Designing Trust with Software Agents: A Case Study

Stijn Bernaer¹, Patrick De Causmaecker², Martin Meganck¹, Greet Vanden Berghe¹

¹ KaHo St.-Lieven, Department Industrial Engineering
Gebr. Desmetstraat 1, 9000 Gent, Belgium
Email: {stijn.bernaer,martin.meganck,greet.vandenberghe}@kahosl.be

² Katholieke Universiteit Leuven Campus Kortrijk,
Computer Science and Information Technology
E. Sabbelaan 53, 8500 Kortrijk, Belgium
Email: Patrick.DeCausmaecker@kuleuven-kortrijk.be

Abstract

In this paper, we will address privacy and trust issues that arise in more advanced software systems. Though a lot of information is currently available in electronic form, not all of it should widely be accessible to everybody. The involved parties need full control on how their data are used and who has access. If the system consists of autonomous software agents, this problem requires extra attention and new working principles.

We illustrate this in the case of a communication platform for multimodal transport. The major aim of the communication platform is to enhance exchanging information and to ultimately improve organisation/collaboration within the transport sector. A better informed view of the transport sector will facilitate better considered decisions for users of the communication platform. The software system merits credibility by accurately modelling all the relevant real world interactions of potential users of the system. We opted for a connectivity solution in which software agents act as representatives of the parties involved. All agents can be equipped with human-like skills and qualities such as intelligence, autonomy, and the ability to cooperate, coordinate and negotiate.

We demonstrate how cooperation between parties can be achieved while respecting their sensitivity concerning information.

Keywords: privacy, trust, software agents

INTRODUCTION

Secure interchange of information can nowadays be supported by well known and well understood software components. Familiar solutions include digital signatures, encryption and hashing. When two parties, unknown to each other, want to safeguard privacy and trust in their communication and transactions, we enter into a more abstract level of security. A trusted third party, if there is one, can support the communication in this case. It mimics real world situations. An even bigger challenge is to build trust and privacy into a software system that lacks trusted third parties. In this case, software agents can bridge the gap between existing security technology and the user requirements. The challenge then is to incorporate two levels of trust: Trust questions which exist in the real world have to be modelled in the software agents, and the system in itself must show trustworthiness to the involved parties.

In this paper we intend to reflect upon these trust questions. After a presentation of software agent technology as a further development of computer science, a philosophical framework is drawn that is relevant for the ethical discussion. Next, it will be illustrated how trust sensitivities in real life have implications when implementing a software agent system; the example chosen...
is an agent model designed to provide decision support to all possible parties involved in a multimodal transport landscape. Finally, some conclusions will be drawn as to the further development of this technology, taking into account these ethical questions.

COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE

Computers originated in an era governed by machines. They were conceived as such by their first users. More than being mere machines, they could actually be considered as automates. Ever since they have been in use, people have speculated about their possibilities. While their theoretical limitations were established early on, their practical implications have always been overrated. This is especially true in the domain of artificial intelligence. The name alone has raised expectations that could, certainly until now, not be fulfilled. The mere possibility has been at the centre of a lively philosophical discussion. Conceptually one distinguishes strong and weak definitions of artificial intelligence. A machine is artificially intelligent in the strong sense if it possesses consciousness in some form (Searle, 1980). It should be able to reason and solve problems, and be aware that it has this ability. Its behaviour however, need not be human-like. Weak artificial intelligence uses computers and software as tools to study the nature of human like intelligent behaviour. It is interesting to confront this discussion with the original purpose of computers as automates. They were designed to help people in performing tasks which would otherwise be far more difficult or even impossible. In this sense, artificial intelligence can be viewed as the science of developing tools that overtake tasks that would normally require significant mental efforts from a human intelligence. Successful realisations in this sense are ample and can be found in all subdomains of artificial intelligence (e.g. computer vision, machine learning, language understanding, automated reasoning,...). Despite these successes, specialised applications often exhibit limited functionality or performance, essentially contradicting the broad ambition inherent in its naming. In the nineties, computers became more mobile, connective, smaller, cheaper and pervasive. The possibilities on a single palm top are often not impressive from an artificial intelligence point of view. Through its ease of use and its connectivity however, it realises an environment that supports humans very efficiently in their every day tasks. In this setting, it is natural to identify autonomy as an important asset for a computer system. The palm top is communicating with its peers while the human is carrying it around. It can perform routine tasks like synchronisation of agendas or sending a file to a printer at the time when this is most appropriate, without intervention of the user. It can be location aware and adjust its behaviour accordingly. These and similar considerations lead to the paradigm of agent-based systems.

