INTRODUCTION

To obtain maximum benefit from the increasingly widespread use of the Internet in today’s information-oriented society, powerful tools are required for gathering and preparation of information. Intelligent agents, which have emerged in recent years, provide a successful technological approach to information handling. Intelligent agents are becoming increasingly important because of their flexibility and power for co-operative working in solving complex tasks. Usually such an approach requires agents which have the capability for extensive co-operation and communication amongst themselves. This raises the question of security. But, up to now, little work has been done which addresses the security issues associated with intelligent agents.

For a new technology, such as that offered by intelligent agents, to be successful and widely accepted, it is necessary for systems, based on that technology, to be capable of maintaining security and consistency of operation when integrated into the existing infrastructure of an organisation. This paper explores some of the security issues relating to application of intelligent agents and the integration of such systems into existing organisations.

First, existing information security issues for enterprises are considered. Then, a short introduction to the new technology of agents and agent systems is given. Following this, the special security problems of the new technology of software
agents and the emerging risks for software and enterprises are discussed. Finally, a new security architecture for multi-agent systems is proposed, together with an explanation of how this multi-level architecture can help to improve the security of agent systems.

Security Issues in Information Technology Systems

Because of the increasing importance of timely reaction by an enterprise in an ever-changing, competitive market environment, convenient access to and use of up-to-date information is becoming a key factor for economic survival. Indeed, the continuing existence of an enterprise may be heavily dependent on the support provided by well designed and secure information systems made possible through the application of state-of-the-art information technology (IT). Secure and robust information systems are necessary to protect enterprises from bankruptcy after a possible collapse of the IT infrastructure.

The last few years have seen a steep rise in the globalisation of markets. As a result, commercial enterprises are forced to enter the global competitive arena and to adopt a strategy directed towards inter-organisational co-operation. The result is the deployment of increasingly more complex software systems, leading to extended communication requirements between the different IT systems, and hence to the need for an inter-organisational standard for IT infrastructure.

To avoid, or at least minimise the risks resulting from the introduction of new complex IT systems and their co-ordination among existing IT systems, it is necessary to identify the problems, risks, or threats in the overall IT infrastructure and to develop strategies to guarantee security in these systems.

In considering security issues we can distinguish between security and safety. The word security describes the ability of software to fend attacks against the system. But, safety means the robustness of information systems versus emerging problems and exceptions. This article mainly treats security issues in connection with safety; security problems resulting from the robustness of the system are more the object of software engineering research.

According to (Ranneberg et al. 1997) security is considered to mean the enforcement of protective goals against intelligent attackers. They identify four security threats with their associated protective goals, which those threats aim to attack (Table 1).

The above-mentioned classification of threats can be very helpful in the identification of possible risks. Many opportunities may exist for malevolent persons to create threats. It is possible to gain unauthorised access to information through eavesdropping on communication channels. Unauthorised modification

<table>
<thead>
<tr>
<th>Threat</th>
<th>Protective Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unauthorised information gain</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>unauthorised modification of information</td>
<td>integrity</td>
</tr>
<tr>
<td>unauthorised derogation of functionality</td>
<td>availability</td>
</tr>
<tr>
<td>inadmissible loss of bind</td>
<td>accountability</td>
</tr>
</tbody>
</table>
of information may be achieved through either active modification of messages (possible when the attacker has, for example, access to a router) or when the message is held and is transmitted later in another context (which is also possible when the attacker has access to a router).

The availability of information may be under threat either when the attacker can intercept transmitted messages en-route, or when the attacker is able to prevent the transmission of message by overloading the communication channel with spamming signals. The threat to accountability may be easily effected since users usually do not send a confirmation of receipt of routine messages; thus it is very difficult to ensure that a message was sent from a specific source.

A possible approach (see Figure 1) for a systematic research of security issues can be a combination of the causal model and level model of security in information systems, according to (Stelzer 1993).

The source of the risk becomes a menace to the elements in the system—leading to a primary consequence and thus to a secondary consequence. The threat may relate to one of several levels in a system: the physical, logical, organisational-social, and law and business levels, in the level model according to Stelzer (see Figure 2).

For a complete consideration of security threats it is necessary to consider also the nature of the attackers. Here it is possible to distinguish among four different groups:

- strangers,
- system users,
- system carriers,
- system developers.

