Automated eContract Negotiation in Web Service Environment: Trust Management Aspects

Marius Šaučiūnas

Vilnius University, Institute of Mathematics and Informatics, PhD student
Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Akademijos Str. 4, LT-08412 Vilnius
E-mail: m.sauciunas@gmail.com

Albertas Čaplinskas

Vilnius University, Institute of Mathematics and Informatics, Principal Researcher
Vilniaus universiteto Matematikos ir informatikos instituto vyriausiasis mokslo darbuotojas
Akademijos Str. 4, LT-08412 Vilnius
E-mail: albertas.caplinskas@mii.vu.lt

The paper addresses trust management problems in automated eContract negotiation among software agents in the web service environment. From the point of trust management, the aim of the negotiation process is to choose the most trustworthy providers from those who provide services that satisfy certain functional and other requirements. In order to negotiate about the trust, the negotiation process should provide some mechanisms to reason about requesters' policies, specifying who and under what conditions nay access private information (Kagal et al., 2004) and to guarantee that no legal norms would be violated in the contract. The paper familiarizes with details of the trust negotiation problem and with the approaches that have been proposed to solve this problem. It presents also a critical analysis of the proposed approaches and summarizes their challenges and drawbacks. The author analyses also one of the more advanced conceptual frameworks of negotiation process from the trust modelling perspective, highlights its drawbacks and proposes how to improve this framework.

Introduction

The object of this paper is a critical analysis of the automated trust negotiation process among software agents in the web service environment. At present, the whole contract lifecycle in eBusiness, including negotiation, the preparation of eContract and its acceptation, is handled mainly manually. In order to develop an electronic contract, people should not only write and agree upon it, but also to translate it manually into a certain computer-readable internal representation (Hasselmeyer et al., 2006). Automated negotiation and the usage of eContracts is still is a challenge.

Automated negotiation is especially important in the dynamic environments in which short-time contracts prevail. Such contracts have to be dynamically set to meet the short term needs of end-users' and service providers'. In such circumstances, it is impossible to rely on the partners' trustiness characteristics because they are not tested by a long-time experience or are even completely unknown. According to Comuzzi et al. (2005), in dynamic environments, negotiation seems to be the most suitable mechanism to agree on the features of a dynamic contract.

The goal of the paper is to discuss the state of the art of the automated trust negotiation problem, to highlight the challenges and drawbacks of the proposed solutions, and to contrast the conceptual modelling problems of trust management aspects with the current negotiation process modelling concepts. The main contribution of the paper is the proposal how to improve one of the more advanced negotiation process object-oriented modelling frameworks (Lin, 2008).

The rest of the paper is organized as follows. Section 2 familiarizes with details of the trust negotiation problem, Section 3 surveys the current state of the art of the proposed solutions, Section 4 analyses the concepts used to model the trust negotiation process and proposes how to improve Lin's conceptual modelling framework and, finally, Section 5 concludes the work.

Trust negotiation problem

The trust negotiation problem arises in the context of eContracting. One of the most important requirements is that in the semantic web environments the eContracts should be prepared automatically, without human intervention. Contracts should be prepared by negotiation of software agents. Trust is one of the negotiated issues. Traditional security mechanisms, which assume that parties are known to each other and that trust can be granted only on the basis of partner identity, are insufficient in the Semantic Web environment. To identify the requester in this environment, the provider requires additional information sufficient for him to make the access permission decision. On the other hand, the requester wants to restrict the conditions under which his personal information will be automatically disclosed. In some cases, even requirements to be fulfilled between parties cannot be publicly disclosed. In such cases, parties can disclose the confidential information (e.g., credentials or sensitive business rules) to each other iteratively only, by negotiation, at each step increasing the level of trust. To specify the information associated with parties and the requirements to be fulfilled, an agent-understandable language with well-defined semantics is required. In addition, the negotiation mechanisms should be semantically enriched so that the required authorisation process would be supported; the illegal disclosure of information would be not possible; the access to sensitive resources would be controlled, and the trust between contracting parties would be ensured (Bonatti, Olmedilla, 2007).

From the latter requirement it follows that the trust management problem in the context of eContracting is essential because many different aspects, including the implementation of trust relationship among the parties, choosing the relevant trust model should be considered in order to solve this problem. In more detail these issues will be discussed in the next section.