SOFTWARE AGENTS

According to Wooldridge (2001) five trends have governed evolutions in computer science: ubiquity, interconnection, intelligence, delegation and human-orientation. The last decades have seen a speed-up in this evolution. Computers became smaller, cheaper and more flexible in their designs and consequently their ubiquity has grown significantly. Computers influence our lives when we look up information or when we interact with the bank, but they also pervade our cars, our kitchens and can nowadays be found on the chip cards in our wallets. Advances in communication networks provide a high-level of connectivity to these small intelligent entities. Intelligence has grown as indicated in the previous paragraph, but also in the design of user interfaces and functionality. We trust computers sometimes more than humans to control the airplanes we fly in. It is clear that computers have become immensely more human-oriented since their invention but also since the time of the large mainframes.

In summary, computer systems have evolved into sets of interconnected autonomous units that are supposed to intelligently perform tasks on behalf of their users in order to create a comfortable environment allowing those users to concentrate on their core concerns. Where the environment in the previous examples was a real world situation in which computers are being carried around and which they can observe through their sensors, it can just as well be a network of interconnected systems or a software environment where autonomous software units perform a variety of tasks on behalf, but without direct control of their users. We have reached a point where users accept or even expect that the systems they use behave autonomously up to a certain level. These considerations lead to the concept of software agents. Autonomy is the central property of software agents. They are designed with a goal, a means to achieve this goal and, ways to observe the environment in which they live. The control of their actions is not explicit, but relies on an automated decision making system. From a software development point of view, they resemble software objects in that they hide implementation issues and can only be approached through well defined inter-
faces. But unlike objects, these interfaces must not be seen as a panel with buttons and switches hard wired to the methods of the object. Instead, agents accept messages to which they autonomously decide how to react. This decision making process may be straightforward and simple, or may be based on any device from artificial intelligence. In most agent models, information fed through the message interface is considered incomplete. Agent models are thus easily tailored to environments in which computing and communicating resources are limited or scarce. For this reason, and because the agents can decide which information to store or not, agent models tend to scale better. Because of their inherent autonomy and the message paradigm, they can easily be made mobile and can migrate from one computing device to another, or to a storage device, making them persistent. The agent design problem is to define behaviours that allow it to reach its goals in a non-deterministic environment.

Multi-agent systems make use of the agent properties to create societies of interacting agents. They are typically equipped with a communication language which they use according to certain protocols. Agents can play selfish or collaborative games. They will follow appropriate strategies to realise their goals within the society. The multi-agent design problem is to define the protocols and agent goals in such a way so that the agent society realises the multi-agent system goal.

Within the field, a major theme of discussion has been the complexity of the agents. One school advocates intelligent, formally reasoning agents that apply high-level artificial intelligence decision making (Dunne et al., 2004). Another party rejects this approach on the basis of both performance reasons and the complexity of the modelling task (Dorigo et al., 2004). While the first school wants to build upon classical results of a mature research field, the second argues that the unpredictability in the environment does not allow a formal approach. Instead, they propose agents with a very simple and fast reactive behaviour. As so often, practitioners are somewhere in the middle and have to select the best from both sides in order to match their particular problem.