In the search to identify a threat the developers of the system, in particular, may be overlooked, even though they are the group posing the highest potential threat to security. For the system developers it is very simple to introduce, for instance, “Trojan Horses” (Chess 1995) or viruses into the system. The avoidance of this threat can, in general, be very problematic and, in practice, difficult to solve.

In the particular area of communication Pfitzmann (1993) has identified different requirements to guarantee security (Table
2). But these requirements may, in certain situations, be in conflict and, in effecting requirements, a compromise solution may have to be adopted.

**INTELLIGENT AGENTS**

Agents represent a new approach to the development of software applications. In modern software development agent technology is becoming more and more important because it offers flexibility and also great potential to solve complex problems. Although there is much discussion about what an agent actually is (Franklin and Graesser 1996), perhaps the most widely accepted and most used definition, given by (Wooldridge and Jennings 1995), attributes the following properties to agents:

These properties of agents enable the construction of flexible software systems, in the form of multi-agent systems (MAS) which are able to adapt their

**Table 2: Security requirements**

<table>
<thead>
<tr>
<th>Protective Goal</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>confidentiality</td>
<td>message contents readable only by the communication partner</td>
</tr>
<tr>
<td></td>
<td>anonymity of sender and / or receiver of message</td>
</tr>
<tr>
<td></td>
<td>no possibility of determining the location of a mobile communication partner</td>
</tr>
<tr>
<td>integrity</td>
<td>detection of forgery of message contents (including message originator)</td>
</tr>
<tr>
<td>availability</td>
<td>allow communication between all partners who wish to communicate</td>
</tr>
<tr>
<td>accountability</td>
<td>receiver can prove that sender x has sent message y</td>
</tr>
<tr>
<td></td>
<td>sender can prove that he/she has sent the message (better than simply that the message has been received)</td>
</tr>
<tr>
<td></td>
<td>communication net provider has proof for his/her service</td>
</tr>
</tbody>
</table>

**Table 3: Agent properties**

<table>
<thead>
<tr>
<th>autocny</th>
<th>agents act without human control and have restricted control over their actions;</th>
</tr>
</thead>
<tbody>
<tr>
<td>social ability</td>
<td>agents interact with other agents (or humans) via a special communication language (e.g. KQML);</td>
</tr>
<tr>
<td>reactivity</td>
<td>agents are able to perceive their environment and to react to changes in them;</td>
</tr>
<tr>
<td>pro-activeness</td>
<td>agents do not just react; they are also able to behave in a goal-directed manner to achieve their aims.</td>
</tr>
</tbody>
</table>
behaviour dynamically to match the needs of the problems addressed. The autonomous and pro-active features of agents can provide the potential means of creating simpler software systems in which the overall task may be accomplished through the delegation of simple tasks to agents.

**Types of Intelligent Agents**

Agents may possess some form of intelligent problem-solving capability which enables them to solve tasks independent of the control of a user. There are several types of such agents: reactive, deliberative and hybrid agents.

A **reactive agent** possesses sensors with which to perceive its environment and effectors to cause changes to that environment. The agent receives a stimulus through the sensor and then decides, with the help of action patterns, which plan to adopt to solve a problem. Thus in response to a stimulus the agent produces a reaction. This form of agent is not goal directed and does not initiate the problem-solving activity.

A **deliberative agent**, in contrast, possesses a knowledge base which stores knowledge relating to the agent's environment and the problem domain. When the agent receives a stimulus, it is able to formulate a strategy for solving the problem at hand. The agent is pro-active; that is, because it 'knows' about its environment it is able to plan a set of steps to pursue its goal. The particular advantage of a deliberative agent is this possession of a kind of "intelligence", which enables it to operate in a more flexible manner than is possible in the case of a reactive agent.

A reactive agent is limited in its operation by the number of implemented and available stimuli, which, in turn, restrict its planning opportunities. However, one advantage of a reactive agent is its possible fast response to a stimulus. To combine the advantages of reactive and deliberative agents, hybrid agents were developed.

A **hybrid agent** receives a stimulus it searches for a plan appropriate to its current task. If no suitable plan has been implemented the agent then operates as if it were a deliberative agent, with the help of knowledge relating to the current task and which is stored in a knowledge base, together with a problem-solving mechanism.

Intelligent agents have been developed for various purposes:

- interface agents, e.g. MS Agent (1999)
- agents for information retrieval
- physical agents: i.e. in the form of a robot, e.g. RoboCup (1999)

**Mobile Agents**

The rapid developments in networking technology in recent years, in particular the Internet and the emergence of Java, have made it possible to design and implement mobile programs, for example Java Applets (1999).