The critical analysis of the proposed solutions

Authorization and privacy for Semantic Web Services

A significant amount of research has been done in trust negotiation for Semantic Web Services. Such services should be discovered and invoked automatically. The interaction with services is also performed automatically, and the decision which information has to be exchanged needs to be autonomous. To meet these requirements, Semantic Web services should handle users' private information that has to be protected, autonomously decide who can access it and under what conditions. Policies, as part of Web Service representations, could be used for this purpose (Kagal et al., 2004). In addition, Web Services should know how to reason about their users' policies. The main role of policies is to specify who can use service and under what conditions, and to define the information handling rules. However, the notion of policy is not unambiguous. Policies are used also for other aims. For example, security policies constrain access to some resources, and trust management policies are used to collect agent properties in open environments. Policies can also define business rules, formalize and automate business decisions. Besides, policies can be reactive, include actions to collect information about events (e.g., event logging) and have some side effects. On the other hand, all kinds of policies share common information and tightly interact with each other (Bonatti et al., 2005).

Privacy and authorization policies have been proposed by Kagal et al. (2004). They determine under what conditions the information can be exchanged, what usage of this information is legitimate. They also constrain the provider to accept

requests for service only from certain requesters. The use of these policies is symmetric; they constrain both the provider and the requester. It is assumed that the requester and the provider discover each other's policies during negotiation on the contract. Afterwards, they must decide whether they can satisfy each other's requirements. The privacy policies can be interpreted as a contractual obligation. If some partner provides details of another partner to a third party, the person represented by the injured partner could take a legal action against the guilty partner on the basis of the policy. This approach was developed for the client–server architecture, but it can be easily extended also for service–service interactions. Some authors of the work Kagal et al. (2004) were also involved in the development of the OWL-S (Martin et al., 2004) specification and for this reason are quite familiar with the details of OWL-S. So Kagal et al. (2004) proposed the so-called semantic markup that specifies the security characteristics of Web Services' I/O parameters in OWL-S, "keeping information about the data's structure but without revealing its value" (Kagal et al., 2004). This provides the basis for determining whether a service parameter fits a requester's requirements and whether the two services' I/O parameters match. The Rei language (Kagal, 2002) is used to describe such policies. The Rei language is based on the first-order logic and includes an RDF interface based on a given ontology (McBride et al., 2004). In this language, the deontic concepts of rights, permissions, obligations, dispensations, and policy rules are represented as the Prolog predicates. The Rei framework provides for the policy engine that reasons about the policy specifications. The OWL-S Matchmaker acts as a service discovery agency, takes the OWL-S description of a service that matches the requester's functional requirements, extract this service, retrieves the requester's policies and extracts the policies from the provider's profile, and sends the OWL-S description and the policies to the Rei reasoning engine which reasons about the compatibility of the partners. If the policies are not compatible, the reasoning engine returns the value as false, and the Matchmaker continues to check the next service for compatibility. Otherwise, the reasoning engine returns the value as true, and the Matchmaker returns this service to the requester. Although the Rei framework has many advantages, it has also some serious drawbacks. Firstly, it assumes that all policies are public and that a policy engine or a matchmaker decides in a single evaluation step whether two policies are compatible or not (Bonatti, Olmedilla, 2007). However, in some scenarios, the sensitive policies should be protected and disclosed iteratively by negotiation. Secondly, it assumes that both the requester and the provider trust the Matchmaker and will disclose to it all policies, including sensitive ones. However, in the decentralised or multi-centre environments, such assumption cannot be accepted. Since requesters interact with the services unknown to them, they are not sure whether they can trust them. Consequently, some means should be provided to achieve trust. According to De Coi, Olmedilla (2008), in such environments even the access control based on identity mechanisms may be ineffective and should be replaced by role-based ones (Herzberg et al., 2000).

Role-based mechanisms

Access control, which uses role-based mechanisms (Herzberg et al., 2000), splits the authorization process into two steps – assignment of roles and checking whether a member of the assigned role(s) is allowed to perform the requested action. For this purpose, some access control policy is usually used. Such a policy consists of the rules that specify what roles must be satisfied that sensitive information could be shared. To prove its role, the party must present the signed credentials that represent a statement by some authority that the party performs a particular role. The credentials could be exchanged with different granularity (Nejdl et al., 2005). This means that some attributes unessential for the policy could be hided. The role memberships required to satisfy a policy are called provisions (Puchalski, Swarup, 2008). The drawback of provisions is that they must be known to both parties. However, each party may rep-

resent the same provisions in different ways. In decentralised or multicentre environments, this fact may cause the negotiation process to breakdown because of the schema matching problem. One more problem with provisions is that they do not provide any control of usage of information once it has been shared. Puchalski, Swarup (2008) propose to solve this problem by using the obligations together with the provisions and in such a way to specify the actions that must occur after disclosure of sensitive information. The authors propose also a trust negotiation model that includes support for both provisions and obligations. They model obligations as sets of actions in a bounded time range and extend a parsimonious automated trust negotiation strategy proposed by Winsborough et al. (2000) in such a way that obligations are used to replace provisions in cases when the necessary credentials are not available. The parsimonious strategy aims to achieve a successful negotiation with a minimal exchange of credentials. A drawback of this strategy is that the requirement to present signed credentials is insecure because sensitive information about the party's credentials, and maybe about other sensitive attributes, can be disclosed. The main challenge and open question of the strategy is how to minimize the disclosure of sensitive information, if it can be done in general.

