Inf Syst Front (2007) 9:181–194
DOI 10.1007/s10796-007-9031-x

From trading to eCommunity management: Responding to social and contractual challenges

Lea Kutvonen · Janne Metso · Sini Ruohomaa

Published online: 12 May 2007

Abstract The increasing pressure for enterprises to join into agile business networks is changing the requirements on the enterprise computing systems. The supporting infrastructure is increasingly required to provide common facilities and societal infrastructure services to support the lifecycle of loosely-coupled, eContract-governed business networks. The required facilities include selection of those autonomously administered business services that the enterprises are prepared to provide and use, contract negotiations, and furthermore, monitoring of the contracted behaviour with potential for breach management. The essential change is in the requirement of a clear mapping between business-level concepts and the automation support for them. Our work has focused on developing B2B middleware to address the above challenges; however, the architecture is not feasible without management facilities for trust-aware decisions for entering business networks and interacting within them. This paper discusses how trust-based decisions are supported and positioned in the B2B middleware.

Keywords Inter-enterprise collaborations · B2B middleware · Trust management · Interoperability

L. Kutvonen (✉) · J. Metso · S. Ruohomaa
Department of Computer Science, University of Helsinki,
P.O. Box 68, FI-00014, Finland
e-mail: Lea.Kutvonen@cs.helsinki.fi

1 Introduction

In the current trend, electronic business networks are built from autonomous business services. This trend can be seen in the use of Web Services (Booth et al. 2004), various consortia standards on inter-enterprise business process management (e.g., OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee 2002; Thatte et al. 2005), and in the rise of service-oriented architecture (SOA) (Papazoglou and Georgakopoulos 2003; Singh and Huhns 2005). It can also be seen in the number of eContract-related research projects in action (e.g., Chiu et al. 2005; Dellarocas and Klein 1999; Griffel et al. 1998; Daskalopulu 2002; Grosof and Poon 2003; Xu and Jeufeld 2003; Linington et al. 2004; Schoop et al. 2003; Angelov and Grefen 2003).

We call the collaborative, inter-enterprise business networks *eCommunities*. An eCommunity is dynamically established to serve a certain business scenario or opportunity and is governed by an electronic contract that is multilaterally negotiated. The contract, eContract, is structured by a business network model (BNM) that represents the selected business scenario in terms of roles for the business services involved, and required interactions between those roles. In the eContract, the actual role players are identified, and policy rules for the whole eCommunity are agreed at a more detailed level than the business network model can define.

To support this view, the Pilarcos architecture provides generic middleware services for inter-enterprise collaboration management (Kutvonen et al. 2005, 2007). Within this frame, the contractual aspects addressed range from information representation issues to technical and business aspects. Capturing the dependent elements from the business level and the technical level to the same eContract makes the Pilarcos solution differ from most eContracting proposals.

The management services include a number of pervasive functions as follows. First, tools and repositories support developing and publishing of new models for business networks, and defining new service types for business services in such a way that the service types match the needs of the business network roles (Ruokolainen and Kutvonen 2006). A service type defines common properties of a class of services in terms of the interface definitions, business protocols, and data semantics for properties such as communication and computing platform requirements of a service and other application-area-specific properties. Second, service offer repositories enable enterprises to publish business services to the open service markets together with metainformation as required by the denoted service type. This metainformation is later used for automated matching of services to roles and for interoperability testing against peers in the business network (Kutvonen et al. 2007). Third, means are required for declaring policies that govern the use and the availability of business services. Fourth, new protocols are needed for negotiating eContracts to govern a new business network (Kutvonen et al. 2005); the establishment phase is partially performed by a third-party population process, partially by a collective, refining or dropping-out negotiation protocol between becoming peers. Finally, facilities are needed for monitoring the behaviour within eCommunities and manage breaches within them as specified in the eContract (Metso and Kutvonen 2005).

For the pervasive services, there is a network management agent (NMA) for each enterprise, to represent the enterprise to the rest of the network and to serve as an interface to the external services, such as the common repositories.

We believe that by this kind of generic B2B middleware services that are available through private agents at each enterprise, the right kind of software investment cycles can be supported. The middleware services themselves are separated from the application software, thus making applications less dependent on the platform technologies. At the same time, the granularity of provided services grows to be understandable at the business strategies level; understanding the relationship between business services and the computational counterparts is a necessary requirement for controlling them (Kutvonen and Metso 2005). Furthermore, the development of B2B middleware and SOA-guided eContract-based architectures require the separation of various business and technical concerns in the contracting process, for example, security, trust and reputation, and business policies.

This paper elaborates on the business network establishment phase in which decisions on required interoperability are done and enhances it by addressing issues of trust management; it also discusses the operational time monitoring needs. The middleware agent that performs the establishment phase analysis is called the populator, and its task is to fill the different roles of a business network model with service offers of acceptable types, and to check that the selected services are able to interoperate. In the present situation, the importance of the populator lies in its ability to check interoperability conditions, but not in becoming an automated contract initiator with new partners from open service markets. The main hindrance in automated selection of partners is the lack of trust in unknown service providers and the lack of any framework contracts to govern the service markets. In the operational time environment, monitoring of contracted behaviour, adherence to enterprise policies, and managing breaches of trust are of importance. The monitoring results are to be used for feedback through the reputation system for more aggregated information in the later business network establishment events.

This paper discusses the effects and techniques of introducing trust-related decisions into eContracting. Trust is evaluated between peers, while the middleware layer should provide a trustworthy platform from which trustworthy information can be retrieved, and where trustworthy private agents are running. A secure communication infrastructure is assumed to be in place.

This paper is structured as follows. Section 2 discusses the social and contractual challenges addressed with dynamic collaborations formed from open service markets. Section 3 addresses the trusted role of the new infrastructure agents for providing an environment in which to manage these collaborations. The populator functionality and its implementation are further discussed in Section 4, while Section 5 introduces the trust concepts and discusses embedding trust considerations into the populator functionality. The monitoring methods and the effect of their usage is discussed in Section 6. We conclude with future challenges on research and standardisation on open business network management.

2 Addressing social and contractual needs by eContracts

Establishing new eCommunities from business services at the open markets raises problems that can be considered social. As the services involved are developed independently, there is no inherent knowledge for the intended business processes between partners, or knowledge of the competence of the potential partners. Thus, the interoperability demands between partners emerge to sharing external business processes, meeting on business value, understanding the pragmatics of enterprise policies, and furthermore, embedding management of trust between potential collaborators. This situation of autonomous domains with a need for federated management of collaboration relationships in a dynamic manner is very challenging.