A mobile agent is an intelligent agent which is "able to transport itself from one machine to another." (Franklin and Grasser 1996, p. 29). During the migration process the information associated with the agent moves with it. This enables a mobile agent to gather information on different computers and learn in real time, as it progresses.

At the present time, most mobile agents that have been developed are reactive in nature, because the necessary parts of the deliberative agent structure, i.e. the
knowledge-base and problem-solving mechanism, are too large for migration across the current Internet. But in the foreseeable future, with further developments in the Internet and agent technology, the development of mobile deliberative systems will be an important aspect of agent systems development, with important consequences for mobile agent security.

Multi-Agent Systems

A multi-agent system is a computational system in which there are a number of agents, having specific roles within some system or organisation, and being capable of interacting with each other and with their environment. The notion of an agent is an important concept in distributed artificial intelligence (DAI). Two major areas of DAI are: distributed problem solving (DPS) and multi-agent systems. According to (Bond and Gasser 1988), distributed problem solving considers “... how the work of solving a particular problem can be divided among the number of modules ... that co-operate at the level of dividing and sharing knowledge about the problem and about the solution”, whereas multi-agent systems are concerned with: “... co-ordinating intelligent behaviour among a collection of ... autonomous intelligent agents...”, and “... how they co-ordinate their knowledge, goals, skills, and plans jointly to take action or to solve problems”.

The agents of a multi-agent system co-ordinate their operations with the overall goal of solving a complex task together. This approach leads to what is known in agent terminology as emergence; that is, the overall problem solving ability of the MAS may be greater than the sum of the problem solving abilities of the individual agents.

Different levels of security threats exist with respect to intelligent agents, mobile agents, and multi-agent systems. These security issues will now be discussed.

SECURITY THREATS OF INTELLIGENT AGENTS

The general trend in the advance of software technology is towards increasingly more complex and sophisticated software systems. Alongside these developments, software standards, for example CORBA (OMG 1997), have evolved with the aim of enabling client-server based co-operation of software modules produced by different vendors in a simple way.

In recent years the interest in, and popularity of intelligent agent technology has steadily increased. The adoption of an intelligent agent approach can result in a reduction in the complexity of software systems. The properties of autonomy, social ability, reactivity and pro-activity (Wooldridge and Jennings 1995) enable the construction of software systems which are able to adapt their behaviour dynamically to the problem at hand. The autonomous and pro-active properties of intelligent agents allow the delegation of simple tasks to an agent, whose goal is then to carry out these tasks. Moreover, multi-agent system technology enables the separate problem solving capabilities of individual intelligent agents to be combined in order to solve a complex problem in an efficient and “natural” way (Bond and Gasser 1988). The ensuing co-ordination and communication processes
necessary in multi-agent systems can be very sophisticated: appropriate protocols are required to handle this problem (Durfee et al. 1989). Also, the technology of multi-agent systems and co-operative problem solving assumes the availability of appropriate problem-solving knowledge (Lander and Lesser 1992).

In relation to the above-mentioned properties of intelligent agents, various security issues arise, which, to date, have been insufficiently investigated. The security issues of intelligent agents can be of a technical nature (for example, problems with the communications technology, such as passwords and login procedures), or of an organisational nature (for example, the result of introducing multi-agent components into a legacy software system).

One problem in intelligent agent technology lies at the communication level. How, in the communications among agents, or among agents and humans, can one guard against: system crashes, unauthorised information gain, unauthorised modification of information, unauthorised reduction in functionality, and inadmissible interruption in connection, or prevention of connection between sender and receiver?

Because of the potentially powerful nature of their properties, there are further risks associated with the deployment of intelligent agents. Intelligent agents may be involved in different problem-solving situations, but largely, because they are autonomous entities, their behaviour can not be completely forecast. A possible scenario is to involve agents in an electronic market application, in which different software agents from different vendors are able to buy and sell goods in a common market place (Maes 1999). In such a scenario, since agents are not aware of the consequences of their own actions, this can lead to a market crash, caused by the amplification of the behavioural effects of the interacting agents (Kephart et al. 1991). With the aim of ensuring acceptable agent behaviour in market scenarios, self-co-ordination and automated negotiation among self-interested agents, and decentralised business information systems, have been investigated (Müller et al. 1999).

Security problems can arise when intelligent agents are integrated into a legacy software system, in which some security mechanism already exists. It is difficult to forecast how the existing security mechanism will react to the introduction of intelligent agents: will the agents be able to bypass this mechanism?