In conclusion of this section, it is important to mention one more property of agent based systems. As should have been evident from the above, agent models are designed to be realistic models of the world. When modelling a specific situation by agents, one will try to identify autonomous entities and map them to agents in the model. This is an attractive property for two reasons. Firstly, it allows defining the goals and the behaviours of agents very closely to the real world situation. Secondly, it makes the model recognisable for the users. The resulting system will be more transparent. One expects users to easily understand the way it functions. This has important consequences for the level of trust, safety, security and privacy. Since users can see how the decision making works, they can verify whether it satisfies their needs.

**ETHICAL AND PHILOSOPHICAL CONSIDERATIONS**

In an anthropological approach to philosophy of technology, technology develops as an extension or completion of human organs or skills, or as a compensation of human deficiencies. Aspects of human competence or behaviour are said to be objectified in technological artefacts. Depending on the level of actions or possibilities, the technological objectifications differ. Inspired by the analysis of Coolen (1992), a typical distinction can be:

- **Instruments** can be seen as technological objectifications of movement; energy supply and control of the movement still are in the hands of the human actor;
- **Machines** are instruments in which the energy supply has been incorporated; steering or control are still under the human dominance;
- **Automates** (automatic devices) are machines with the additional level of control incorporated;
- **Informates** find their origin when man reflects about his own thinking processes.

This style of philosophy of technology often comes to surface in ordinary language, where technical terms may be used for human actions or properties, or when human metaphors are used to describe technological artefacts. Clothes can be seen as a second skin, glasses may be referred to as an additional pair of eyes. A very clear example can be found in the comparison between human thinking and computers where the studying activity of students can be described as storing the information on their hard disk.

There is discussion between philosophers about the logical connection between these two registers: some say that technological objects are the results of human reflection upon their own being and acting. Others prefer the reverse: man could better understand himself and his actions by studying the operating mode of technical artefacts. Or at least the human metaphors, when used in technological contexts, tell us as much about humans as they do about technology.

The vagueness of the human/artefact distinction often makes people feel uneasy. Hence, for example, the fascination for literature and films in which human-like ro
bots appear, with the implicit question about the moral status of these artefacts. Whereas little or no people would object to the use of crutches or dental prostheses (compensating for the deficient walking or biting capabilities of some), the implantation of an electronic chip for deficient brain functioning is often seen as a threat to human identity. Yet, at least Coolen’s analysis is that, even when confronted with an artefact behaving in each and every way like a human, man will never experience artefacts as full equivalents to humans.

That these ways of objectification are experienced differently, may have to do with the feeling of insight and control one has about one’s physical actions on the one hand (objectified in instruments, machines, and - to a certain extent - automates), and one’s mental functioning on the other hand. Seeing the movements of an excavator as an objectified equivalent of what a ground worker does with his shovel is quite easy (maybe also because it can be easily visualised). Hearing the information treating process of a computer being compared to human thinking may have an alienating effect when the human comprehension of the thinking process (either individually or for humans in general, including psycho-analytical insights about the role of unconscious or subconscious factors) is not self-evident. Man may feel outdistanced by this technological evolution.

With agent technology, an additional level of objectification seems to have been reached. Software agents do more than merely treat information. They seem to be an objectification of human interaction and negotiation, taking into account needs and interests, preferences and aversions, even attitudes such as sympathy and trust (De Causmaecker et al., 2000).

Whereas the rational structure of ordinary reasoning and thinking is not always very transparent, the sphere of these needs, preferences and attitudes may be even less transparent - things becoming even more complicated by the factor of interaction with other participants (and their agents). If moreover software agents are entrusted with some kind of autonomy, the alienation effect which popped up at the previous level of objectification, may become even stronger. Instruments and machines could be considered as extensions of human limbs or organs, suggesting that (like limbs) they are under human control. The more autonomous the technical artefacts appear, the more questions of trust and distrust in technology become preponderant.

*To design trust* in software agents is then important on two levels: first, technology must gain trust from users; and second, trust as an interhuman attitude must be transferred to the representing agents.