The goal is to provide automated support for establishing new eCommunities, but in a controlled way. The automation should be limited to routine cases, and more vulnerable, new, or otherwise delicate decisions should enforce human decision-making. Although the automated part can be considered somewhat trusted, the autonomy of partners in the process still requires that privacy of decision-making (motivations, strategies, policies) should be protected by the overall architecture. The strengths of the automation should come from the management of routine configuration work that is dependent on the collaboration decisions, noted level of trust between partners, and available technological solutions.

From the business point of view, there are two contradictory requirements for making business services available. On one hand, it is preferable to have all potentially marketable services openly available for all potential clients and collaborators, while on the other hand, the integrity and privacy of enterprise ICT systems require efficient access management, secure transfer of information and strict authentication procedures. In open business networks, traditional hard security falls short in protecting an enterprise, because it divides other actors too narrowly into those trusted (authenticated and authorised) and untrusted (all others), with little ability to adjust to the misbehaviour of trusted actors, for example. Social control methods, such as trust management, allow the system to be more open for collaboration, while still protecting itself both from unknown actors as well as those authorised for the time being (Rasmusson and Jansson 1996). In the centre, the service itself is aware of its required integrity and security constraints, and refuses access that would break these limits, regardless of the requestor.

At present, there are no commonly accepted eContract structures that would sufficiently cover the various business and technical aspects of the eContract. We believe the necessary aspects should be captured in a common upper-level ontology that is further refined with published business network models. Final details are added from service offers, role-by-role, as partners enter eCommunities. In addition, the eContract structures should address the needs of eCommunity membership and life-cycle management at runtime, including interoperability monitoring.

The business network models, specific to their business-areas, should define a sufficient structure for each eCommunity type to support the actual eContracting (negotiation, establishment, monitoring). These models bring in aspects of regulatory systems, business targets, and common practices; the descriptions of available business services in turn define the limits within which the providing enterprises are willing to assume responsibilities in the potential eCommunities. Information related to the eContracts becomes defined by designers, policy creators, service implementors, and enterprise system owners in separate steps of systems engineering and use. Figure 1 illustrates the flow of business-related and technology-related metainformation in the eContracting process in the fundamental steps of metainformation and software production processes for inter-enterprise collaborative systems (Kutvonen and Metso 2005; Ruokolainen and Kutvonen 2006). The elements are described below.

A business network model defines the topology of an eCommunity in terms of roles and interactions between them. A role is a placeholder for a business service: the role definition sets direct requirements with which the service types must conform, and it can, in addition, define assignment rules for other features, for example non-functional aspects or the identities of the participants acceptable for the role. The interaction declarations set conformance requirements for the business processes to be executed between participants. The design of business network models is a profession on its own, requiring understanding of regulatory frameworks on the business area, best business practices, and strategical methodologies suitable for the business.

A service type defines the syntactical structure of interfaces, the semantics of documents to be exchanged, and the service behaviour in terms of the local business process, as observed outside of the software module providing the service. For each service type, there is a set of associated properties that are required for each service offer for this type. A service offer is a declaration of a provided service, naming its service type and giving values to the required properties.

A computational service is a collection of business-relevant software modules. However, it has been a design aim here that the software elements do not need to consider the business strategies or policies. Instead, the runtime environment provides metainformation-driven monitors for governing the software elements.

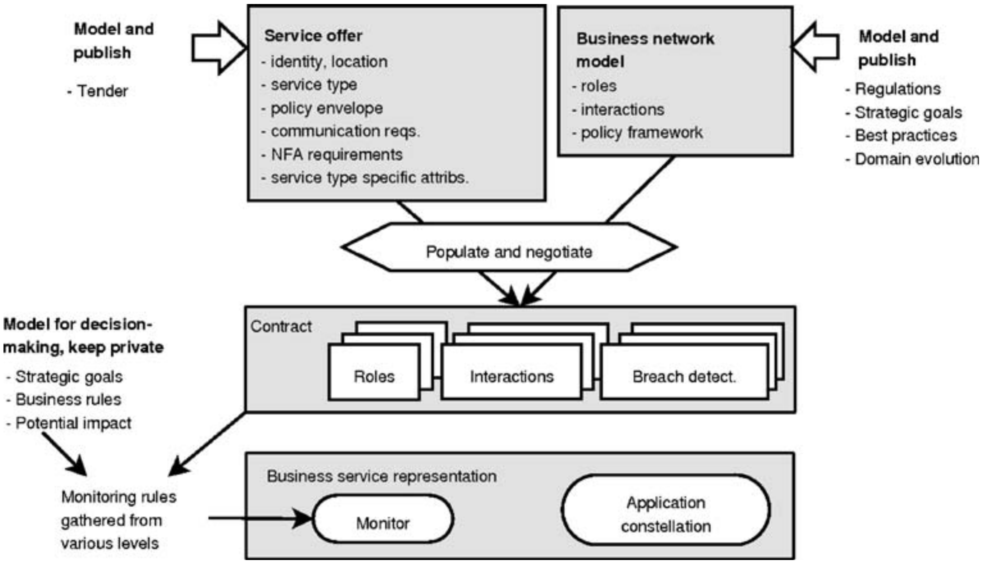

Fig. 1 Information flows for building eContracts and business services

We call the combination of the monitor, the governing rules, and the computational service a business service. It should be noted that part of the governing rules are public as well as part of the eContract, while others are private and known only to the provider of the business service.

While the eContract structuring by business network models capture most social behaviour requirements in the eCommunity, we must consider other layers of interoperability simultaneously. We understand interoperability, or the capability to collaborate, as the effective capability to mutually communicate information in order to exchange proposals, requests, results, and commitments. The term covers technical, semantic and pragmatic interoperability. Technical interoperability is concerned with connectivity between the computational services, allowing messages to be transported from one application to another. Semantic interoperability means that the message content becomes understood in the same way by the senders and the receivers. This concerns both information representation and messaging sequences. Pragmatic interoperability captures the willingness of partners to perform the actions needed for the collaboration. This willingness to participate refers both to the capability of performing a requested action, and to policies dictating whether it is preferable for the enterprise to allow that action to take place.