Advanced policy languages

Advanced policy languages, for example, EPAL (Ashley et al., 2003), WSPL, and XACML, provide means to specify the mechanisms that make authorization decisions based directly on the properties of the requester and do not split the authorization process into two parts (De Coi, Olmedilla, 2008). Finally, the languages that have been developed to support the trust negotiation (Winsborough et al., 2000) ensure that "trust between peers is established by exchanging sets of credentials between them in a negotiation which may consist of several steps" (De Coi, Olmedilla, 2008).

The trust negotiation aspect for Semantic Web Services in advanced policy languages has been discussed also in many other papers written by a research group headed by Olmedilla (L3S Research Center and University of Hannover) and other researchers cooperating with this group. Olmedilla et al. (2004) suggest that the problem of trust negotiation can be solved including trust policies into the WSMO standard (De Bruijn et al., 2005), together with the information disclosure policies of the requester, by using the PeerTrust language (Gavriloaie et al., 2004) developed by the authors. This language provides the means to specify the trust negotiation and the delegation of authority. In this language, a policy is defined as "a rule that specifies in which conditions a resource (or another policy) might be disclosed to a requester" (Olmedilla et al., 2004). A service requester should include his/her policy in the request. Using this policy, the service discovery agency, or matchmaker in terms of Olmedilla et al. (2004), can compare it with service providers' policies and take into account the policies' compatibility. As a result, trust is established iteratively through the negotiation process. Such approach requires that the service discovery agency would have access to both the requester and the provider policies. This requirement impacts the architecture of the registry and the service discovery agency. The authors propose some distributed architecture allowing the service providers to keep their policies private, and an algorithm that matches the requester and provider policies in order to determine whether trust between them can be established. A drawback of this approach is that it does not provide any explicit reputation-based trust information, such as feedback from other trusty parties, service supply history, the quantities of delivered services, etc. It is important because the matchmaker in the service discovery process would take into account such information and compare the service providers by trustiness parameters if other requirements are satisfied. The same could be applicable to the service requester.

Reactive behaviour control

A number of trust-related policy languages, including PAPL (Bonatti, Samarati, 2000), PeerTrust (Gavriloaie et al., 2004), Ponder

(Damianou et al., 2001) and Protune (Bonatti et al. 2005), have been proposed. Some of them were created in line with the basic requirements for Semantic Web, such as simplicity, expressiveness, scalability, enforceability, analyzability (Blaze et al., 1998). However, early policy languages did not provide any means to specify policies controlling reactive behaviour, for example, when "decisions have to be made by taking events into account and consequences of decisions have to be turned into real actions" (Alferes et al., 2008). In addition, the reactive behaviour should take into account also the peculiarities of the distributed and heterogeneous environment where heterogeneity exists in languages and data. It means that the semantics of the policy language should define actions with respect to events, their sequences and flow control. An attempt to remove this drawback has been made by Alferes et al. (2008), who proposed a framework for the specification and enforcement of reactive Semantic Web policies. Similar frameworks have been proposed also by Bailey et al. (2005), May et al. (2005), Paschke et al. (2007). Typically, such frameworks implement the reactive behaviour control using the Event-Condition-Action Rules (ECA) that are neither able to model agent control in the process of negotiation nor some other particular interactions like delegation of authority or information disclosure, which are obligatory in the electronic contracting for Semantic Web Services. Thus, the ECA rules do not provide for any final solution of the problem of agent control in the electronic contracting, either.

