to process its subtask. For instance, such resource tradeoffs are memory vs. computation/bandwidth *(cache)* and energy vs. computation/movement *(sleep)*. In conclusion, resource assessment specifies behavior of the agent's entities without inducing an explicit strategy choice of such entities.

The notion of resource assessment may also be applied to the services that an entity provides. Apparently, this becomes necessary if entities consume added value services and, thus, are not aware of the resources they implicitly consume. In such a case, each entity has to assess the value of the services that it provides by estimating the resources that it requires therefor. Consequently, service assessment is derived from resource assessment and may be treated accordingly. Therefore, in the following, we will focus on resource assessment.

The principal determines the behavior of its agent's entities by assessing its resources. Resource assessment consists of a static and an adaptive part. The principal statically assesses the scarceness of every resource regardless of its transient usage. E.g. a sensor agent might be assigned high values for its battery power. In addition, the principal assigns a value function that introduces adaptivity to resource usage. Adaptivity is required with respect to the availability of specific resources. For instance, resources of heavily loaded agents become more expensive. More generally speaking, the adaptivity of assessment takes the resources' opportunity costs into account.

More advanced resource assessment strategies also take future circumstances into account. For example, when an autonomous agent departs for an energy-intensive trip without opportunities of battery recharging, it makes sense to anticipatorily increase the value of its energy resources. Therefore, the agent requires dedicated entities that forecast future circumstances and adjust accordingly the principal's static resource assessment.

4 Resource Management for Cooperative Agents

In general, an organization deploys several autonomous agents. If they are within reach, their decision making has to take into account the opportunity for cooperation. Regardless of the existence of inter-agent autonomy, the agents might be well aware of their common principal and, thus, pay attention to the other agents' resource assessment. Therefore, in this section, we apply the aforementioned concept of resource management to networks of cooperative agents. The concept's generalization by the means of distribution makes its strong points evident.

The perception of an agent as a composition of entities renders the generalization to distributed resource management straightforward. Whether the resources are remote or not, their assessment is considered as valid by the consuming entity. Therefore, we assume that the agents are cooperative in terms of valuing local and remote resource consumption equally. Take for instance a gripper robot and a transmitter that are both able to establish an earth link. The gripper robot does not possess dedicated hardware and superfluous energy resources. Therefore, its link entity assesses its service at 20 units, whereas the transmitter's link entity assesses the same service at 5 units. If the gripper robot requires the earth link, it will get known of the transmitter's assessments *(see Section 6)* and establish the link with its help. Yet, if there is no transmitter in its vicinity, it might find it not worth of consuming 20 units for such an earth link. The other way round, if the transmitter requires the link, it will not use the gripper robot's link because it is more valuable. The transmitter would exhibit different behavior, if it is selfish and therefore is only concerned by the consumption of its own resources. Section 5 examines resource management for such uncooperative agents.

In conclusion, local resource assessment not only determines the behavior the respective agent but also from the agents in its vicinity. Furthermore, the proposed resource management is self-configuring and results in efficient use of resources.

5 Resource Management for Uncooperative Agents

In a networks of autonomous agents, agents are not necessarily assigned by the same principal or they might not be aware of it. Then, agent behavior is solely contingent upon the consumption of its own resource and, thus, agents become uncooperative. Without adjustment, the aforementioned resource management concept would fail since, then, remote resources are for free and their consumption is preferred. In this section, we refine the resource management scheme for this purpose.

Uncooperative agents do not compare the scarceness of other agents' resources with the benefits that arise from consuming them. Therefore, the costs of consuming remote resources have to be incured locally on the agent that initiates resource consumption. A straightforward solution of localizing costs and utility consists of making the unit of resource assessment explicit by the means of remuneration. Remunerating resource management for networks of autonomous devices has been widely examined for ad hoc networks [5].

If remuneration is solely subject to resource assessment, resource management does not differ from the one for cooperative agents *(Section 4)*. Yet, the relatedness of economic entities and autonomous agents yields a more generic assessment of the remuneration. Let us assume that the providing entity assesses its service inferior to the utility that the consuming entity assesses to derive from it. Then, there exists a surplus that should be bargained on [8]. If there is no surplus, cooperation does not materialize since it would result in inefficient resource consumption. Apparently, efficient resource management requires a means of efficiently discovering provider-consumer pairs with a surplus to bargain on. This topic is discussed in the next section.