To capture these interoperability levels, we use the five ODP-RM viewpoints (Open Distributed Processing Reference Model) (S10746 1996) to structure the metainformation in service offers and eContracts. The Enterprise viewpoint is focused on defining the roles and interactions needed between them in order to reach the goal of the community. This corresponds to the definition of external business processes and policies over the eCommunity. The Information viewpoint is for defining the information repositories and the exchange of information elements, as well as calculi for invariants and well-formed changes of the state of the information. The Computational viewpoint is for defining the computational services involved with the community, in terms of interfaces and behaviour towards them. The techniques for describing and comparing behavioural types of services are still immature (Ruokolainen and Kutvonen 2006). The Engineering viewpoint is for expressing how the computational services and the supporting infrastructure are to be used. The Technology viewpoint is for expressing which standard solutions are required for computing or communication platforms, or information exchanges.

The brief analysis above brings us to structuring eContracts and service offers as shown in Table 1. The eContract is structured according to the roles defined in the business network model, and refined by instructions found for each service type required in the roles. In addition, the eContract is structured by epochs, periods of activity where the jointly provided service and the structure of the eCommunity is stable. Separate epochs can be used for breach recovery or otherwise well-limited activity with different set of roles still progressing the work of the eCommunity. The final level of detail captures the requirements on the technical communication. The eContract must also address breach detection and recovery by choosing a published model for that.

In contrast to some upper-level ontology development initiatives, where the aim often is to define a

Table 1 Technical structure and XML-tags for eContract contents

Contract element label	Information type and source	Explanation
Identification and state management		
contractID	String assigned by the initiating NMA	Identity for the eCommunity; potentially jointly with sessionID
description	String assigned by the BNM designer	Purpose of the business network model; business scenario.
startDate	Set by initiating NMA during the negotiation process	If the contract validity is time-triggered, the startDate and endDate are used, indicating date and time.
endDate	Date and time, as above	
state	Integer upkept by the NMA. The eCommunity life-cycle is controlled by a state machine with states of populated, in-negotiation, agreed, established, in renegotiation, terminated.	During the established phase the progress of the conversations (external business processes) can be viewed as steps of considerably large task blocks.
Management of repetitive execution of eCommunity behaviour		
sessions	Array of contractSessions where elements encoded in string-valued tagged fields	Each ContractSession element contains the contractID and sessionID within that contract, identifier for the current epoch, and an integer coded state indicator.
allowedSessions	Integer, not mandatory	Maximum limit of sessions for this eCommunity.
usedSessions	Integer	Counter for controlling the max limit.
concurrentSessions	Integer	Limit for maximum number of concurrent sessions.
The eCommunity structure and behaviour		
businessNetworkModelID	String	Identifies the correct model in repository
participants	Array of participantInfo; participantInfo elements encoded as string-valued tagged fields	A participantInfo element contains service offer information, especially logical and technical addresses of communication end-points for the participants, the management interface location, the partner's digital signature, the role it is associated with and whether this participant is the coordinator or the eCommunity.
bindings	Array of logical connections assigned by NMAs	Reference to the binding type for the mediating channel; provides technical requirements.
modelPolicies	Array of policies; policies expressed as a name-value pair.	Policies governing the eCommunity over all epochs.
architecturePolicies	Array of policies	Policies governing the eCommunity during one epoch.
rolePolicies	Array of policies	Policies governing each role in an architecture.
globalRecoveryProcess	Array of process references	Process models are available in the type repository.
conversationRecoveryProcess	Array of process references	
roleRecoveryProcess	Array of process references	

universal contract structure, we consider the business network model developed for a specific business domain as the right scope for the "universe of discourse" when defining contract structures and ontologies. First, the full range of elements affecting interoperability is not present. Due to the autonomy of service providers, part of the knowledge is private, and failures to conform to the category-forming selection criteria or monitoring rules will raise issues to be addressed by breach recovery processes at the community level. Second, the structure of an eContract is not defined by one template only, but the construction rules for the eContract structure are retrieved from the business network model, service type descriptions, and service offers.

To pair up with this structure of the eContract, the corresponding protocol stack is depicted in Fig. 2. The main difficulty to overcome here is that each stack layer involves a different set of participants. The technology level protocols are used by the peers in the business network to fulfil basic communication interoperability needs, while service level protocols are used between potential peers and the open service market to determine the compatibility of single services. The community-level business processes are used to manage the dynamics and interoperability of the business network as a whole. Besides this, the architecture must support mapping of the business rules and enterprise policies of the members of the eCommunity to the community management protocols on the layer below. Even contract breaches should be resolved by community-level business processes. Therefore, the community-level processes form a backbone for interoperability and collaboration management, placing high demands on the supporting middleware to enable that. In addition, the lack of workflow enactment in the stack is intentional. The business applications are expected to execute their private (local) business processes independently, only interacting according to a monitored external business process. As the coordination approach here expects business services to be able to initiate the necessary activities themselves, only breach detection and recovery processes are needed. The essential failures of service behaviour that we should expect to address are involved with various non-functional aspects (NFA), such as trust, security, QoS, or discrepancies between business policies of autonomous participants.

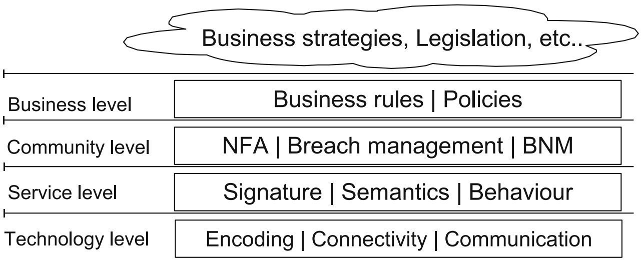
Business strategies, Legislation, etc..
Business level
Business rules | Policies
Community level
NFA | Breach management | BNM
Service level
Signature | Semantics | Behaviour
Technology level
Encoding | Connectivity | Communication

Fig. 2 Interoperability management

3 New infrastructure services and their trusted role

The introduction of middleware level services that are allowed to make commitments on behalf of enterprises raises problems in legal terms as well as in terms of enterprises being able to trust their own middleware agents and the infrastructure services available in the open network. In the following, we only address a few aspects of the trustworthiness of the infrastructure services.

We use a two-phase approach in eCommunity establishment. First, a populator is used to match multiple service offers into a frame formed by a business network model. Then, the eCommunity participants are further negotiated based on the proposed eContracts. The negotiation is performed by network management agents, NMAs, that represent each enterprise.

The populator is responsible for providing a reliable facility to produce interoperable sets of service offers in such a way that they fulfil the requirements of a selected business network model. The interoperable set of service offers means that based on the network model, each service that must communicate with others can do it technically and semantically. The willingness of the participants to interoperate (i.e. pragmatic interoperability) is not considered during the population process and it will be determined at a later time during the negotiations.