Some time ago, an attempt to solve this problem has been made by Bonatti et al. (2010) who extend the concept of reactive Semantic Web policies in such a way that applying such policies a trusted communication among the agents would be ensured. The policies ensure that changes of knowledge stored in some Semantic Web database cause appropriate actions in the real world. Since the policies have the form of ECA rules in which, *inter alia*, the guarding condition provides for security checks that may be carried out only by trust negotiation, the policy-compliant communication should be trusted. SLD (Selective Linear Definite) derivation is used to evaluate the conditions. To this end, an explicit trust information exposed on the Semantic Web by other trusty parties is used. In order to access external semantic data, the policy definition language offers a special kind of predicates, the in-predicate "that allows calls to external methods to be integrated into the policy evaluation process" (Bonatti et al., 2010). In addition to a language to describe reactive Semantic Web policies, Bonatti et al. propose also some policy-compliant negotiation protocol. This protocol provides for "obeying as well as enforcing Semantic Web policies, automated agreement with other systems and trusted interactions with Semantic Web agents" (Bonatti et al., 2010). The details of the negotiation model are not described. The Reactive Semantic Web policies define in a declarative way the behaviour control of agents and combine reactive behaviour control with the trust and security features.

The main features of the proposed framework are as follows:

- well-defined semantics of the proposed language;
- seamless integration of Semantic Web sources into the reasoning process;
- support for trust negotiation;
- the possibility to use in the negotiation process both strong (e.g., digital signature) and lightweight (e.g., user name and password) evidences.

Reactive language may be based on different programming paradigms (Berstel et al., 2007). For example, production rules can be used instead of ECA rules. While the execution of the ECA rules system is event-driven, the execution of the production rules system is state-driven. Thus, the ECA rules ensure a more detailed behaviour control than does the production rule, but designing this kind of system is often more expensive than of state-based one. Besides, in some systems, state handling could be more important than event handling. Therefore, the choice of the paradigm depends on the system

under design. To make the right choice, it is important to know the kind of application the rules are suited for. The ECA rules are recommended for a distributed application in which the demand to operate with events exists. Meanwhile, production rules should be used in logically rich applications in which the demand to manage the state of the system in each web node is more important than to manage the distributed aspects of the whole system. A drawback of the ECA rules is that it is unclear how the natural bottom-up evaluation schema of ECA rules should be integrated with the top-down evaluation adopted by the policy languages (Bonatti, Olmedilla, 2007).

The proposed policy languages differ also by other properties (De Coi, Olmedilla, 2008), for example, what is the underlying formalism, how well the semantics of the language is defined, monotonicity of the language, expressiveness of conditions, what kind of actions can be specified within a policy, the means to describe the delegation of rights, supported evidences, to which degree the negotiation is supported, the kinds of answers sent by the policy engine to requesters, and extensibility. An exhaustive comparison of the current policy languages according to these criteria has been done by De Coi and Olmedilla (2008). According to them, only a few current policy languages, namely PAPL (Bonatti, Samarati, 2000), Cassandra (Becker, Sewell, 2004), PeerTrust (Gavriloaie et al., 2004), and Protune (Bonatti et al., 2005) directly support the negotiation.

Main challenges of the trust negotiation problem

A number of challenges still exist in the trust negotiation problem. The main challenges are as follows (Bonatti, Olmedilla, 2007; Bonatti et al., 2010):

- negotiation success: in which way to guarantee a successful result of negotiations in cases when some serious difficulties arise (e.g., rules are not disclosed because of the lack of trust; credentials cannot be found because their repository is unknown, etc.)?
- optimal negotiations: what strategies should be used to optimize information disclosure in the negotiation process? Is it possible to prevent not obligatory information disclosure by reasonable preconditions?
- choosing of service: how should the requester choose a particular service when the request can be fulfilled in several different ways? Both a language for expressing preferences and efficient optimization algorithms are required to solve this problem. Although the problem is more or less explicitly assumed by most of approaches on trust negotiation, so far no concrete solution has been proposed.

Additionally, as pointed out in the previous section, the integration of ECA rules is an open issue.

Conceptual modelling of negotiation process and trust management aspects

Lin's conceptual framework

Lin's conceptual framework (Lin, 2008) is one of widely accepted conceptual models of the negotiation process for web services contracting. He sees this process as a collaboration of three conceptual entities: the service requester, the service provider and the service discovery agency (Figure 1).

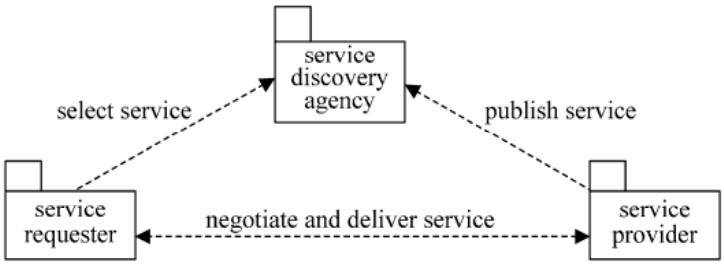

Figure 1. ***Lin's architecture for negotiating in a service-oriented environment (Lin, 2008)***