The decision to confide in technology may be enhanced by designing and presenting agent technology as a mere reflection of *the real word* and of human interactions therein. The description of the communication platform further on in this paper, follows this model. The transfer of interhuman trust to software design in itself, is treated in the next section.

For the time being, the metaphor seems to work in one direction only. The design of the communication platform described later on in this paper, attempts to reflect the *real world of multimodal transport* as faithfully as possible.

**TRUST**

A sole guarantee of technological trustworthiness, if that is feasible, will presumably not convince enough people to make any agent based system a success. A trusted system is one where legitimate rights are implemented (Whitworth and De Moor, 2003). Online applications require extra efforts because perceived security and other psychological factors are much more important (Castelfranchi and Pedone, 1999). The perception of trust is strongly influenced by reputation or by the knowledge of past behaviour. Also, trust can be stimulated by the presence of common goals (Axelrod, 1984).

When it comes to trusting computer systems:

- other aspects than personal trust in a partner are required (e.g. reliable environment, reliable authorities);
- technical security is not sufficient;
- the traditional organisation of interaction needs to be revised.

In De Wever et al. (2001), trust is defined as, “The willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to thetrustor, irrespective of the ability to monitor or control that other party.”

Kracher et al. (2005) point out directions for modelling online trust in e-commerce. Their opinion is based on an extensive review of literature on trust, collected in the fields of philosophy, psychology, sociology, management and marketing. Frameworks exist through which an understanding of the complexity of online trust can be explored but the main conclusion of the review is that online trust should be based upon the large body of research results on offline trust.

Trust is essential in building up relationships. According to Hudson (2004), it is seen as important in promoting adaptive forms such as network relations, reducing
harmful conflict, decreasing transaction costs and promoting effective responses to crises. Trust also involves risk and vulnerability. Things would be different if we had complete knowledge of other parties’ intentions and future actions. Many parties are interdependent but they should all have free power of decision. If that is not the case, trust is not an issue.

Disclosing delicate information to a natural person can be conceived differently from consigning the same information to a software system. Providing information can serve different purposes. Either the intention is to only commit information or to make information available for further actions. In the first case, the system should guard over privacy whereas in the latter, the system should be reliable. In some cases, a software system will come across as more trustworthy (e.g. saving files on a hard disk) whereas when a professional or an advisor is required, a natural person can be preferred.

Software agents have been designed to defend the interests of the offline parties that they represent. In order to do that, they must know the objectives, the preferences, the preconditions, etc. of these actors, and of course behave tactfully. Since the developers have no reason to impose any method upon the users, all parties will be invited to configure their representative agent in the system, without necessarily involving an intermediary. It is therefore important that the parties involved gain a clear insight into their own way of working and into the operation of the agent based system. Eventually, they will need to consign delicate information to the software agent, which is no longer an impersonal information system, but behaves more like a professional advisor or a human assistant.

Among the factors influencing mutual trust among humans, we distinguish at least four elements: context, reputation, good and bad experiences and hypotheses about the other’s mental state. The context consists in the background against which a person meets another person (e.g. a tennis player gets hurt during a tournament. The individual’s usual doctor is not available, so he/she has to rely on the tournaments’ doctor). The reputation is the knowledge about the person from information by third parties (e.g. this doctor has successfully treated other athletes). Good and bad experiences build-up or break-down the relationship after it has been established (e.g. the doctor does not seem to recognise the medication the tennis player usually takes for this kind of injuries). The hypotheses about the other’s mental state deal with the estimation of the other’s intentions and interests (e.g. the tennis player knows that the doctor would like to be selected as a member of the medical staff for the next Olympic Games).

The designers’ challenge is to build agents which, given the available resources, implement a trust model matching human trust as closely as possible. In principle, agents can, up to a certain level, implement each of the four factors by relying on developments in specific fields of computer science. But even if interhuman trust can to a certain extent be modelled by agent technology, the agent system will never beat the real thing: the relationship between humans is far more complex and involves far more factors than our simple model does. On the other hand, agents can process larger amounts of data than humans, leading to a higher resolution picture of the other. These two compensating properties are similar to the situation in automatic boardgame where humans are often better than computers in the judgement of a specific position, which is compensated by the computer’s power to process millions of positions in a deep search tree.