The populator chooses the most suitable service offers for each role. First of all, the offers must be of an acceptable service type for the role. The selection is further restricted by policy constraints defined in the business network model or required by the initiator. Finally, a group of additional requirements is raised when the properties declared (e.g., expectations on communication platform and properties) in service offers for interacting roles are matched.

The population process results in a set of eContract proposals, still requiring a negotiation round amongst the proposed partners before the eCommunity establishment phase is completed. The protocol in itself is simple: the initiating NMA receives a specified maximum number of contract proposals from the populator and the initiator orders them according to its preferences. Then it sends out the first proposal to all partners referred to in that proposed eContract. These peers can respond by accepting the proposal, or making a refined proposal, or rejecting it. The responses are sent back to the initiator for combination and further refinement cycles, or for initiation of a new round with the next eContract proposal. During the negotiations, the participating organisations refine the contract terms until they are satisfactory.

The technical environment of the populator is created by the other Pilarcos middleware services (Kutvonen et al. 2007, 2005). As a representative of open service markets, the populator uses a service offer repository. The technical contents of a service offer is described in Table 2. Service offers have mandatory typeIDs which define the mandatory elements for the offer, including attributes.

The metainformation elements provided through the infrastructure repositories must be trustworthy, as the populator builds on the model and typing information to refine it into business network proposals. Trusting the eContracting infrastructure requires strict control over the type repository and business network model repositories. Before published entries can be stored, they must be validated, also in relation to the existing entries. The asserted relationships between stored entries must remain consistent.

These repositories have a considerable organisational effect too, as they provide a means to regulate electronic service markets. Service offer repositories can be controlled by requiring well-formed offers, or even requiring certified enterprises to test offers before accepting them. However, the service provider remains autonomous, and its actions in the eCommunity may not be in accordance with the service offer or the negotiated eContract. In other words, trust in the infrastructure does not directly imply trust between potential partners in the eCommunity that is being formed. Trust between eCommunity partners is a concern of its own, and is one of the aspects to be included into the eContracting process.

The populator uses the type and service offer repositories to produce interoperable business network proposals. Like the repositories, population can be provided as a service by a third party, although a peer implementing a populator for itself is not unfeasible either. A populator must be trusted by the initiator of an eCommunity to match the business network model and service offers as specified, but no further. The populator operates on published information only, and it is not necessary to trust it with a private partner preference policy, for example, unless there is a benefit in doing so. The populator is not told which of the proposals it produces is accepted in the end.

A network management agent (NMA) represents an eCommunity member in the business network (Metso and Kutvonen 2005). It handles negotiations with potential new members and renegotiations if members are changed, it upkeeps state information for the eCommunity, and determines the suitable reaction to the information passed to it by local monitors. For example, if the monitors detect a breach of the terms of the eContract, the violation can at worst lead to a reorganisation of the business network. Every member of the eCommunity has its own network management agent, and they are considered to be fully trusted local agents.

In order to bring trust considerations into the decision processes, support for trust management mechanisms must be added into the infrastructure. Our approach is based on a dynamic combination of experience information and a subjective analysis of the situation in which trust is needed. Earlier experience with the eCommunity member being evaluated is gathered both locally and received through a global reputation network, and it forms a basis for predicting the member's future behaviour. On the other hand,

Table 2 Technical structure and XML tags of service offers

Element	Mandatory	Instances	Explanation
typeID	Yes	1	Identifies the service type the offer is based on.
portOffer	Yes	1-*	Defines operations and their order regarding one port. Describes the properties of each port, and contains the pre and post conditions of each operation.
syncStruct	No	1	Provides causal relation of the events for synchronization.
typingContext	Yes	1	Defines the typing hierarchy that contains the service type which is used by this service offer.
serviceProperty	No	*	Gives values to service attributes. Defines a name-value pair. The value can either be a single type or a value range. The attributes must correspond to the ones in the service type.
providerProperty	Yes	1-*	Describes properties of the service provider. The description is based on a common ontology.

subjectively estimated risk and tolerance for it also depend on various factors not directly dependent on the particular member being evaluated, and our model contains factors to accommodate these considerations as well.

4 eCommunity population

When an eCommunity is wanted for accomplishing a joint goal or for some collaboration, one of the partners initiates the eCommunity establishment via its local NMA. This NMA first calls the populator, then, based on the proposed eContracts, it runs a negotiation with the NMAs of the other proposed partners.

The population request carries two information elements. The first, general part includes a reference to the business network model to be used during the population and directions for the populator for selecting service offers for any of the roles. These directions can advise on the desired number of returned sets of offers, or the maximum time the populator can use for searching the interoperable sets. The directions can also restrict possible service providers or attribute values. The initiator can also refine the properties expressed in the business network model. The model itself expresses requirements for the eCommunity participants, for example, the offers can be required to indicate capability to support transactions. The second part expresses advise on filling each role separately and can include a pre-selected service offer, or directions to use specific selection criteria, or role-based utility functions. The initiator can also fill in service offers for known partners which will participate in the following eCommunity. The populator respects these preliminary choices made, and even makes use of the knowledge by restricting the potential search space accordingly.

Although the initiator is not required to include its own service offer in the population request to represent its own role in the business network, this is beneficial. The included offer will go through the same checking process as all other service offers that will be considered for the business network. At the same time the included service offer and its attribute values acts as the starting point of the properties for the business networks. Similarly, the properties in the business network model have an effect on the eCommunity and its properties.

The population algorithm has seven steps (Ponka 2004):

1. Retrieve the business network model and service types referred to in the role descriptions.
2. Create role populators, set utility functions.
3. Request matching service offers for roles from the service offer repository using all appropriate service types.
4. Check the interoperability of pre-filled roles.
5. Find service offers for each role.
6. Walk through the search tree and test interoperability of service offer combinations.
7. Return business network proposals.

At the first step, the populator retrieves the business network model from the corresponding repository. The model infers the roles and properties of interest. If there is a conflict with the properties of the business network model and the properties given in the population call, the population algorithm is terminated.

For the second step, the populator creates role populators for each role named in the network model, to maintain role-specific information. This information includes current limits for attribute values, and the available service offers based on the attribute values. Utility functions are set as defined in the call; general utility functions are individually set to each role.

Steps from three to five can execute concurrently. During the third step each role populator retrieves service offers from the service offer repository for their own role, taking into consideration the current limitations. A queue of service offers is attached to the role populator, and each offer is flagged either to fulfil or not fulfil the current additional requirements. While the role populators are waiting for the offers, they check the interoperability of service offers given for the pre-filled roles, potentially finding discrepancies and need for terminating the algorithm.

