A Real-World Approach to Secure and Trusted Negotiation in MASs

Roberto Bianchi, Andrea Fontana, Federico Bergenti
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
Parco Area delle Scienze 181/A, 43100 Parma, Italy
{rbianchi,afontana,bergenti}@ce.unipr.it

ABSTRACT

The problem of providing tools to support legally valid negotiations between agents is becoming more and more critical. Agents are supposed to perform crucial tasks autonomously; however, they cannot exploit an extensive set of laws since the development of a full legal corpus for the computer world is yet to come. In this work we present an innovative model of interaction between agents that leads to an increase in the level of trust in negotiation-intensive MASs. In particular, we address some common problems related to trust and security in real-world negotiations and outline a set of abstractions that we can use to increase the level of trust that we can expect from agreements with third parties.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Legal Aspects, Security

Keywords

Security, Privacy, Trust, Multiagent Systems

1. INTRODUCTION

Agent technology is quickly evolving towards the realization of complex societies of agents. Just to cite one recent example, the aims and scope of the IST project CASCOM [1] show how agents are becoming more and more relevant in important sectors, e.g., healthcare and personal data management. This evolution is not yet matched by an equivalent legal development. The lack of a legal substrate capable of grounding the interactions between agents ultimately means that every aspect of negotiations must be explicitly treated by the developer. Furthermore, if we cannot guarantee traceability [4] of the operations of individual agents, no law would be sufficient to prevent and punish mendacious agents.

Our work tackles some important aspects related to security and trust [3] in real-world MASs by describing an innovative approach to the realization of negotiation-intensive MASs that allows agents to: *(i)* Negotiate in a secure and traceable way; and *(ii)* Guarantee the desired level of security and trust exploiting the minimum possible number of trusted parties. This is done through the introduction of two closely-related abstractions: *Validation-Oriented Ontologies* (*VOOs*) and *Guarantors*.

2. VOOS AND GUARANTORS

The first assumption that we take in the description of our model is that proposals and agreements between negotiating parties are exchanged in the form of individuals of known ontologies. This assumption allows agents to manage the information contained in proposals and agreements in a friendly way, e.g., to reason about proposals and to assert the formal validity of proposals against the constraints of the ontology. This assumption leads immediately to two general problems: *trusting ontologies* and *trusting identities*.

Problem 1. Trusting Ontologies: Ontologies seem to be a suitable means for describing agreements, but any attempt to use them in real-world scenarios immediately encounters a problem: How an agent could trust a new ontology? Could an agent (in some sense) validate the ontology to decide whether to trust it or not?

Moreover, an ontology may be partially non-disclosed because, e.g., it contains some marketing strategies of a seller. In this case, a full fledged reasoning on the ontology could be done only by accessing the whole ontology, and only partial reasoning is possible for third parties. In addition, we have to take into account a third (very serious) facet of this problem: there is no way to validate the adherence of the ontology to real-world laws without involving highly specialized jurists.

In the end, it should be quite clear that trust cannot be given to an ontology per se: it must be accorded to its signer. Ontologies used to model formal agreements and contracts must be provided by trusted and liable signers.

Problem 2. Trusting Identities: The ultimate aim of our model of interaction is to guarantee legal validity. Therefore, the problem of checking the identities of involved agents is obviously critical. Unfortunately, a simple static control of identities by means of certificates [2] is inadequate because, e.g., certificates can be revoked or keys can be stolen.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

From this simple consideration, it should be quite clear that the identification of agents in a secure and trusted MAS can be performed only through a set of runtime tools capable of validating certificates, and thus realizing a trusted source of identification.

The problem of checking identities is closely related to the representation of identities. In our model, we decided to design an ontology describing legal persons and their attributes and to associate this ontology with a set of general-purpose tools for addressing the majority of problems related to identification.

2.1 VOOs

The two very general problems outlined above are addressed by the introduction of *VOOs* and *Guarantors*.

A VOO as a signed set containing: *(i)* An ontology that models a domain; and *(ii)* A set of runtime tools capable of asserting properties of individuals of this ontology.

Runtime tools are intended to provide a means for validating assertions on the domain described by the ontology without requiring a full-fledged reasoning on the domain.

One important advantage of the introduction of VOOs is that they reduce the amount of distributed trust, since in a single signed object lay both the semantic description of thing and a set of related actions.

