

AMETAS

Good Migrations.

AMETAS White Paper Series

by Klaus Herrmann
and Michael Zapf

On AMETAS

July 2000

Johann Wolfgang Goethe-Universität Frankfurt/Main, Germany
Department of Computer Science

www.ametas.de

A High-Level Overview on AMETAS

This paper presents an overview on AMETAS and its terminology. The major concepts are introduced to guide the reader through the following White Papers and put the individual components, the possible application scenarios and the foundations of AMETAS into perspective. After reading this paper, the reader should have an understanding of what AMETAS is, what its goals are, what its major components are and in which domains it can be applied.

What is AMETAS?

AMETAS is an acronym for *Asynchronous Message Transfer Agent System*. It is a *middleware platform* for mobile-agent-based distributed applications. AMETAS provides a runtime environment called a *place* that must be installed on each computer that is to host mobile AMETAS agents. A place consists of several core services that enable agents to communicate and migrate in a secure way. Besides the places, AMETAS also provides classes for developing mobile agents and other components that allow the programmer to build complex and powerful multi-agent applications. A distributed naming service that enables AMETAS users to build up an arbitrary hierarchy of places and lets agents migrate between them completes the system.

What are Mobile Agents and How could I Make Use of Them?

Mobile agents originate from the research area of DAI (*Distributed Artificial Intelligence*) and present a new paradigm to build distributed applications. Today, such applications are mostly built on top of some middleware for RPC (*Remote Procedure Call*)-like communication between its components. Commonly used systems in this area are CORBA and Java RMI. They support client/server systems with statically installed objects on different computers to enable distributed computing. The abstraction of method calls is used to simplify the development of complex, distributed applications running on different hardware platforms.

Mobile agents are a totally different approach to distributed applications. An agent is a self-contained program that can communicate with other agents and perceive its environment to act according to changes in this environment. Mobile agents enjoy the additional feature of being able to travel across a network from one host to another. We call this travelling *migration*. An agent may decide on its own when and where to migrate to achieve a task given to it by a human user or another agent.

In client/server computing, data is shovelled from one host to another to be remotely processed and the results are shovelled back to the client. With mobile agents this scheme is reversed. The code migrates to the host where the data lies and processes it locally. This may greatly reduce network traffic and allow for disconnected computation, i.e. clients do not need to stay connected while the data is processed. Of course, mobile agents need to be somewhat more intelligent to achieve their task than stationary clients since they are expected to act without the constant need to be controlled. This new paradigm enjoys big advantages when large scale distributed applications need to be developed. Examples for such applications are given in the section on *application scenarios* below.

The implicit distribution of processing load and the increased robustness due to the elimination of *Central Points of Failures* makes the resulting applications much more scalable than traditional distributed applications. The possibility to disconnect while one's agents roam the net, working on a given task, may reduce connection costs.

The System's Goals

The goals of AMETAS could be summarized as follows:

- It is to provide a general-purpose, sophisticated mobile agent platform for large-scale distributed applications.
- It aims at enabling scientific research, educational use and commercial application likewise.
- It is supposed to be a contribution to the mobile agent research by supplying a system that puts a very high emphasis on agent autonomy by consequently supporting this autonomy within the system.

How can I Apply AMETAS for My Company?

AMETAS is available under specific license terms. A publicly downloadable version can be found at www.ametas.de. This version has full functionality and is reserved for educational and trial purposes only. Commercial users must contact the AMETAS Team at www.ametas.de and get a commercial license.

The Philosophy

As can be seen from the name *Asynchronous MESSage Transfer Agent System*, asynchronous message-based communication plays an important role in AMETAS. We consider this mechanism the only one that is compatible to the notion of agent autonomy. Therefore, AMETAS agents are build around this concept and their resulting autonomy may be put to use in all kinds of applications. Autonomy was not only the reason for introducing this communication concept but also for the development of several key ideas which not only proved to serve the goal of agent autonomy but are also usable in practice to develop conceptually clean applications with a high degree of flexibility and scalability. One such idea is splitting up the central term of *the agent* into three categories of *Place Users* as can be seen in the corresponding section below. One Reason for this separation was that a component that waits for requests, services them and waits again does not comply with the notion of agenthood. Therefore, services are not agents in AMETAS. By identifying conceptually different components within the system and by shaping AMETAS accordingly, we developed a conceptually clean system not only from a theoretical but also from a practical standpoint. Clean separations enabled us to develop a powerful security system to achieve a high degree of security while still being flexible enough to supply practically useful applications. To summarize our philosophy:

AMETAS puts the highest emphasis on agent autonomy and a conceptually clean system design for two reasons: 1) to be able to realize future trends

and developments and 2) to enable application developers to build well-structured, maintainable and effective software.

Application Scenarios

In this section we will briefly look at some application scenarios that profit by the usage of mobile agents. This list is far from being complete. Mobile AMETAS agents can be used in a wide variety of applications but the scenarios given below could be viewed as typical since they possess some key properties that make the application of mobile agents worthwhile in general.

Network Management Network Management (NM) is the process of monitoring and controlling the effectiveness and efficiency of a computer or telecommunications network. Ensuring that all systems work within specified parameters is the main task of NM. Achieving this task typically involves the transport of vast amounts of data from the managed devices to a central computer on which the data is processed and NM actions are initiated. This traditional view renders NM applications unscalable, inflexible and not very robust because they rely on a central management application. However, these problems can be solved if we apply mobile agents to decentralize NM by expressing NM tasks as agents that can migrate to managed devices and process the data locally. This way the application consisting of several agents can grow with the network and bandwidth is preserved for more important tasks.