The fifth step forms the main body of the populator. The population advances as a depth-first search in their queues of service offers. This corresponds to a technique called *forward checking*, although the populator implementation includes other variations as well.

Here, a role populator locks the first offer of the queue into the corresponding role. This proposed selection arises further requirements for offers to be accepted for other roles, and those additional restrictions are propagated to the other role populators. Those role populators flag mismatching offers in their queues, thus reducing the search space. However, this temporary removal also allows the process to roll back in case one or more remaining roles have no possible offers left. The locking of service offers to roles is repeated at each role populator until every role has a service offer locked, all possible combinations are exhausted, or the time limit given for the search is exceeded. The role populators may retrieve more service offers from the offer repository, if the queue becomes empty before the search limits have been reached.

The populator uses *attribute frameworks* to manage chains of attributes in the roles of the network model that must all have the same value, because the value has an effect on the interoperability of all roles in the chain. An example of such a requirement is transaction support along the whole supply chain. Essentially this means that each service offer must have the same attribute value for a given set of attributes if a role is a part of an attribute framework. Attribute frameworks make the propagation of constraint values easy, and they enable the populator to detect which attribute values affect which roles.

The populator is able to match several different types of attributes while testing service offers. The main XML Schema simple data types are supported (all numeral types, string, anyURI, time, date, datetime, and boolean). In addition, there are a few different ranges which can be used. These include SomeOf and Exactly. The *SomeOf* range means that a number of the given values must be the same but not all. *Exactly* means that all values must be the same as in other service offers. For continuous values, the ranges are given as a minimum–maximum value pair and for non-continuous values the ranges are given as sets of values.

Utility functions are used to determine the benefit of including a given service offer to the eCommunity. Utility functions can be role specific, network model specific, or the initiator can do the population without them. The utility functions are defined as follows (Ponka 2004):

$$U(a_1, ..., a_n) = \sum_i w_i f_i(a_i)$$

where a_i is a constraint on attribute i, w_i is the weight of the attribute, and f_i is the function to calculate utility based on the value of the attribute. The function returns a value from range [0,1]. The sum of the attribute weights is scaled to 1. It follows that the value of an utility function U is always in the range [0,1]. The higher the value, the higher the utility.

Even though the populator can use utility functions and first tries the offer with the highest utility value, it does not mean that the resulting business network proposal has the highest possible total utility. This is because the depth-first search. For example, if the best offer for role two is chosen, the populator will try every possible offer to role three before selecting the second-best offer for role two. Therefore the best offer for role two can result in a lower utility on the whole than the second-best offer for role two. This all depends on the values of the attributes in a given service offer and the effect the values have on the remaining roles.

Finally, at the seventh step, the populator returns the business network proposals to the requesting network management agent. The populator cannot guarantee that it finds the requested amount of proposals.

The populator has been found feasible to use for eCommunity discovery (Kutvonen and Metso 2005). The performance behaviour of the populator is acceptable both in terms of delay and scalability. The performance of the populator is dependent on the constraint propagation scheme used. The forward checking model is efficient in reducing the size of the remaining search tree. The size of the search tree will effectively determine how many possible combinations are left at a given time during the population. The size of the tree is not consistent through the whole population. As more roles have been filled with service offers, the size of the search tree will decrease. If the process has to roll back a role, the tree will grow in size again. The main cost in this model is dependent on the efficiency of calculating new constraints on the service offer attributes and propagating them. These constraint values are always recalculated when a role is filled during the population. The utility functions are just another way of calculating the constraints on the service offers. However, the complexity of an utility function plays a factor when using them. The more complex the utility functions, the more time it takes from the populator to calculate the utility value.

Compared to traditional trading facilities such as CORBA Trader (OMG 2002) and UDDI (Uddi registry - technical specification 2006) the main advantage of our approach is the ability to match multiple service offers into a functioning eCommunity, using an enhanced service type system, in a way that is suitable for automated interoperability testing and enforcement. The Pilarcos type repository provides an extensible service type system with a strict type discipline that takes into account aspects of service behaviour and semantics, subtyping, and relaxed matching of independently defined types with assessed relationships or transformations between them. The service types provide a basis for interoperability negotiations in terms of service offers and suitability to roles within known business network models.

5 Trust in eCommunity establishment

The population process acts on the public information available in business network models and service offers. However, entering a collaboration involves additional motivations, policies and reasoning that is of a private nature. Most importantly, the private decisions relate to trust between partners and trust in their business services.

Service offers and business network models are public information, but trust information includes private evaluations which can have averse effects if they become public knowledge. For example, a subcontractor may not wish to make its distrust in a large vendor known to the world, nor reveal details of its evaluations of risks and incentives related to a particular business network composition. Participants should therefore be able to set trust requirements related to their business network models and service offers, while retaining control of their private trust information. In addition, even these trust requirements should be made public only if it adds value to the process.

Standard trust-related requirements, such as certification for a particular service, can be included in network models and service offers and checked by the populator. They can be used as minimum requirements or scored for utility calculations. The initiator can also provide blacklists in its populator request to avoid recurring proposals with unsuitable service providers.

In the Pilarcos middleware, the population of a business network can be provided as a service to the initiator by a third party. If this party would be trusted by all potential partners, the private trust or policy information could be given out, but it is more realistic to keep the private decisions at the local NMAs. After having analysed different methods of using trust information in the population process, we have decided that due to privacy concerns, a populator is not given access to enough information to filter or arrange service offers based on trust (Kutvonen et al. 2006).

Therefore, trust decisions on the populator's proposals must be made at the negotiation phase. First, the initiator selects a proposal it finds optimal and begins the negotiations by sending it to other potential network members, who can either accept it, make changes to it or reject it altogether. Trust decisions are made by the initiator and the other negotiators on whether to join the network and on what terms (Kutvonen et al. 2006); later, during the operational phase, further trust decisions are made on whether a particular risk-relevant commitment is considered reasonable (Ruohomaa et al. 2006).

In this negotiation phase, each NMA makes a trust decision before committing to participate in the eCommunity. A trust decision is the result of a subjective evaluation of local information combined with additional third-party experience information received via a reputation network. More formally, we define trust as *the extent to which one party is willing to participate in a given action with a given partner, considering the risks and incentives involved*.