Another problem concerns the effect that the introduction of software systems, in general, and agent-oriented systems, in particular, will have on the running of an organisation. How will the organisational structure change?

Organisational structures and associated boundaries allow for secured (information) spaces in order to protect their integrity (Picot and Reichwald 1994). Agent systems, especially multi-agent systems, are developed using different concepts to build information and co-ordination structures. These may well be not suitable for building frontiers, disallowing integration, and controlling resources. Thus, the introduction of agent systems with their inherent co-operative behaviour, may require a modification to existing organisational procedures and structures.

Another security issue, which has emerged with the development of multi-agent system technology, is the problem of malevolent agents. These agents are able to spy on other agents in the system; they may also supply incorrect or false
information to the system. Several strategies have been proposed to solve this problem, by enhancing the benevolent behaviour of agent with competitive or antagonistic behaviour (Rosenschein 1985).

SECURITY ISSUES FOR SINGLE INTELLIGENT AGENTS

The security issues of intelligent agents are very similar to those relating to conventional programs. But the “intelligent” and autonomous behaviour, which are features of agent systems, aggravate the security risks.

One reason for the problem is that developers can implement Trojan Horses or viruses. This risk is often overlooked. But it has happened that so-called “sniffing” programs have been included in programs which are distributed over the Internet. These sniffing programs allow the distributor of the programs, to “sniff” the whole of the contents of a local drive, including the files, where passwords or login codes are stored. Most computer users are well aware of the chaos that can ensue from viruses. But viruses can also take the form of intelligent agents - working autonomously, endowed with some machine intelligence and able to change their own code.

Further problems arise with single intelligent agents, when they are introduced into legacy systems. The agents are often able to adapt their behaviour at existing security borders. So strict access rights are necessary for the protection of data in the system. Well designed security systems must be implemented within the agents. This requires a detailed knowledge of how the security mechanism of the legacy system operates, in order to adapt the agents’ security system so as to ensure compatibility with that of the legacy system.

The administration of resources can also be problematic with the introduction of agents. In current systems it is not usual to observe the resources used by a running program. In a multi-agent system, however, the agents do not necessarily operate within a language run-time environment - for example, that of Java - but in the system environment, where the agents have full access to system resources. Consequently, files can be built which use the whole available disk space, or where the whole of the computer system’s resources can be used. The risk of an agent “sniffing” data on the hard disk and its subsequent transfer to a remote computer via a modem is another problem which must be addressed.

The security problems of single intelligent agents are not so different from the security problems of normal systems. But, in the introduction phase of an agent system into an enterprise, the security mechanisms of the existing information systems must be maintained. It must be guaranteed that the introduction of an agent system does not adversely affect the work of legacy systems and programs, and that the security borders of legacy systems are still kept stable and secure.
SECURITY ISSUES FOR MOBILE AGENTS

“Let’s get mobile! This is the Zeitgeist at the end of this century.” (Tschudin 1999). Such euphoric may often be used to describe the technology of intelligent agents. But the mobility property of agents can result in many new security issues. However, at this point of time the most current work into the security aspect of intelligent agents is in the area of mobile agents.

The problem with mobile agent systems is that the owner of the hardware, the user of the program and the author of the software are different entities, each of which may, in general, follow different security standards and expectations. There are, according to Harrison et al. (1995), three security issues for mobile agent computing:

- authentication of the user who has sent the mobile agent, to prevent “spoofing” (i.e. spying) and authentication of the server and/or agent execution environment;
- determination of whether the user has the authorisation to execute the agent on the server and to scan the code of the agent to examine what resources it proposes to access;
- verification as to whether the agent is able and willing to pay for services provided by the server.

Farmer et al. (1996) list several security threats to mobile agents and their hosts. Against the agents there are the following threats:

- The agent cannot be assured that its state has not been changed on its way to the current host.
- It can not be guaranteed that the agent interpreter on a host has not been altered.
- At its arrival at the host the agent is passive and so it can not be assured that the agent program is correctly executed by the host. Also it can not be not be guaranteed that it is started correctly, that there are no changes to its code before it starts running, or that there is no interference whilst it is running.
- The agent cannot be assured that the execution environment will not stop its execution prematurely.
- It cannot be assured that the agent will arrive at its intended destination.
- The agent can not carry its own key in an unencrypted manner, because the agent interpreter could be spying on that key.
- There cannot be a secure means of communication between two mobile agents.
- The agent may be cloned by a hostile agent interpreter, which may then misuse this cloned agent; the cloned agent may then be indistinguishable from the genuine agent.