Each entity is defined by an UML package and further modelled by the use case, class, sequence, and package diagrams that define the internal architecture of the entity. The use-case diagrams define the goals of each entity and, consequently, all the use case diagrams together model the functional requirements of the negotiation system. Class diagrams specify classes that implement the entities. Sequence diagrams model the interactions of the objects participating in the negotiation. The Lin's conceptual framework provides neither the state machine nor activity diagrams. This means that neither the states of entities nor the methods of the classes are modelled. The main scheme of negotiation is as follows:

- the service requester asks the service discovery agency to find the required service and begin the negotiation process with it. If the request requires that a composition of services would be delivered, the requester "needs to maintain any relationships among these constituent requests for negotiating and issuing them in an adequate sequence and to deal also with the consequences from negotiating or issuing these requests" (Lin, 2008). The discovery agency can also maintain the list of preferred service providers and may predict the future needs of requesters, using the patterns of previous requests. So, it can prepare and sign contracts in advance. For this purpose, it maintains a contract template;
- in order to find the requested service, the service discovery agency maintains a registry of services. It negotiates with the service requester and as a result returns to its descriptions of accessible suitable services together with information about their providers;
- once the suitable service provider is discovered, the service requester negotiates with the service provider on the contract and signs it. The service requester evaluates also the quality of delivered services and updates the trust values in the list of preferred providers. The trust values are evaluations of service requesters to which extent the promises denoted in the corresponding contracts have been fulfilled by the provider;
- any service provider should register its service in the service discovery agency and must negotiate with it for this purpose.

The model provides for the service discovery protocol, service publishing protocol and service contracting protocol for negotiations between the service requester and the service discovery agency, between the service provider and the service discovery agency, and between the service requester and the service provider.

Analysis of Lin's conceptual framework from the trust management perspective

In order to negotiate about trust, the conceptual model should provide the mechanisms to reason about requesters' and providers' policies that determine who can access the sensitive information and under what conditions (Kagal et al., 2004). The model should also guarantee that no legal norms will be violated in the negotiated contracts. Lin's conceptual framework (Lin, 2008) does not provide any details how to do this. Most problematic issues are the way in which the framework models the service discovery agency, and the proposed negotiation protocol. The model assumes that the service discovery agency should be trusted by any party and that all parties would disclose to it all their policies, including sensitive ones. It is an obvious drawback from the trust management perspective. Another drawback is that the agency collects only the evaluations of service providers presented by the requesters (trust values in terms of the author). However, such trust values are insufficient to ensure trust between previously unknown parties. The proposed negotiation protocol does not address any trust negotiation issues, except trust values, and does not provide any mechanisms to extend it by trust negotiation strategies. Besides, it does not provide for any authorization decision process. Serious drawbacks from the trust management perspec-

tive are also the service discovery and publishing protocols which do not offer any rules how to take into account the peculiarities of reactive behaviour (events, event sequences, event flow control, etc.).

Also, Lin (Lin, 2008) does not discuss how the proposed model can be extended for the Semantic Web Services for which the trust requirements are even stronger because in this case the service discovery agency can determine at run time which actual previously unknown services should be employed to satisfy the requirements of a requester. Besides, the assumption that the service requester must maintain and negotiate all relationships among the parts of composite services as well as monitor and evaluate the quality of delivered services is not realistic.

Proposals on how to improve Lin's conceptual framework

To adapt Lin's conceptual framework to the needs of trust management issues, first of all it is necessary to remove the above drawbacks.

Modelling of service discovery agency. In order to change the assumption that the service discovery agency should be trusted by any party, the framework should provide for trust negotiation between service requesters and the agency and between service providers and the agency. This means that any sensitive information should be disclosed for the agency step-by-step in the process of negotiation only. In addition, some trustworthy authority, for example, VeriSign*, should issue signed credentials to the agency. The access control, provision and obligation policies should also be provided to manage the credentials' exchange process. This means that, in addition to the reputation-based trust, the conceptual framework should also provide for the policy-based trust mechanisms. To specify such mechanisms, some trust-related policy language should be used.

Once signed credentials have been exchanged, the further usage of the disclosed sensitive information should also be controlled. For this purpose, the eAgreement between any service requester and the agency and between any service provider and the agency should be signed, and these agreements should ensure legal sanctions for the disclosure of the protected information without permission.

Negotiation protocol. The negotiation protocol should be supplemented with negotiation strategies and incorporate an appropriate authorization decision process.

Service discovery and service publishing protocols. Service discovery and service publishing protocols should be extended by reactive Semantic Web policies that combine the reactive behaviour control with the trust and security management mechanisms.