To produce a trust decision, the trust management system checks whether its completed risk analysis is within tolerated values for that situation. A situational cost-benefit estimate and representation of the tolerance for the particular situation are generated dynamically from 7 factors defined below, and a trust decision is produced by comparing the two.

Our trust model has 7 factors: *trustor, trustee, action, reputation, risk, importance* and *context* (Ruohomaa and Kutvonen 2005). The trustor, trustee and action parameters, together with the current state of the system, determine the situation the trust decision is made in. The party making a subjective trust decision, the trustor, is the guarded service, represented by an agent. The target of the decision is the trustee, another peer in the network. The action parameter denotes a group of SOAP messages exchanged. For partner selection purposes, the action parameter can be seen to extend to cover the entire collaboration from a risk estimation point of view. Technically, however, it remains a set of messages exchanged with the populator, who in essence acts as a proxy of the actual trustee by suggesting it as a possible partner for a collaboration.

Reputation is the measure of a peer's perceived trustworthiness. It is based on a subjective view combined from experience information received through local monitoring as well as through reports from other peers in a global reputation network. The credibility and information content of the statements are evaluated by the recipient in order to build a local reputation value.

The risk factor provides a tactical cost-benefit estimate on the action considered. It expresses the potential benefits and costs of a positive trust decision to different assets, such as money, security and customer satisfaction. The information is stored as probability values for each severity class of effects to a particular asset, for example a 0.1 probability of a "considerable" loss of security, 0.3 probability of a "minor" loss and 0.6 probability of no effect. For example for monetary assets, a positive result is both possible and desirable. The action parameters and the reputation of the trustee affect this estimate, as well as the context adjustments described later.

The importance factor represents strategic valuations in the enterprise, which are independent of any estimate of what the trustee might do. These considerations, such as the cost of denying an action defined in the eContract, or the benefit of good service to creating a working partnership, guide the tolerance of risk.

The context factor represents temporary adjustments made to other factors, especially risk and importance. The changes can be initiated by any of three possible source types: the internal state of the peer's system, the state of the enterprise in general or the state of the eCommunity the peer is a member of.

6 Operational time issues in eCommunity management

The eCommunity establishment phase can consider only those aspects of interoperability that can be expressed statically in the service type and the business network model definitions, or as ranges of acceptable policy decisions in service offers. However, policies and context of the collaboration can change, or the partners can even fail or prioritise some other eContract or enterprise policy. Therefore, operational time support for the eCommunity is essential.

The operational time support consists of monitoring of partners for behaving according to the eContract rules, maintenance of progress information of the collaboration task, and the management of partner-initiated changes or system-initiated changes that are caused by breach detection services. Here we concentrate only on breach detection and breach management. These are the parts mostly involved with trust management and reputation formation.

In the Pilarcos architecture, each business service is guarded. These guards take care of the restriction of the computational service capabilities to those externally available facilities we call the business service. The guards work in two ways. First, they protect the business service from inappropriate messaging from outside. Second, they restrict the business service from using its full capabilities in situations where enterprise policies only allow a restricted form of the service to be provided to partners.

These guards are implemented by rule-based monitors located at the communication end-points of each service. The monitors continuously evaluate whether the observed messaging is conformant to the expected behaviour explicated in the eContract.

The monitors are configured with information from the eContract and internal business policies. The core of the monitors consists of a traffic analyser advised by a state-machine. For the analyser, it is possible to configure different behaviour expectations by describing the incoming and outgoing message exchange of the current partner as state changes, and to define action rules and evaluation rules. The action rules are used for marking the progress of the business processes and for collecting a coarse-grain state of the eCommunity progress. The rule advises the monitor to report the completion of a subsequence of messaging as a completed task to the local NMA, which in turn can report to other NMAs. Logically, this splits the state-machine into an abstract task-oriented machine, and a concrete message-level analyser. The grouping of messages to tasks can be derived from annotations in the business network models. The evaluation rules can address any aspect of the exchanged messages, for example, aspects common in the security area: the content of messages for information content restrictions, or even, use techniques from intrusion detection (Ruohomma et al. 2006; Viljanen 2005b). Based on the evaluation rules, the monitor can raise problem notifications on breach, missing message, and information content mismatch issues.

If a monitor detects a pattern of abnormal behaviour, it sends a report to the local NMA. The NMA decides whether the abnormal behaviour triggers a breach or whether it is a minor incident that is to be repaired locally. If the NMA considers the incident to be serious, it contacts the other NMAs of the eCommunity, suggesting that a resolution process is started.

The monitors can be set either to passive, active, or proactive mode. In passive monitoring, the events are only logged for further examination, while in active mode the monitor logs events and actively reports mismatches to NMAs. Proactive monitoring prevents mismatches from happening by blocking mismatching messages from being sent or received by the services.

The proactive monitoring has the highest cost, but provides the highest level of breach prevention and service interoperability guarantees. Selecting the granularity and mode of monitoring is a major scalability design challenge for the system administrators. This calls for additional, more sophisticated tools for analysing cost of alternative configurations.

The monitoring approach is used in other related projects as well, ranging from monitoring of the success of business processes (Rabelo et al. 2000; Daskalopulu et al. 2002) and monitoring of the business itself (Scheer et al. 2004) to intrusion detection (Viljanen 2005a). Most approaches with the same level of monitoring goals use a passive approach: for example, BCA (Quirchmayr et al. 2002) provides a centralised notary to detect contract breaches post-operatively.

For resolving the detected breaches, the Pilarcos architecture requires the eContract to carry references to the agreed resolution process. In principle, different business network models have different properties in terms of recovery potential, and the choice of the recovery process is not free. Depending on the verified recoverability properties of the business network model, it may be possible to compensate and restart, or replace a member and roll it to the state expected by others in the eCommunity. Furthermore, the participants of the recovery phase may be different from the set of the original eCommunity members. The current prototype is able to initiate a simple negotiation on whether a participant is replaced or not but we have envisioned that a new epoch is started for the resolution.

The resolution process also introduces a position in which bad experience or good experience can be fed into the reputation management system, to be used in future local trust decisions and shared with other members of the reputation network.

7 Conclusion

This paper proposes an automated, generic method for eCommunity management in an inter-enterprise, open environment. There are two phases in the management: community establishment and monitoring of the community for fulfilment of trusted activities. For the establishment phase the Pilarcos middleware provides facilities for selecting eCommunity participants with focus on the social and contractual aspects, especially external business processes, concept of utility, and trust in potential collaborators. The solution is based on multi-partner matching of service offers, guided by a jointly selected, public business network model. It thus extends the traditional trading or brokering architectures. The presented eContract structure pulls out publishable aspects of interoperability issues, still leaving some pragmatic aspects private. For the operational phase the Pilarcos middleware provides facilities for monitoring business services against the expectations of the eContract and local enterprise policies. The monitoring information can be used as feed-in for the reputation management network that affects trust decisions of later eCommunity establishments, and as triggers for breach management processes for the eCommunity involved.