There are also threats directed towards the host computer system which enables the execution of mobile agents:

- The agent may gain access to the operating system of the host and thereby implant a virus or Trojan Horse.
- The agent could overflow the memory of the host and thus paralyse the host.
• The agent could change data in the memory of the host.

To solve the above problems several solutions to ensure adequate security mechanisms for mobile agents have been proposed (Farmer et al. 1996; Chess et al. 1995).

Agents can be protected against host attacks through the following methods:
• It is possible to encrypt the agent’s data using public key encryption to defend the mobile agent against unauthorised spying.
• Defend the mobile agent, via a digital signature, from unauthorised attempts to change the agent’s behaviour and the agent’s program.
• Agents should only co-operate with trusted hosts, to ensure that a host does not observe the agent’s behaviour for its own advantage.
• To ensure the parentage of data, the host can tag the data with a digital signature.
• Employ a certification mechanism in order that agents may identify trusted hosts.
• To ensure that the mobile agent in a network is still alive, the agent should send messages at regular intervals as to its whereabouts.

Methods to protect the host from attacks by agents include the following:
• To ensure the proof of integrity of an agent it is possible to mark the agent with a digital sign (Farmer et al. 1996).
• At this point of time it is impossible to analyse an agent to ascertain whether it is a virus or it includes a virus (Farmer et al. 1996). Harrison (1995) proposed the development of a programming language, with which it is not possible to develop viruses. But the nature of a mobile agent behaviour makes such a proposal impossible to realise.
• Rasmussen and Jansson (1996) developed an approach to ensure the secure execution of an agent, which is capable of distinguishing between hard security and soft security. With soft security the host observes the agent; when the agent performs an illegal operation it will be removed from the system. A record of the agent and the user of the agent will be stored in a database and in future operations the agent will be very carefully observed. The disadvantage of soft security is that the reaction comes after the illegal action of the agent has occurred.

With hard security an attempt is made to ascertain what kind of agent is attempting to run on the host. In this approach, agents are categorised into different groups, according to which they may be allocated special resources and access rights (for instance an editor needs only access rights for text files and minimal performance). A security assistant observes the agent’s behaviour and compares it with the expected behaviour of that type of agent.

In conclusion, one may ask the question: “Secure or Open, Is This the Question?” (Tschudin 1999). It is not necessarily a good policy either to renounce security in favour of openness and mobility, or to renounce mobility in favour of security. It is necessary to determine the optimal point on this spectrum, which allows a maximum amount of freedom of mobility without prejudice to an acceptable degree of security. Mobility is not antagonistic to security.

What are the implications of these problems faced by enterprises? One problem is that the proof of security of a mobile system can not be a static process;
it needs to be an active working process to guarantee the security of the system, because mobile agents may pose different risks for an enterprise during the whole of the time that they are operating. Thus an active running program is required to ensure the security of the system and to maintain a look-out for possible problems (e.g. viruses, false information from the agent, or potential attacks on the host system). The introduction of mobile systems (especially mobile agents) is very risky for enterprises, especially the introduction for security relevant regions in the enterprise. The risks and problems associated with the introduction of agents into an existing system have to be dealt with much more carefully than is the case with the introduction of static, normal programs. The use of mobile agents for tasks in which the agent is required to migrate to a special host and to work on a special problem, without any gathering and changing of information, is an appropriate and relatively secure application of agent technology. An example of this is the managing and maintenance of computer networks.

SECURITY ISSUES FOR MULTI-AGENT SYSTEMS

The paradigm of multi-agent systems is becoming more and more important for business applications, in which, in general, the agents are required to cooperate, and thus, to communicate extensively. In any real business application this raises the problem of security. It is the authors’ strong belief that the industrial success of multi-agent systems requires the identification, and appropriate solutions to specific multi-agent security issues.

Security-related issues, however, have not attracted as much attention as they deserve in multi-agent systems research to date. There is some work on strategic behaviours of agents (lying agents, antagonistic strategies, etc.), but without a clear focus on security. For an overview of such work see (Sandholm 1999). Other work considers organisational approaches to add the notion of stability to multi-agent systems (Kirn and Gasser 1998; Kirn 2000). This work also does not directly address any security-related issues.