E-Commerce Doing business and buying things over the Internet is usually termed *Electronic Commerce*. E-commerce applications consist of seller and buyer entities that enable a consumer to go shopping online, to view and select goods and pay them. The selection normally involves extensive searches through product lists of several vendors to find the desired product at the best price. If different vendors use different software, this process becomes cumbersome, expensive and often impossible. Mobile agents could be used to do the shopping. They can be equipped with interfaces to different shopping software and with a high-level specification of the consumers desires. With this information and the ability to pay goods, they can be started and go to work by migrating across the network to different vendors while the customer goes off-line, saving connection costs and being relieved from searching and comparing the goods.

Mobile Computing The usage of mobile and hand-held devices becomes increasingly popular all around the world. Keeping up an expensive connection while you move from A to B is not always easy and affordable. Here too, the ability of disconnected operation can be a great advantage. When we need to search the Internet for some information, we could as well employ a mobile agent to do this for us. The on-line time could be kept to a minimum without losing any information that could not be found due to time limitations. AMETAS explicitly supports this kind of usage through its concept of *temporary places* where place names and thus migration targets are transparently associated with different physical hosts, i.e. different IP addresses. The user can start his place wherever he wants and be integrated into the system under the same name that his agents know. That way his agents can always come back to him, no matter where he is.

The Core Services

After this motivational information we present a brief overview on the internals of AMETAS to supply the reader with enough information to understand the *AMETAS White Papers*. We start by introducing the *core services* of every AMETAS place. The core services consist of the facilities that every Place User¹ expects to find on an AMETAS place. They are absolutely necessary for the basic operation of Place Users. You could say that these services present food, water and the air to breath for Place Users. If one of these services was missing, an AMETAS application would not be able to function at all. Besides these services, AMETAS provides the possibility to extend a place's functionality by installing services on the Place User level. The core services are now described briefly.

The Post Office Relying on messages as communicational means, a place must provide a system for depositing and retrieving messages. This system is called *Post Office* (PO). It consists of a number of *mailboxes* that store messages. The Post Office takes two kind of requests from Place Users: a *deposit* request and a *retrieval* request. A deposit request is issued by a Place User to store a message in the Post Office which can be retrieved by others. Upon a retrieval request the PO looks at the inquirer's name and at the stored messages' receiver addresses to decide if the inquirer is eligible to receive a message. Periodic cleanups remove old messages and empty mailboxes to keep the Post Office small.

The Event System Information may not only flow from one Place User to another but also from the AMETAS core to Place Users. This information consists of important notifications about certain changes in the system that need to be transported to the receiver in a timely fashion. Therefore, the messaging system is not appropriate for this kind of messages. An event system was developed that lets the core fire events in case of certain situations and forwards these events to Place Users that have expressed their interest in this specific kind of event by registering at the *Event Manager*. Place Users get this event shortly afterwards and can react to it.

The Migration System The migration system provides agents with the possibility to migrate from one place to another. It takes care of packaging and encrypting agents and introduces certain performance enhancements like code caching. It proceeds with the migration protocol upon which a decision is taken whether the agent in question is allowed to migrate to its target place and in which form the migration will take place.

The Place Name Service The PNS provides a naming scheme for AMETAS places. It was inspired by the Internet's Domain Name System (DNS) and provides a hierarchical, domain-based name space possibly divided over several domain servers. Note that only place names are supplied by the PNS, not Place User names. By using the PNS, an existing structure of places can be dynamically extended.

¹*Place User* are agents, services and user adapters which make up the three types of entities that live on a place. We detail this separation in the section on Place Users.

Place Users

Not everything that lives on an AMETAS place is an agent. To build sophisticated applications we need entities that provide a user interface and components that provide services to agents. This cannot be build into the core AMETAS system since such entities are application dependent. Furthermore, as these components need to communicate with each other, we decided that they should *live* on the same layer and be called *Place Users* since they all use the place's core services, especially the messaging system. Three basic types of Place Users exist:

- **Agents** may be mobile and were designed to possess the properties that are generally assigned to agenthood. Therefore, they are restricted in what they are able to do at a place. Since mobile objects may be considered security breaches by their very nature, they have to be restricted to enable secure operation. Agents enjoy a very high level of autonomy: no other entity can do anything with an agent without its consent. This is ensured by strictly separating Place Users from each other and only allowing communication via the Post Office. This may sound like a severe limitation of agent functionality at first glance, but, of course, what an agent is willing to do is totally up to the programmer. In a sensibly designed application, agents will cooperate.
- **Services** are passive entities by nature. We normally expect a service to wait for requests, service them in a timely fashion and afterwards suspend itself and wait for the next request. Such entities cannot be considered agents since they don't enjoy any of the properties inherent to agenthood. Moreover, services must have the opportunity to access system resources like the file system to provide useful service. For these reasons we defined a second type of Place Users that are called services and may extend the basic functionality of a place, for example, by providing application-specific services like the access to a data base located on the local host. Agents may use this service and thus also gain access to system resources but this access is controlled by the service. The agents are only allowed to execute actions that the service provides and the service is able to reject actions issued by non-privileged agents. Therefore, service programmers may enforce any limitation imaginable and encapsulate access to system resources.
- **User adapters** integrate the user into the system. Since agents and services operate on the same communicational layer it does not seem reasonable to introduce totally new communication means for human users to communicate with agents. A user adapter may be seen as a translator. It can display some kind of interface for the user and translate the user's actions into messages that can be understood by other Place Users. *Why can't this be done by agents?* Because displaying an interface (possibly a GUI) and waiting for user input to translate it is conceptually the same behaviour that services exhibit. There is no autonomy in these actions. But, if we perceive the user adapter and the user as one entity, it becomes an AMETAS agent again. The user accounts for the intelligence and autonomy and the user adapter represents the part that is seen by the other Place Users and that communicates with them.