Maintenance relationships among the parts of composite services, monitoring and evaluation of the quality of delivered services. The removal of this drawback requires that Lin's framework would be reworked fundamentally. The volume of this paper does not allow to discuss the required reworking in detail. The main idea is that the mechanisms similar to that provided by the JBoss (Jamae, Johnson, 2009) or Spring (Laddad, 2010) frameworks should be used for this purpose.

Extension of the framework for the Semantic Web. Advanced trust-related and reactive policies adapt Lin's conceptual framework to the requirements of the Semantic Web. The OWL-based ontologies to describe services should also be provided in the model. To ensure the processing of the semantic annotations in service discovery, the matchmaking algorithms should be changed.

Summary and Conclusions

In this study, a critical analysis of the automated eContract trust negotiation process among software agents in the web service environment has been performed. In such environment, the negotiation mechanism should support the authorisation process, control access to sensitive information, prevent its illegal disclosure, and ensure trust between the contracting parties. From the trust management perspective, several

* http://www.verisign.com/corporate/index.html?tid=footer

significant aspects, such as trust relationships among the parties or a relevant trust model have to be taken into account when dealing with the eContract negotiation problem. From this perspective, five major groups of approaches and mechanisms facilitating the trust negotiation problem can be identified: policy-based approaches, role-based mechanisms, trust negotiation models, reactive behaviour control, and trust-related policy languages. These groups do not represent some independent alternatives but rather complement one another in a hierarchical way. The drawbacks and challenges of each group have been discussed in the paper. Further, the object-oriented Lin's negotiation model (Lin, 2008), accepted by many researchers working in the automated negotiation field, has been evaluated from the trust negotiation perspective. Its shortcomings have been highlighted, and some significant improvements of the model have been proposed.

The critical analysis of the automated eContract negotiation problem demonstrates that a lot of different approaches and useful ideas have been proposed up to date. However, there is a lack of works to synthesize all these approaches and ideas and to recommend how to use the results of research in practice. A lot of experimental research should be done to this end. It intends to be a major focus of our further studies.

LITERATURE

ALFERES, J. J.; AMADOR, R.; KÄRGER, P.; OLMEDILLA, D. (2008). Towards reactive semantic web policies: advanced agent control for the semantic web [online]. In *7th International Semantic Web Conference Posters & Demos*, vol. 401CEUR-WS.org [cited 21 February 2011]. Available from: <http://centria.fct.unl.pt/~jja/page3/assets/iswcShort.pdf>.

ANDERSON, A. H. (2004). An introduction to the web services policy language (wspl). In *Policy* 2004. IEEE Computer Society.

ASHLEY, P.; HADA, S.; KARJOTH, G.; POWERS, C.; SCHUNTER, M. (2003). The Enterprise Privacy Authorization Language (EPAL) – How to Enforce Privacy throughout an Enterprise [online]. W3C, 2003 [cited 7 March 2011]. Available from: <http://www.w3.org/2003/p3p-ws/pp/ibm3.html>.

BAILEY, J.; BRY, F.; ECKERT, M.; PATRANJAN, P. L. (2005). Flavours of xchange, a rule-based reactive language for the (semantic) web [online]. In Asaf Adi, Suzette Stoutenburg, Said Tabet (eds.). *Rules and Rule Markup Languages for the Semantic Web*. LNCS 3791, Springer, p. 187–192 [cited 9 March 2011]. Available from: <http://www2.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2005-37.pdf>

BECKER, M. Y.; SEWELL, P. (2004). Cassandra: Distributed access control policies with tunable expressiveness. In *Policy* 2004. IEEE Computer Society.

BERSTEL, B.; BONNARD, P.; BRY, F.; ECKERT, M.; PATRANJAN, P. L. (2007). Reactive Rules on the Web [online]. In *Proceedings of Third International Summer School*, Dresden, Germany, September 3–7. LNCS 4636, Springer, p. 183–239 [cited 10 March 2011]. Available from: <http://www.pms.ifi.lmu.de/mitarbeiter/ehemalige/eckert/publications/ReasoningWeb2007.pdf>.

BLAZE, M.; FEIGENBAUM, J.; STRAUSS, M. (1998). Compliance checking in the policymaker trust management system. In *Financial Cryptography, Second International Conference,* vol. 1465 of Lecture Notes in Computer Science, Anguilla, British West Indies. Springer, p. 254–274.