This last property, on the other hand, may be an additional asset for gaining the possible users’ collaboration. Still, the acceptance of a software agent system by the possible users may depend on the same four factors mentioned above: the context in which it finds its origin, the reputation of the designers, good and bad experiences with the system (and with similar systems), and the way one views the intentions and interests of other possible users. Furthermore, user’s trust will be constructed by good and bad experiences: these are crucial and should show a healthy mix in which good outnumbers bad, sometimes leading towards a conservative design.

Well aware of the above-mentioned limitations, we will argue that implementing trust in software agents can be most easily addressed by a design mimicking the real world situation as closely as possible. The next paragraph describes the case study of a communication platform for multimodal transport, in which some of these aspects of trust will be modelled by agents. The developers of the communication platform are wary of users’ probable distrust of a software system to simplify interaction in the transport domain. We will elaborate on this discussion in the following sections.

A CASE STUDY

In this paper, we will illustrate the application of software agents to ensure trust and privacy by explaining the modelling of a communication platform for multimodal transport. The communication platform will be realised in the MamMoeT research project (http://ingeniieur.kahosl.be/projecten/mammoeet, 2005) supported by the Flemish government and by 10
Flemish transport companies and organisations. Its central location in Europe and its dense transport infrastructure make the northern part of Belgium a very attractive logistic junction. However, numerous economic and ecologic problems related to overloaded roads put pressure on the transport sector. A few years ago, the Flemish authorities started promoting the modal shift (from road to rail, to inland waterways or to any combination of alternative modes) in order to relieve the roads from excessive heavy traffic. A considerable effort has been put into the modal shift and many related projects were recently launched. The experience of most companies is situated merely within a single transport mode. Due to the changes in the method of working, the sector is looking for accurate information and appropriate decision support more than ever. On the one hand, in today’s practice:

1. supply and demand are nearly exclusively matched through personal contacts, mostly by telephone, and nearly exclusively within one single transport mode; little knowledge exists about alternative modes of transportation;
2. the transport sector currently lacks information about the position of freight, expected arrival times, etc. Although the technology is implemented and used in most road transportation and in maritime transport (where load tracing is compulsory), inland navigation is not yet traceable;
3. traffic jams, bad weather conditions, human mistakes, non-observance of agreements, etc. induce a cascade of delays through the logistic chain. There are no mechanisms to counterbalance such events.

On the other hand, neither of the logistic parties desires a completely open system in which all logistic information is visible to other parties or to competitors. If suspicions arise that the communication platform could falsify competition or even interfere with currently existing rules of trade, there will be reserve against collaboration and the initiative risks failure. The MamMoeT research project aims at providing an online decision support system to assist users and providers of logistic services in executing their business. In the complex and ever evolving domain of transport and logistics, the rationale for an intelligent communication platform is threefold. It should:

1. match supply and demand in the transport sector, avoid empty transport and assist all parties to accomplish cooperation,
2. enable traceability of freight and transport for loaders and transporters,
3. handle events, i.e. spread information or deal with the effect of unexpected events.

The most obvious way to model a confidence-building software solution, is to accurately implement the conventions of the real world. We opted for the agent paradigm as a model to pursue the real procedures. Intelligent agents support the real participants in the transport business by imitating the real business processes such as gathering information, negotiating and cooperating. The agents will provide decision support for all the users of the system. Fig. 1 is a schematic representation of the functioning of the MamMoeT communication platform. We distinguish two separate layers in the model: the lower part of the figure represents the real world with companies, organisations and authorities that are concerned through (multi-modal) transport. The upper part represents the communication platform that is occupied by software agents. Any offline party will have a software representative in the agent community. Requests for collaboration, information exchange, negotiations, and the execution of contracts can take place virtually on the MamMoeT software platform. Of course, the companies will expect complete reliability of their agents: they must scrupulously respect the conventions of the real world.