Some work in multi-agent systems, and in co-operative problem solving in particular, assumes general accessibility of knowledge in a global information space (Lander and Lesser 1992; Gasser 1992). But no mechanisms exist which guarantee the security of information. Neither are there any database-like mechanisms to control information access, nor to guarantee information consistency.

Some research has been carried out which is directed towards ensuring secure communication between agents. One approach is the security architecture of KQML (Thirunavukkarasu et al. 1995). KQML (Knowledge Query and Manipulation Language) is a communication language and protocol, which enables agents to share their knowledge. This is an essential prerequisite for multi-agent systems and for co-operative problem solving. The KQML approach concentrates on the authentication of message partners (which are agents), to ensure the integrity of messages and accountability of the agents for the message that they send. At this point of time KQML is a quasi standard for communication in multi-agent systems, but the language does not include any mechanism for ensuring security.

Another well-known approach for defining security issues is the agent secu-
Intelligent Software Agents (FIPA 1998). In this specification there is also a subset of those threats which have been identified earlier in this paper. Thus the members of the FIPA committee identify possible attacks by:

- wiretapping (resulting in a loss of confidentiality),
- altering of data (attacking the integrity of messages),
- copying and replaying (leading to a deprivation of integrity or accountability),
- denial of service (resulting in a sacrifice of availability)
- abnegation and masking (both resulting in a violation of accountability).

The solution, which is provided consists primarily of an agent platform security manager (APSM) as a service running on each agent platform. But there are harsh restrictions associated with the warranty of this security manager. All messages between agents have to be sent using those platforms. If there are agents, which are not running on the same computer system as the APSM, they have to use a stub generated on their computer to ensure secure communication. The agent management system (AMS) and the directory facilitator (DF) have to communicate using public key encryption, and the APSM only accepts messages from agents within its own domain or from other APSMs. Additionally, each agent can encrypt its content tag before sending a message to the APSM. Security threats by non-agent software are not addressed and the FIPA committee concedes the possibility of further threats from the co-operation of different agents which can not be foreseen.

There are problems relating to the FIPA security specification, particularly with respect to the following points:

- A user model exists for each user of the multi-agent system. This is managed by an extra service, the user personalisation service, which automatically collects and evaluates data about the system user.
- The agent management system demands the private key of the user in order to sign the messages of the user agent on behalf of the user. This means that every public key for each user agent has to be deposited at a central location, which may then become a target for security attacks.
- An auditing mechanism logs all activities conducted at the agent platform. So the activities of all agents could be overseen by reading the audit log files which are normally filed unencrypted on the hard disk of the computer hosting the agent management platform.

It can therefore be seen that the FIPA solution for agent security management has some drawbacks, as a result of the centralisation of the information relating to the proposed security measures. In particular, when there is a possible integration of intelligent agents into organisations it needs to be ensured that the humans working together with such agents can become confident in the security mechanisms of the multi-agent system.

Most work on agents still assumes benevolent operation of a multi-agent system. But even when the possibility of dishonest or malevolent agents is acknowledged, and an attempt is made to deal with such a situation, there is insufficient attention paid towards the establishment of a framework that will help...
in: identifying security risks, countering any attacks, and resolving problems arising from successful attacks. At present, there are not even any proposals as to how an agent may protect itself. The following example illustrates further problems which may appear, following the introduction of a multi-agent system at the law and organization level.

**Example**

A hospital uses a multi-agent-based scheduling system for arranging patient appointments. Every ward in the hospital plans its own schedule. Every ward scheduler co-operates with an agent which is responsible for co-ordinating the different local schedules at the hospital level. This central agent is required to collect knowledge (and, as a consequence, to establish a kind of knowledge layer) related to the co-ordination task at the global level. Among other things, this knowledge will enable the co-ordination agent to derive information not only about spare resources, and patient schedules, but also about:

- individual appointments of the employees (nurses, doctors),
- the task load of each employee,
- the co-operation profile of each employee,
- disease profiles of patients, etc.

What can be learned from this simple example? It seems necessary to distinguish authorised agents from non-authorised agents. A possible solution is to establish a formal certification procedure, which guarantees that only approved agents will join the multi-agent system. This kind of information is suitable, for instance, for influencing the competition between enterprises and the behaviour of stockholders. The borders of multi-agent systems have to be secure, so that information can not be obtained from the system to the advantage of other companies. Also the introduction of false information and purposeful disinformation can cause serious problems for the behaviour of the system and its associated operations. Only authorised agents should be allowed to join a multi-agent system to guarantee security. These precautions can be compared with organisational rules in relevant security departments, where an authorisation, in the form of an identity card or a letter of recommendation is necessary to gain access to the department and thus to the available information.