The solution differs from other eContracting approaches by capturing all three aspects, social, contractual and technical, into an automated process where all functional and non-functional aspects of the collaboration are treated according to a few simple principles. The main design goal has been to separate interoperability and eCommunity management tasks into a B2B middleware layer that is founded on metainformation repositories for business networks, business services and contractual rules. The solution is closely related to work on virtual enterprises and virtual enterprise breeding environments, but takes a more pragmatic view in the separation of generic B2B negotiation and eCollaboration management routines.

The Pilarcos approach is strongly based on federation across enterprises and services that are encapsulated and autonomously administered. This trend is becoming visible on larger scale standardisation activities and new EU research agendas. Because of the service-oriented nature of our approach it aligns well with RM-SOA (McKenzie et al. 2006), although the level of automation aimed at requires us to introduce a more extensive set of concepts than the RM-SOA. NESSI (Nessi strategic research agenda 2006) is a new European initiative to bring service oriented business models closer to reality, with a goal to outline an ICT framework for future service-oriented architectures and economy. The NESSI goals are similar to those in EU FP7 (FP7 2006) where the key issues of Pilarcos goals appear: federation, model-governed management, trust management with local trust decision but with global reputation information and others. Many other breeding environment projects for virtual enterprises, like ECOLEAD (Camarinha-Matos and Afsarmanesh 2006; Rabelo et al. 2006), focus either on supporting collaboration between humans by joint facilities, or require stepwise human negotiation for designing the actual collaboration-supporting agent system.

The proposed management of trust consists of local trust decision when entering eCommunities and at each trust-guarded transaction. The decisions take into consideration globally available reputation information, either positive or negative. The reputation information must be associated with fairly permanent targets with well-known identities; the targets shall be business services. Our approach differs from other trust-management work by emphasising private, subjective decisions at each enterprise at the level of business services, based on both technical and business-level information. Otherwise the goals are fairly similar to those of the TrustCOM project (Wilson et al. 2006) or SECURE (Cahill 2003). However, TrustCOM enforces distributed business process execution, and UDDI-based service discovery. For the SECURE project that has implemented a trust management system aimed for private persons, the battle against the Sybil attack (results from inexpensive new identities) is essential. In contrast, we require stable identity management, and furthermore support of a robust reputation management network (Ruohomaa et al. 2007).

A number of challenges have to be addressed for further maturing the federated management architectures. First, the framework for eContracts should be standardised and a global knowledge base for interoperability information established (Kutvonen 2007). Second, a suitable identification mechanism needs to be created for associating trust, reputation, security and contract information to business services. The existing development does not address the required granularity. Third, the experience turned into reputation information should be based on a commonly acceptable framework of concepts, ranging, for example, from successful and correct performance in business transactions to illegal transactions or breaches of technical criteria. For all these axes, ontologies should be developed to capture the metrics to be used. Finally, the role we envision for reputation systems, service selection systems and interoperability knowledge-bases in the open collaborations creates new vulnerabilities. We have started a comprehensive threat analysis, but additional work is still needed for creating a system that would resist these new threats beyond the means already embedded in the architecture. The current facilities already address these threats in ways that determine architectural decisions, such as encapsulation of service type information into trusted knowledge bases, being prepared for operational time breaches for autonomy reasons, and including a set of negotiation protocols in the management facilities.

Acknowledgements This work has been carried out at the Department of Computer Science at the University of Helsinki. The research group involved, Collaborative and Interoperable Computing group, builds on work done in various projects funded by the national technology development center TEKES and industrial partners.

References

Angelov, S., & Grefen, P. (2003). The 4W framework for B2B e-contracting. *International Journal of Networking and Virtual Organisation, 2*(1), 78–97.

Booth, D., Champion, M., Ferris, C., McCabe, F., Newcomer, E., & Orchard, D. (2004, February). *Web services architecture*. (http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/)

Cahill, V. (2003, August). Using trust for secure collaboration in uncertain environments. *Pervasive Computing, 2*(3), 52–61.

Camarinha-Matos, L. M., & Afsarmanesh, H. (2006, September). Modeling framework for collaborative networked organizations. In *Network-centric collaboration and supporting frameworks. Seventh IFIP working conference on virtual enterprises* (pp. 3–14). Berlin Heidelberg New York: Springer.

Chiu, D. K. W., Cheung, S. C., Hung, P. C. K., Chiu, S., & Chung, K. (2005, July). Developing e-Negotiation Support with a contract template meta-modeling approach in a Web Services environment. *Special issue on web services and process management in the Decision Support System (DSS), 40*(1), 51–69. (http://teaching.ust.hk/~csit600c/eNeg.pdf)

Daskalopulu, A. (2002). Evidence based electronic contract performance monitoring. The INFORMS journal of group decision and negotiation. *Special issue on formal modelling in E-Commerce.*

Daskalopulu, A., Dimitrakos, T., & Maibaum, T. (2002, November). Evidence-based electronic contract performance monitoring. *Group Decision and Negotiation, 11*(6), 469–485.

Dellarocas, C., & Klein, M. (1999, December). Designing robust, open electronic marketplaces of contract net agents. In *Proceedings of the 20th International Conference on Information Systems (ICIS), Charlotte, NC.*

FP7 (2006, April). *European commission 7th framework program*. (http://ec.europa.eu/research/fp7)

Griffel, F., Boger, M., Weinreich, H., Lamersdorf, W., & Merz, M. (1998). Electronic contracting with COSMOS – How to establish, negotiate and execute electronic contracts on the Internet. In *Proceedings of second international Enterprise Distributed Object Computing Workshop (EDOC'98).*

Grosof, B. N., & Poon, T. (2003). SweetDeal: Representing agent contracts with exceptions using XML rules, ontologies and process descriptions. In *Proceedings of International Conference on the world wide web.*

IS10746 (1996). *Information Technology – open systems interconnection, data management and open distributed processing. Reference model of open distributed processing.*

Kutvonen, L. (2007). Building B2B interoperability middleware – Knowledge management issues. In *Interoperability for Enterprise Software and Applications (I-ESA2007).*

Kutvonen, L., & Metso, J. (2005, September). Services, contracts, policies and eCommunities – Relationship to ODP framework. In P. Linington, A. Tanaka, S. Tyndale-Biscoe, & A. Vallecillo (Eds.), *Workshop on ODP for Enterprise Computing (WODPEC 2005)* (pp. 62–69).