We will adopt Hudson’s basic ideas on trust (Hudson, 2004) as they are a foundation for cooperation and the basis for stability in the transport and logistic market. It follows that agents in MamMoeT:

- guarantee sufficient privacy surveillance (are aware that internal information can be public, semi-public, interchangeable or private),
- assure discretion concerning certain actions or initiatives,
- give preferential treatment to privileged partners,
- have the capacity to build relationships based on mutual trust.

**USERS’ CONCERNS**

Most logistic players have no objections to the modal shift but it takes time for implementations in practice to become visible. The main problem is that many manufacturers are not sufficiently informed to smoothly make the change from road transport to other or combined modes. Although research revealed that the need for automated decision support is indeed very high in the logistic sector (Vannieuwenhuyse et al., 2003), there is further reserve against widely accessible software systems because of very obvious reasons. We will highlight a few of the concerns in this section.
Currently, transport charges result from negotiations between provider and customer. Transport suppliers are wary of software systems threatening to become market places that could give rise to a price war. Any such system would most probably allow large carriers to undercut all the small transport companies. It is clear that none of the endangered companies would support such a software platform. They all prefer the prices to be the result of negotiations between provider and customer, and they would rather ignore such a platform instead of support it.

There is an example of a recent initiative to replace the phased out rotation system for inland navigation, in which the captains of barges were commissioned to transport tasks by turns. It was supposed to match demand and supply of cargo, making use of modern telecommunication and information technology. As a result, an increase in the traffic along the inland fairway network was expected. The resulting system does not interfere with negotiations regarding the price of services, which are a matter of discussion between the cargo owner and the transport company. However, the individual skippers have a profound distrust of such online support and, up until now, have not given the system a chance. The effort has not led to an operational prototype.

Another objection against a publicly available communication platform is the fear of loosing privacy. Tim Berners-Lee (2000), the inventor of the world wide web, stated that “No-one will take part in the new web-like way of working if they do not feel certain that private information will stay private”. Confidential information, although in general indispensable in the search for collaboration, should remain private. It will not be accepted that companies make their list of preferential partners or their bad experiences public. Also, numerous carriers have a one-man business, which gives them the freedom to organise their timetable for the day in accordance with their private life. An inland skipper, for example, can decide to moor in a city along the waterway and take a break for a couple of hours. The skipper does not really want any interference in his private life from any authority or customer.

Manufacturers too prevent their activities and strategy from being visible to the entire sector. If the load of any vessel can be obtained from the communication platform, competing companies could get indirect information about their competitors’ activities. Suppose, for example, that large amounts of car parts are transported along a canal. Any party involved in the car business could, from this knowledge, guess that the goods are shipped to X, the largest car manufacturer along that canal. It is up to X to decide whether the company wants such information to be public or not.

The system must, in addition, avoid giving undue preference to certain modes of transportation. We are aware that parties like rail and road transport companies are very reluctant to cooperate. Especially since the local authorities have recently made great efforts to promote inland navigation. The project is open to all parties concerned the common transport modes in Flanders. Still, the vast majority of enthusiastic collaborators come from the inland navigation sector. They probably expect a higher benefit than other, more computerised, modes of transport. Consequently, that leads to a strong cooperation which simplifies the access to real data from exactly that sector.

**DESIGNERS’ BURDEN**

The decision support tool we initially had in mind was meant to structurally improve multimodal transport. Close collaboration with transport companies and with software developers in that field, however, compelled us to slightly change the focus. It is an experience that we share with the developers of Hemacard (Dieng, 2001). The current design of MamMoeT (Fig. 1), with its analogy between the real and the online world, was eventually conceived to mainly support the end users by computerising and improving time-consuming tasks that are currently carried out by hand. Yet, limiting the model to current working practice and how that should be automated, risks creating false expectations.