6 Service Discovery and Resource Efficiency

An isolated agent has direct access to the assessment of resources since it is performed locally. However, for distributed resource management, potential consumers have to discover agents that assess the respective resources inferior to the utility that the consumer derives from them. Apparently, such a discovery is not contingent upon the cooperativeness of the agents and, thus, it is a prerequisite for the resource management of both Section 4 and Section 5.

Existing approaches of decentralized service discovery for autonomous agents [9][10][11][12] apply overlays in order to route efficiently the providers' service advertisements and the consumers' service requests. It seems promising to couple such a service discovery with resource and utility assessment. In the following, we examine such a coupling in the presence of static and dynamic assessment.

6.1 Discovery in the Presence of Static Assessment

The basic concept of making service discovery aware of resource and utility assessment consists of including them into the description of discovery messages, i.e. advertisements and requests. Therefore, service assessment is added to advertisements, whereas requests comprise utility assessment. Consequently, matching requests to advertisements has to take both assessments into account.

If both resource and utility assessment are static, such an extension of service discovery suffices in order to efficiently establish consumer-provider pairs. Nevertheless, resource management is rendered more efficient if the difference between resource and utility assessment is maximized. Even for uncooperative agents, both the consumer and the provider take advantage of such a surplus maximization.

Obviously, discovery overlays should exploit the semantics of assessment in order to become more efficient. For example, if the advertisements of two equivalent services have to be dispersed in the overlay, it might suffice to disperse the service that is assessed at a lower value.

6.2 Discovery in the Presence of Dynamic Assessment

On the one hand, dynamic resource assessment arises from the adaptivity of the value function and the anticipatory assessment. On the other hand, dynamic utility assessment inherently stems from the transient demands of the agents. Hence, in most scenarios, service discovery has to take into account dynamic assessment of resources and utilities.

Yet, the extension of service discovery as applied for static assessment might still be applicable. If the discovery overlay privileges one type of discovery messages, e.g., by caching them, the unprivileged message type may be subject to dynamic assessment. For instance, current discovery overlays [9] privilege service advertisements by caching them in specific nodes of the discovery overlay. In such a case, service requests are processed instantly so that dynamic utility assessment does not pose any problems. However, if resource assessment is more dynamic than utility assessment, it would make more sense to privilege service requests. Consequently, discovery overlays should be contingent upon the respective assessment profile.

If neither resource nor utility assessment are static, the aforementioned extension of service discovery has to be altered. Service advertisements may contain a lower bound for the assessment of the respective service. In general, such a lower bound is obtained by assuming that the required resources lie idle. Apart from adding such necessary conditions to advertisements and requests, an estimation of the assessment's volatility, e.g. by the means of probability distributions, has to be included in order to increase the chances of finding the most efficient provider-consumer pair.

In case of extremely volatile assessment, any extension of service discovery is doomed. Yet, this is matched with the inherent inefficiencies of discovery overlays for extremely volatile topologies.

7 Related Work

Agoric computing [13] suggests efficient resource allocation and consumption by the means of pricing mechanisms. In analogy to economics, it is argued that local decision making and pricing induce efficiency. Agoric computing requires a non-tamperable operating system that enforces market rules, e.g. property rights of resources, and asserts the entities' integrity, i.e. their encapsulation. However, agoric computing applies market mechanisms in order to prune efficient entities. The granularity of entities is not defined a priori yet it emerges with respect to the tradeoff between transaction costs and monolithic inefficiencies. As a result, agoric computing assumes uncooperative entities so that pricing cannot be reduced to its signalling role, as it is done for cooperative agents. In addition, the set of assessed resources is known a priori since service assessment is not part of the operating system.

Resource tradeoffs and means of resource substitution are explicitly considered by introducing resource classes [2]. In this context, the storage hierarchy yields distinct resource classes that reflect the tradeoff between access time and storage price. In [14], this principle is transferred to bandwidth in infrastructured WLANs. On the one hand, the stable allocation *(SA)* class allows for streaming communications. On the other hand, the instant allocation *(IA)* class offers a best effort service. Mobile devices notify their respective access point about their priorities with respect to the resource classes. The paper shows that, the tradeoff of stability *(SA)* and throughput *(IA)* given, the mobile devices choose the class that suits best to their service consumption profile. However, the approach encourages lavish behavior since there is no incentive to refrain from allocating when no bandwidth is required.