Kutvonen, L., Metso, J., & Ruohomaa, S. (2006, October). From trading to eCommunity population: Responding to social and contractual challenges. In *Proceedings of the 10th IEEE international EDOC conference (EDOC 2006)*. Hong Kong: IEEE.

Kutvonen, L., Metso, J., & Ruokolainen, T. (2005). Inter-enterprise collaboration management in dynamic business networks. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE.* (Vol. 3760). Agia Napa, Cyprus.

Kutvonen, L., Ruokolainen, T., & Metso, J. (2007, January). Interoperability middleware for federated business services in web-Pilarcos. *International Journal of Enterprise Information Systems*, Special issue on Interoperability of Enterprise Systems and Applications, *3*(1), 1–21.

Linington, P. F., Milosevic, Z., Cole, J., Gibson, S., Kulkarni, S., & Neal, S. (2004, October). A unified behavioural model and a contract language for extended enterprise. *Data Knowledge and Engineering Journal, 51*(1), 5–29.

McKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., & Metz, R. (2006, July). *Reference model for service oriented computing 1.0*. (http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf)

Metso, J., & Kutvonen, L. (2005, September). Managing virtual organizations with contracts. In *Workshop on Contract Architectures and Languages (CoALa2005)*. Enschede, The Netherlands.

Nessi strategic research agenda (2006, February 13). *NESSI SRA. (Public draft 1)*.

OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee (2002, September). *Collaboration-protocol profile and agreement specification*. (http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2_0.pdf)

OMG (2002). *Corba trading service*. (http://www.omg.org/cgi-bin/doc?formal/2000-06-27)

Papazoglou, M. P., & Georgakopoulos, D. (2003). Introduction. *Communications of the ACM, 46*(10), 24–28.

Ponka, I. (2004). *Populaattori*. University of Helsinki. (Internal report, in Finnish.)

Quirchmayr, G., Milosevic, Z., Tagg, R., Cole, J., & Kulkarni, S. (2002). Establishment of virtual enterprise contracts. In *Database and expert systems applications : 13th international conference* (Vol. LNCS 2453, p. 236). Berlin Heidelberg New York: Springer.

Rabelo, R., Camarinha-Matos, L. M., & Vallejos, R. V. (2000). Agent-based brokerage for virtual enterprise creation in the moulds industry. In *E-business and virtual enterprises*. (http://gsigma-grucon.ufsc.br/massyve)

Rabelo, R. J., Gusmeroli, S., Arana, C., & Nagellen, T. (2006). The ECOLEAD ICT Infrastructure for Collaborative Networked Organizations. In Network-centric collaboration and supporting frameworks (Vol. 224, pp. 451–460). Berlin Heidelberg New York: Springer.

Rasmusson, L., & Jansson, S. (1996). Simulated social control for secure Internet commerce. In Proceedings of the 1996 workshop on new security paradigms (pp. 18–25). Lake Arrowhead, Calfornia, USA: ACM.

Ruohomaa, S., & Kutvonen, L. (2005, May). Trust management survey. In *Proceedings of the iTrust 3rd international conference on trust management* (pp. 77–92). Paris, France: Springer-Verlag (LNCS 3477/2005).

Ruohomaa, S., Kutvonen, L., & Koutrouli, E. (2007, April). Reputation management survey. In *Proceedings of the second international conference on availability, reliability and security (ARES)* (pp. 103–111). Vienna, Australia: IEEE Computer Society.

Ruohomaa, S., Viljanen, L., & Kutvonen, L. (2006, March). Guarding enterprise collaborations with trust decisions—The TuBE approach. In *Proceedings of the first international workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems (IS-TSPQ 2006)*. Bordeaux, France: Springer-Verlag.

Ruokolainen, T., & Kutvonen, L. (2006). *Service typing in collaborative systems. Interoperability for Enterprise Software and Applications Conference (I-ESA'06)*. Bordeaux, France: Springer-Verlag.

Scheer, A.-W., Abolhassan, F., & Jost, W. (2004). *Business process automation: ARIS in practice*. Berlin Heidelberg New York: Springer.

Schoop, M., Jertila, A., & List, T. (2003). Negoisst: A negotiation support system for electronic business-to-business negotiations in E-Commerce. *Data and Knowledge Engineering, 47*(3), 371–401.

Singh, M.P., & Huhns, M.N. (2005). *Service-oriented computing: Semantic, processes, agents*. New York: Wiley.

Thatte, S., et al. (2005). *Business process execution language for web services*. (ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf)

Uddi registry - technical specification (2006, February). (http://uddi.org/pubs/uddi_v3.htm)

Viljanen, L. (2005a). *A survey on application level intrusion detection vol.C-2004-61; Tech. Rep*. University of Helsinki, Department of Computer Science.

Viljanen, L. (2005b). Towards an ontology of trust. In *Proceedings of the 2nd international conference on Trust, Privacy and Security in Digital Business (TrustBus'05)*. Copenhagen, Denmark: Springer-Verlag, LNCS 3592/2005.

Wilson, M., et al. (2006, March). The TrustCoM approach to enforcing agreements between interoperating enterprises. In *Interoperability for Enterprise Software and Applications Conference (I-ESA'06)*. Bordeaux, France: Springer-Verlag.

Xu, L., & Jeusfeld, M. A. (2003). Pro-active monitoring of electronic contracts. In *Proceedings of CAiSE 2003*. Berlin Heidelberg New York: Springer.

Lea Kutvonen holds a PhD in Computer Science from University of Helsinki. She leads a research group on Collaborative and Interoperable Computing (CINCO), within the specialisation area of Networking in Open Distributed Systems of the Department of Computer Science. Her interests include inter-enterprise computing and interoperability, developing B2B middleware solutions, distributed computing architectures, trust management and other nonfunctional aspects of enterprise interoperability.

Janne Metso holds a MSc in Computer Science and he is currently a PhD student in the CINCO group at the University of Helsinki. His research is focused on runtime negotiation facilities and expert system support for establishing and maintaing inter-enterprise collaborations.

Sini Ruohomaa holds an MSc in computer science and she is currently a PhD student in the CINCO group. Her research interests include trust and reputation management in open collaboration systems and inter-enterprise computing.