The developers’ focus is on modelling and on technology, whereas the users, for which that is a side issue, expect the technology to impeccably serve their needs. We attempt to bridge that gap by organising communication between developers and non-technical end users through discussing graphical user interfaces. The look of buttons and menus carrying familiar terminology induces more confidence than valid conceptual models. Users will not feel disappointed by a system when the (graphical) user interface truthfully visualises the underlying model. The user interface thus makes the system more transparent.

When users think they understand how the agents behave, the problem of ensuring trust shifts. Instead of giving very detailed commands to a machine, users should entrust their agent with an objective and with all the information they can provide to help meet it. In that way, the agent model will inspire confidence in the software system.
Figure 1: Schematic overview of the intelligent communication platform
Before any software support is available, transporters and shippers have no overview of all the possible parties to collaborate with. The companies rely on the partial knowledge available to them. They have to make enquiries about where and when preferable partners are available. Of course, it is always possible to improve the operation by collecting more information but that would probably cost more time and money than it would save. The MamMoeT platform can handle much more information than any single user can process.

**AGENT MODEL**

De Wever’s definition of trust (see above) applies to MamMoeT. The vulnerable parties therein encompass all the transport companies or other users that entrust information to the system. The delicate actions refer to making contracts. According to De Wever et al. possible outcomes of the system include:

- access to the value-generating resources and capabilities of partners;
- leveraging its own capabilities with access to complementary capabilities of partners;
- exchange and imitation of organisational knowledge;
- absorbing extramural knowledge, taking advantage of the skills and resources of the other partner, or acquiring access to new markets or to other customers.

The Platform for Privacy Preferences Project (P3P), developed by the World Wide Web Consortium, is emerging as an industry standard providing a simple, automated way for users to gain more control over the use of personal information on Web sites they visit. PISA (Privacy Incorporate Software Agent) is a research project that attempts to build a privacy guardian for the electronic age (Korba et al., 2003).

In the MamMoeT project, we have adopted the following principles for addressing privacy. A user wants to exercise supervision over the information he put into the web (in our case the communication platform), as the information has to reflect what the user knows, what he thinks, and even how, he feels. Users who trust information to the platform should have the possibility to set their preferences at any time:

- retrieve, update, delete information from the web that the user had put into the web,
- (re)define for what purposes certain information may be used.

Techniques such as encryption, hashing, digital signatures, etc., help to achieve a specific level of trust. However, they require that all those concerned know each other before the communication starts. A slightly more flexible way is to appeal to a trusted third party (TTP). One can compare TTPs with (local) authorities that for example, grant identity cards, driver’s licences, etc. By using certificates signed by TTPs, parties that wish to cooperate can reach a specific level of trust. Even more problems arise when there are no trusted third parties available or when their assistance is not wanted. We adopted the following rule. In general, those who want to communicate have to expose something valuable in order to gain trust. Once trust is achieved, all the obstacles
are cleared away and the trusted parties can freely communicate. Let us describe this model in more detail.

**Trust model**

Each participating user can label personal data with figures (by configuring a software agent) taking into account that the higher the figure the higher the need for trust towards a possible partner. Parties that want to set up a trustful environment for communication on the MamMoeT platform, have to pass through the sequence of the trust model.

The model, which is schematically presented in Fig. 2 involves the following steps:

1. **Achieve ‘trust’**
   - Identify: is the other party really the party that it claims to be?
   - Qualify: set a value that corresponds to the level of trust the other party deserves. The figure can be based on earlier collaboration experiences (feelings of sympathy can influence the figure), context (peak periods or periods of low activity might affect the figure)
   - Certify: make agreements concerning the encryption required to set up a confidential environment.

   Start the negotiation/communication/cooperation.

2. - Only information labelled with a value up until the qualification of trust can be used. Say that A issues the figure 6 to B. During negotiations with B, A will not make use of information labelled with a figure higher than 6.