BONATTI, P. A.; KARGER, P.; OLMEDILLA, D. (2010). Reactive policies for the semantic web [online]. In *Proccedings of 7th Extended Semantic Web Conference*, ESWC 2010, Heraklion, Crete, Greece, LNCS 6088. Springer, p. 76–90 [cited 12 January 2011]. Available from: <http://www.l3s.de/web/upload/documents/1/reactive_policies.pdf>.

BONATTI, P.; OLMEDILLA, D. (2005). Driving and monitoring provisional trust negotiation with metapolicies [online]. In *Proccedings of 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005), June 2005*. IEEE, 14–23 [cited 7 December 2010]. Available from: <http://www.l3s.de/~olmedilla/pub/2005/2005_policy_protune.pdf>

BONATTI, P. A.; OLMEDILLA, D. (2007). Rule-Based Policy Representation and Reasoning for the Semantic Web. In *Lecture Notes in Computer Science,* vol. 4636/2007, p. 240–268.

BONATTI, P. A.; SAMARATI, P. (2000). Regulating Service Access and Information Release on the Web. *J. Comput. Secur*, vol. 10(3), p. 134–143.

COMUZZI, M.; PERNICI, B. (2005). An Architecture for Flexible Web Service QoS Negotiation. In *Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference (EDOC'05)*, p. 70–82.

DAMIANOU, N.; DULAY, N.; LUPU, E.; SLOMAN, M. (2001). The Ponder Policy Specification Language. In *Proceedings of the Second IEEE Workshop on Policies for Distributed Systems and Networks* (POLICY 2001), Bristol, UK, 29–31 Jan. 2001. LNCS 1995. Springer, p. 18–38.

DE BRUIJN, J.; BUSSLER, C.; DOMINGUE, J.; FENSEL, M.; HEPP, D.; KELLER, U.; KIFER, M.; KONIG-RIES, B.; KOPECKY, J.; LARA, R.; LAUSEN, H.; OREN, E.; POLLERES, A.; ROMAN, D.; SCICLUNA, J.; STOLLBERG, M. (2005). Web Service Modeling Ontology (WSMO) [online]. World Wide Web Consortium (W3C) [cited 20 January 2011]. Available from: <http://www.w3.org/Submission/WSMO/>.

DE COI, J. L.; OLMEDILLA, D. (2008). A review of trust management, security and privacy policy languages. In *International Conference on Security and Cryptography SECRYPT 2008, INSTICC Press, July*, p. 483–490.

GAVRILOAIE, R.; NEJDL, W.; OLMEDILLA, D.; SEAMONS, K.; WINSLETT, M. (2004). No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web. In *Proceedings of 1st First European Semantic Web Symposium*, Heraklion, Greece, May 2004. LNCS 3053. Springer, p. 342–356.

HASSELMEYER, P.; QU, Ch.; SCHUBERT, L.; KOLLER, B.; WIEDER, Ph. (2006). Towards Autonomous Brokered SLA Negotiation [online]. In P. Cunningham, M. Cunningham (eds). *Exploiting the Knowledge Economy: Issues, Applications, Case Studies (eChallenges 2006),* Barcelona, Spain, October 2006. IOS Press, p. 44–51 [cited 17 November 2010]. Available from: <http://www.hasselmeyer.eu/pdf/echal06.pdf>

HERZBERG, A.; MASS, Y.; MICHAELI, J.; RAVID, Y.; NAOR, D. (2000). Access control meets public key infrastructure, or: Assigning roles to strangers. In *2000 IEEE Symposium on Security and Privacy*, IEEE Computer Society, p. 2–14.

JAMAE, J.; JOHNSON, P. (2009). *JBoss in Action*. Manning Publications.

KAGAL, L. (2002). Rei: A Policy Language for the Me-Centric Project [online]. *Technical Report HPL-2002-270,* HP Laboratories Palo Alto, September 30 [cited 7 January 2011]. Available from: <http://ebiquity.umbc.edu/_file_directory_/papers/57.pdf>

KAGAL, L.; FININ, T.; PAOLUCCI, M.; SRINIVASAN, N.; SYCARA, K.; DENKER, G. (2004). Authorization and privacy for semantic web services [online]. *IEEE Intelligent Systems*, July/August, p. 52–58 [cited 12 January 2011]. Available from: <http://www.ai.sri.com/daml/services/owl-s/pub-archive/AuthAndPrivForSWSIEEE.pdf>.

LADDAD, R. (2010). *AspectJ in action*. Enterprise AOP with Spring applications. 2nd edition. Manning Publications.

LIN, J. (2008). A conceptual model for negotiating in service-oriented environments. *Information Processing Letters*, vol. 108, issue 4, p. 192–203.