It is possible that an agent should not be allowed access to certain information of a system, because the enterprise operating that system is in direct competition with the owner of the agent and thus does not wish to lose the advantage that this information may bestow. But a multi-agent system normally behaves co-operatively and benevolently. The (visiting) agent may ask the agents of the host's multi-agent system for help in accessing this information. These agents may not be able to detect whether or not the requesting agent is malevolent in nature. For such scenarios it is necessary to develop architectures and plan approved sequences of work to guarantee the security and safety of systems and their information.

Another problem relates to the integration of multi-agent systems into the organisational structure of an enterprise. Organisational structures are necessary in enterprises in order to co-ordinate the behaviour of employees, to control the
flow of business processes, to establish a system of responsibilities, to allocate resources, to establish organisational borders, and so on. This is, at least partially, in contrast to the general openness, and problem solving flexibility, of multi-agent systems. As a result, the behaviour of a multi-agent system (or of each of its agents) may annul organisational rules, which, up to that point, may have reliably guaranteed information security. It is thus important to recognise that the application of multi-agent systems can create a special type of technology-specific security risks.

As an example, in a hospital there exist many organisational structures, for instance the sub-division into different departments for specific tasks (radiology, surgery, etc.). Furthermore, in these departments there exist organisational rules, such as the scheduling of work, or the resource allocation of a hospital to the departments with fixed budgets under the control of a head of department. But, the introduction of multi-agent systems, for instance for resource allocation through the aid of an electronic marketplace (dealing with the scheduling of work or necessary hospital material from the cheapest dealer), may lead to a weakening of these borders. The head of department may no longer be responsible for some tasks. Some organisational structures may no longer be necessary such future work. Perhaps, some smaller departments can be combined, because the administration of some important tasks may now be handled by a multi-agent system. Some workflows may need to be organised in a different way from that of the legacy system. For example, the supply chain may have changed in such a way that the multi-agent system is able to order the necessary resources for itself without human help.

**Research Approach for Multi-Agent Security Assurance**

The approach being followed in the Multi-Agent Systems Security Project, which has recently been established in the multi-agent systems group at the Technical University of Ilmenau, is now briefly outlined. Multi-agent systems essentially solve problems by co-operation. In the research being pursued, the following entities and concepts are involved:

- agents which possess some degree of intelligence (e.g. deliberative agents, reactive agents),

| Layer 1: | creation of a new multi-agent system, and dissolution of an existing multi-agent system, |
| Layer 2: | management of agents joining, and member agents leaving an existing multi-agent system, |
| Layer 3: | modification of member agents’ involvement in / contribution to a multi-agent system, |
| Layer 4: | provision of access to the global information space of multi-agent systems, |
| Layer 5: | maintaining co-operation security within multi-agent systems, |
| Layer 6: | handling of security issues relating to the interaction of a multi-agent system with its environment / other multi-agent systems. |
• intelligent and strategic behaviours (rational, benevolent vs. antagonistic, mobile etc.),
• agent architectures of different types (static, dynamic),
• architectures for agent interaction (e.g. blackboards, contract net),
• co-operation procedures (bargaining, auctioning, voting, etc.).

Furthermore the project will consider and develop additional security-preserving layers, on top of the basic communication infrastructure, to provide the following security-oriented functions shown in Table 4.

Figure 3 shows a layered architecture for supporting the secure operation of a multi-agent system.

In a hospital scenario a multi-agent system can be built for the management of resource allocation. The common task of the agent system is to guarantee the supply of material for the hospital (medicaments, dressing material, etc.). The multi-agent

Figure 3: Security architecture for multi-agent systems
system builds an electronic marketplace where the supplier (or dealer) agent and the agents from the different departments of the hospital (customer agents) interact to meet their specific needs.

Layer 1 builds or destroys a multi-agent system on behalf of the user. The system will be built when a task needs a new multi-agent system for solving a problem. Once the task is solved, this system may then be destroyed. In the example, the multi-agent system will be created when the customers (hospital departments) need different materials or resources for their work (medicaments etc.). The agent system will be destroyed when all needs have been satisfied.