Every user who wants to participate in the MamMoeT communication platform can set up a personal profile. All configuration information concerning how other users of the platform can use the personal data are held within that profile.

Reciprocally trusted users can form a user group or community within which information can be exchanged freely up to a certain level. We call these communities **closed user groups**.

**Closed user group**

Fukuyama (1995) states that a society’s strength is strongly influenced by the level of trust or cooperative behaviour based upon shared norms. He thus distinguishes between high-trust societies and low-trust societies. Innovative ideas arise much easier in high-trust societies and there is a positive influence on the organisation of business. Low-trust societies on the contrary, tend to counteract initiatives by imposing and regulations and by controlling. It is interesting to reflect whether we should model the communication platform as a high-trust or a low-trust environment. The arguments above seem to advocate the high-trust model
provided that certain conventions are believed to be respected. Based on that idea, we propose the closed user group model that is illustrated in Fig. 3, for enabling more flexible processes within groups that have reached a certain level of trust.

When communication between more than two parties is required, a virtual community can be set up. This is a very natural way of projecting the real world situation, in which partners meet to inform each other and to discuss business, on an agent community. Since offline trust is evolving, the model should be flexible. It must meet the fluctuating desires of the parties involved.

If a party wants to join a community it has to check-in through a doorkeeper. The latter will keep out unintended parties wanting to join. Within a Closed User Group (CUG), users can freely communicate without any restrictions. CUGs were for example used in systems like Teleroute (an online freight and vehicle exchange program).

CONCLUSION

Trust always implies a moment of surrender: knowingly admitting that other parties take partial control of a situation in which one is vulnerable.

People have learned to trust traditional software products. The integrity of stand alone applications is only rarely subject to doubts. When it comes to distributed applications, where the communication element is essential, people tend to feel less easy: where does the information that I supply go and is the information I see on the screen to be trusted? When a software agent is offered as a representative, a user may feel more confident. This agent inherits the trust of the stand alone application because all communications with the user is local. At that time, the user controls its behaviour and settings in order to make it known how to behave loyally when the user is not available for advice. Agents in this sense allow the exploitation of facilities for communication between people and strengthen the performance of a society while leaving people in full control.

Any software system needs to be trusted in order to be accepted. Yet trust becomes more important even when the system has an online character and supports interconnection between heterogeneous and independent parties. It asks for guarantees on privacy, authentication and security. When the system starts behaving more autonomously, be it on behalf of its users, the level of objectification rises, and hence also the risk of alienation. Trust requirements then reach another level. Users have the right to know and be able to check how the information they supply is used or is extracted. A first step is to build the system as a true representation of the real world situation, eventually mirrored in the graphical user interface. It makes its actions recognisable. While implementing agent technology for communication, negotiation and eventually decision taking situations, the presentation of the software agent as an alter ego of the real world participants, may help gaining trust of the participants in so far as these participants can themselves configure the information, objectives and other characteristics of their agent. Yet this does not solve the problem completely: the question of trusting the informational environment in which the agents will operate, remains to be taken care of. In the development of MamMoeT, this is one of the aspects requiring continuous attention.

Further measures will be needed to gain the full confidence of the users. It is natural for them to ask whether the system is actually built and implemented as it was designed. In other words, do the agents really behave as they were intended to. Again, part of these concerns can be brought down to accepted practice for checking software components. As the agents have autonomy however, other procedures may be required. One probably needs online guardians that can detect and signal unwanted behaviours (e.g. the benevolent and malicious agents (Meyer, 2003)). The more complex and abstract level at which computerised tools are operated, the more it requires complex and abstract ways to deserve users’ trust. By building the instrument as a mapping of the real world (as truthfully as possible) we can achieve the essential first step in this direction.

Acknowledgement

The MamMoeT research project is funded by IWT (TETRA 40191).

REFERENCES


Trust in interorganisational networks: towards a conceptual model Workshop on Trust within and between organisations, EIASM, Amsterdam