MARTIN, D.; BURSTEIN, M.; HOBBS, J.; LASSILA, O.; MCDERMOTT, D.; MCILRAITH, S.; NARAYANAN, S.; PAOLUCCI, M.; PARSIA, B.; PAYNE, T.; SIRIN, E.; SRINIVASAN, N.; SYCARA, K. (2004). OWL-S: Semantic Markup for Web Services [online]. World Wide Web Consortium (W3C) [cited 20 January 2011]. Available from: <http://www.w3.org/Submission/OWL-S/>.

MAY, W.; ALFERES, J.J.; AMADOR, R. (2005). Active rules in the semantic web: Dealing with language heterogeneity. In Asaf Adi, Suzette Stoutenburg, Said Tabet (eds.). *Rules and Rule Markup Languages for the Semantic Web.* LNCS 3791. Springer, p. 30–44.

MCBRIDE, B.; BRICKLEY, D.; MILLER, E.; RDF Core Working Group (2004). Resource Description Framework (RDF) [online]. World Wide Web Consortium (W3C) [cited 17 December 2010]. Available from: <http://www.w3.org/RDF/>.

NEJDL, W.; OLMEDILLA, D.; WINSLETT, M.; ZHANG, C. C. (2005). Ontology-Based Policy Specification and Management. In *2nd European Semantic Web Conference (ESWC)*, vol. 3532 of *Lecture Notes in Computer Science*, Heraklion, Crete, Greece. Springer, p. 290–302.

OLMEDILLA, D.; LARA, R.; POLLERES, A.; LAUSEN, H. (2004). Trust Negotiation for Semantic Web Services. In J. Cardoso, A. P. Sheth (eds.). *Semantic Web Services and Web Process Composition, First International Workshop,* SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers. LNCS 3387. Springer, p. 81–95.

PASCHKE, A.; KOZLENKOV, A.; BOLEY, H.; TABET, S.; KIFER, M.; DEAN, M. (2007). Reaction RuleML 0.2 [online]. RuleML Initiative, e-publication [cited 4 January 2011]. Available from: <http://ruleml.org/reaction/docs/ReactionRuleML-v0.2-Primer.pdf>.

PUCHALSKI, J. P.; SWARUP, V. (2008). Obligations in Trust Negotiation [online]. Technical paper. The MITRE Corporation [cited 15 January 2011]. Available from: <http://www.mitre.org/work/tech_papers/tech_papers_08/08_0191/08_0191.pdf>.

WINSBOROUGH, W.; SEAMONS, K.; JONES, V. (2000). Automated trust negotiation. In *Proccedings of the DARPA Information Survivability Conference and Exposition,* vol. I, p. 88–102.

PASITIKĖJIMO UŽTIKRINIMO METODAI AUTOMATINIU BŪDU SUDARANT ELEKTRONINES PASAULINIO SAITYNO PASLAUGŲ GAVIMO SUTARTIS

Marius Šaučiūnas, Albertas Čaplinskas

S a n t r a u k a

Straipsnyje aptariamos pasitikėjimo problemos, su kuriomis susiduriama derantis programiniams agentams dėl pasaulinio saityno paslaugų gavimo elektroninių sutarčių sudarymo. Derybų metu iš paslaugų teikėjų, teikiančių visus kitus reikalavimus tenkinančias paslaugas, reikia pasirinkti tokį, kuriam galima atskleisti savo konfidencialius duomenis. Pasitikėjimas turi būti abipusis, nes tai dažnai aktualu ir paslaugos teikėjui. Abipusis pasitikėjimas įgyjamas derybų būdu, palaipsniui vienas kitam atskleidžiant savo konfidencialius duomenis. Taigi derybos turi būti reglamentuojamos taisyklėmis, nustatančiomis, kokia informacija kokiomis sąlygomis gali būti atskleista. Straipsnyje analizuojamos abipusio pasitikėjimo įgijimo uždavinio detalės, svarbiausi to uždavinio sprendimo metodai. Išryškinami šių metodų pranašumai ir trūkumai, aptariami dar neišspręsti klausimai. Pasitikėjimo problemų požiūriu vertinamas vienas iš plačiai pripažintų derybų proceso koncepcinių modelių, parodyta, kad šis modelis neužtikrina pasitikėjimo problemų sprendimo, ir pasiūlyta, kaip jį tobulinti. Pagrindinė darbo išvada yra ta, jog šiuo metu aktualiau ne kurti naujus pasitikėjimo metodus, bet kritiškai ir eksperimentiškai analizuoti jau pasiūlytus metodus ir idėjas, juos apibendrinti, integruoti ir rengti rekomendacijas, kaip jais pasinaudoti praktikoje.