At Layer 2, agents may join and leave a multi-agent system. It is beneficial for an agent to join an agent system, where they may obtain information or help from other agents to assist in carrying out their own tasks. Within a multi-agent system the agents co-operate to solve tasks together. But every agent can address a different task from the common task of the multi-agent system, when the aim of such a task is not contrary to the aims of the common task. Only legitimate agents, with legitimated tasks, may join the system. In the hospital scenario the customer agents join the system and request resources via the agent marketplace. The marketplace may send invitations for offers to the different suppliers and the suppliers send their agents to deal with the customer. The dealer agents need authorisation to join the system.

At Layer 3, the multi-agent system may change the tasks of agents or bring agents together for co-operation. For instance, it is possible, according to this scenario, that different departments need the same material. To get a special price for a larger order the agents of these departments behave co-operatively and send a joint order to a supplier.

Layer 4 checks information from the environment, which is passed on to the agents through sensors. It is possible that the agents receive misinformation about market prices that may influence the negotiation process.

Layer 5 checks information on the co-operation between agents. It is conceivable that the supplier agents have common agreements about prices. Layer 5 can avoid such agreements and prevent this kind of co-operation.

At Layer 6 the agents interact with the environment. To guarantee that confidential information is kept within the system, the contact of agents with the environment can be observed and, if necessary, stopped at this level.

The architecture presented above provides a secure approach for communication with, and co-operation within a multi-agent system. But the main disadvantage of this architecture is its strong centralised concept, which obstructs the autonomy and flexibility of the whole multi-agent system and of the single agents. All agents have to submit to the security system. The realisation of such an architecture is a sophisticated process and is only suitable for systems with a large necessity for security, as in the above example, which deals with expensive goods and materials.
CONCLUSION

This paper has identified a number of important security issues which need to be addressed to ensure secure information processing in systems which involve the application of agent technology. The paradigm of intelligent agents represents a substantial and exciting development in software technology, but this technology also presents a new menace to security. The field of security in relation to intelligent agent systems has not been sufficiently investigated to date. Some reported work on security problems in intelligent agents exists. However, the main focus of attention in publications in this area so far has been mainly directed towards mobile agent systems, whereas the field of security threats in multi-agent systems has received relatively little attention. It must be made clear that a potential security menace exists in agent-based systems, at a level above that of the communication level and which cannot be solved by means of established communication security approaches: for instance, the menace to systems through the actions of malevolent agents, or the misappropriation of, or unauthorised access to knowledge of a confidential nature.

In the case of single intelligent agents, certain security problems can often be solved using conventional communications and computer science methods. Well developed login and access rights procedures can solve some problems in this context. But, some security aspects of single intelligent agents and, in particular, mobile or multi-agent systems cannot be solved through communications security approaches. The introduction of additional components, involving intelligent agents, into legacy systems can cause problems of security. Another often neglected and important point, is the maintenance of security across organisational boundaries, which, again, may be compromised through the introduction of agent systems.

The field of mobile agents is perhaps the best analysed, and most researched aspect of agent security because this area represents the most interesting and most realisable industrial application of intelligent agent methods. This is the reason why there are currently a number of commercial companies working in this field, for example: IBM Java Aglets (1999), BT Zeus (1999), Mitsubishi Concordia (1999). Some of these ventures have been developed for commercial use and have abandoned the field of pure research activity. But, the developments in the security approaches of these applications are also largely based on communication security levels.

Although these commercial applications are able to function without the investigation and development of security levels above the communication level, it should not be forgotten that these agents operate at a very low level of intelligence. In such situations the distinction between a program and an agent may be very small. In the future, in applications involving agents which are able to exhibit a greater degree of pro-active and autonomous behaviour, more rigorous investigation of, and attention to potential security threats above the communication security level will be necessary.

Multi-agent systems, in particular, warrant greater emphasis on the development of adequate security approaches, yet little research has been carried out in this direction. The issue of security with respect to multi-agent systems therefore
Intelligent Software Agents requires more extensive investigation. With MAS other security problems emerge. Dishonest agents, unwarrantable knowledge appropriation and transactions involving imaginary goods, are only some of the new security problems, connected with the operation of multi-agent systems. To solve these problems it is not sufficient just to develop communication security procedure, such as login or access right procedures. Aiming to recognise the behaviour of an agent and predicting the consequences of that perceived behaviour is only one possible approach. The approach which is being followed in the recently started Multi-Agent Systems Security Project represents an alternative strategy for the solution of the emerging problems in the application of agent technology.

REFERENCES