Moreover, VOOs promote software reuse and help standardization, since many ontology-related tasks are delegated to external bodies (the tools of the VOO) in a standard, well-defined, trusted and secure way.

VOOs are not sufficient to address all issues related to real-world agreements because we need to trust both the VOO and the signer of the VOO. In fact, if we go back to the human world, the proper way to stipulate contracts is through a notary public. This happens because only legal person trusted by the State can perform critical tasks (e.g., querying databases containing privacy-critical information). This is the reason why we introduce the abstraction of Guarantor, and we say that an agent is a Guarantor for an interaction between two other agents if it can sign a VOO that the two other agents can use in their interaction.

2.2 Guarantors

In our model, a Guarantor is responsible for the following tasks: *(i)* Provide identity certificates; *(ii)* Provide signed ontologies compliant with real-world laws; and *(iii)* Provide signed runtime tools for its ontologies and/or certifying external tools under its responsibility.

Then, if we remember that identity certificates are provided as signed instances of concepts of an ontology, and if we go back to the previous definition of VOO, these three responsibilities of the Guarantor can reduce to a single responsibility: *provide VOOs*.

The Guarantor takes the responsibility of catalyzing the trust of an interaction in various ways, e.g., through: *(i)* A signed list of trusted tools; *(ii)* A certified public key whose private key is provided only to trusted tools; and *(iii)* A certified set of APIs that could access the Guarantor's database and whose use could be detected by the tools' user.

Examples of Guarantors in the real world are States, notaries, and Municipalities. In the agent world, Guarantors are the agent counterparts of private organizations, e.g., Certificate Authorities, as well as public organizations, e.g., States or Municipalities.

Agreements are based on VOOs published by a Guarantor. If two agents trust different Guarantors, they will use different VOOs and therefore they will never reach an agreement. Since it is clear that a worldwide Guarantor is unrealistic and unfeasible, we require mutual recognition between Guarantors in multi-Guarantor agreements. Obviously this implies mutual knowledge and recognition of their public keys, but this is not sufficient. What happens if two Guarantors share their keys but not their ontologies? We need to express mutual recognition between Guarantors as a sharing of their respective VOOs.

Mutual recognition works both for hierarchical and non-hierarchical models: in the first, shared VOOs are defined and signed by the upper-level Guarantor, while in the second, shared ontologies are negotiated between Guarantors and then jointly signed.

The multi-Guarantor model reduces to single-Guarantor model only if there is a collective recognition, i.e., all Guarantors share and sign a common set of VOOs.

3. CONCLUSION

The central focus of this work is on the motivated introduction of two abstractions, VOOs and Guarantors, that we can use to provide general-purpose mechanisms to realize secure and trusted MASs. The introduction of these abstractions has two interesting properties:

Property 1. Concentrated Trust: Guarantors are sorts of trust catalysts that we use to keep trust concentrated on the minimum number of parties. From the point of view of interacting agents, this is good because the number of operations related to according or revoking trust is minimized.

Property 2. Pragmatic Interactions: The strict coupling between an ontology and a set of tools capable of performing general-purpose, critical tasks on the individuals of this ontology (i.e., the idea of VOO) guarantees the possibility of performing secure and trusted interactions also to agents with minimal reasoning capabilities.

In conclusion, we believe that the introduction of VOOs and Guarantors provides a solid ground for the concrete development of trusted and secure MASs. Many issues related to these properties are encapsulated by these abstractions and we believe that their in-depth study can lead to a better understanding of the subtle behaviours of these complex systems in real-world situations.

Acknowledgements

This work is partially supported by project CASCOM (FP6-2003-IST-2/511632). This article reports on joint work that is being realised by the consortium. The authors would like to thank all partners for their contributions.

4. REFERENCES

[1] CASCOM Web site `http://www.ist-cascom.org`

[2] Ellison, C., *SPKI Requirements*. IETF RFC 2692, 1999.

[3] Gambetta, D. (Ed.), *Trust: Making and Breaking Co-operative Relations*, Basil Blackwell, 1985.

[4] Szomszor, M., and Moreau, L., *Recording and reasoning over data provenance in web and grid services*, in Procs. of ODBASE'03, LNCS, 2003.