The aforementioned approaches require entities that enforce the pricing mechanisms. In agoric computing, the non-tamperable operating system takes on this task, whereas, in infrastructured WLANs, the access point enforces efficient bandwidth allocation and counters misbehavior of mobile devices. In our resource management approach, enforcing entities are only required for uncooperative agents. E.g., built-in security modules allow for the distribution of such an enforcement [5].

For cooperative entities, pricing mechanisms have already been suggested, e.g., in network pricing [15] among other things for load balancing. However, in the context of autonomous agents, the concept of assessment becomes even more attractive because it reduces the complexity of autonomous decisions, as pointed out in Section 3.

8 Conclusion

Agents process tasks autonomously so that they have to decide by themselves under which conditions and to which means they cooperate in order to assert efficient resource consumption. Such local decision making is self-configuring and may increase the system's fault-tolerance. Therefore, in this report, we recognized the relatedness to the REMM model of economic entities. We pointed out that there are two dimensions of agent autonomy. Both induce varying behavioral patterns of the agent. We proposed and elaborated a resource management approach for networks of autonomous agents. It is based on local resource and utility assessment. We showed that the approach is seamlessly applicable to uncooperative agents by introducing the notion of remunerations. Finally, we examined the coupling of decentralized discovery and our resource management scheme. It appears that, for most scenarios, it may be efficiently integrated into the discovery framework.

We currently work on such an extension of our discovery overlays [9][11][12] in order to take into account the discussed extensions of service discovery. More specifically, we aim at including static assessment, bounds, volatility and assessment profiles into the discovery mechanisms.

Bibliography

[1] Jensen, M.C., Meckling, W.H.: The nature of man. Journal of Applied Corporate Finance (1994) 4–19

[2] Miller, M.S., Drexler, K.E.: Markets and Computation: Agoric Open Systems. In: The Ecology of Computation. B. Huberman (ed.) (1988)

[3] Jensen, M.: Self-interest, altruism, incentives, and agency theory. Journal of Applied Corporate Fincance **7** (1994) 4–19

[4] Akerlof, G.: The market for lemons: Quality uncertainty and the market mechanism. Quarterly Journal of Economics **89** (1970) 488–500

[5] Obreiter, P., König-Ries, B., Klein, M.: Stimulating cooperative behavior of autonomous devices - an analysis of requirements and existing approaches. In: Second International Workshop on Wireless Information Systems (WIS2003). (2003) 71–82

[6] Obreiter, P., Klein, M.: Vertical integration of incentives for cooperation - inter-layer collaboration as a prerequisite for effectively stimulating cooperation in ad hoc networks. In: Second Mediterranean Workshop on Ad-Hoc Networks (MED-HOC NET 2003), Mahdia, Tunisia (2003)

[7] Lee, C., Lehoczky, J., Rajkumar, R., Siewiorek, D.: On Quality of Service Optimization with Discrete QoS Options. In: IEEE Real Time Technology and Applications Symposium. (1999) 276–285

[8] Muthoo, A.: A non-technical introduction to bargaining theory. World Economics (2000) 145–166

[9] Klein, M., König-Ries, B.: Multi-layer clusters in ad-hoc networks - an approach to service discovery. In: Proceedings of the First International Workshop on Peer-to-Peer Computing (Co-Located with Networking 2002), Pisa, Italy. (2002) 187–201

[10] Moore, M., Suda, T.: A decentralized and self-organizing discovery mechanism. In: Proc. Of the First Annual Symposium on Autonomous Intelligent Networks and Systems (AINS). (2002)

[11] Klein, M., König-Ries, B., Obreiter, P.: Service rings – a semantical overlay for service discovery in ad hoc networks. In: The Sixth International Workshop on Network-Based Information Systems (NBIS2003), Workshop at DEXA 2003, Prague, Czech Republic. (2003)

[12] Klein, M., König-Ries, B., Obreiter, P.: Lanes – a lightweight overlay for service discovery in mobile ad hoc network. In: 3rd Workshop on Applications and Services in Wireless Networks (ASWN2003). Berne, Swiss. (2003)

[13] Miller, M.S., Drexler, K.E.: Comparative Ecology: A Computational Perspective. In: The Ecology of Computation. B. Huberman (ed.) (1988)

[14] Liao, R.F., Wouhaybi, R.H., Campbell, A.: Incentive engineering in wireless LAN based access networks. In: Proc. 10th International Conference on Network Protocols (ICNP), Paris, France (2002)

[15] Wang, X., Schulzrinne, H.: Pricing network resources for adaptive applications in a differentiated services network. In: Proceedings of the INFOCOM'01. (2001) 943–952