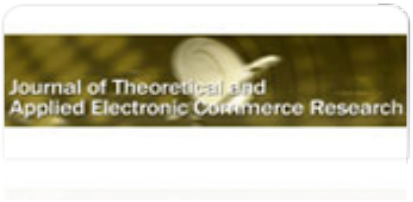

Journal of Theoretical and Applied Electronic Commerce Research
E-ISSN: 0718-1876
ncerpa@utalca.cl
Universidad de Talca
Chile

Ruohomaa, Sini; Kutvonen, Lea
Trust and Distrust in Adaptive Inter-enterprise Collaboration Management
Journal of Theoretical and Applied Electronic Commerce Research, vol. 5, núm. 2, agosto, 2010, pp. 118-136
Universidad de Talca
Curicó, Chile

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

Trust and Distrust in Adaptive Inter-enterprise Collaboration Management

Sini Ruohomaa[1] and Lea Kutvonen[2]

University of Helsinki, Department of Computer Science
[1] sini.ruohomaa@cs.helsinki.fi, [2] lea.kutvonen@cs.helsinki.fi

Received 15 February 2010; received in revised form 31 May 2010; accepted 9 June 2010

Abstract

The success and competitive edge of enterprises has become increasingly dependent on the enterprises' agility to become members in business networks that support their own business strategies. Therefore, integration solutions with their well-weathered strategic networks are no longer sufficient. Instead, there is need for more open business service ecosystems where previously unknown services and partnerships can be utilized. The ecosystem is to be supported with infrastructure services to solve the evident problems of semantic and pragmatic interoperability and collaboration-governing contract management. Furthermore, the ecosystem must support the creation of trust relationships with previously unknown partners, and reacting to encountered breaches of trust within collaborations. This paper proposes a trust management system where autonomous enterprises make automated, private trust decisions about their membership in each collaboration separately, while taking advantage of globally shared reputation of business peers in earlier collaborations. The trust decisions are adjustable to different and changing business situations.

Key words: Trust Management, Reputation-based Trust, Inter-enterprise Collaboration, Service Ecosystems, Trust Decisions

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

1 Introduction

The success and competitive edge of enterprises has become increasingly dependent on the enterprises' agility to become members in business networks that support their own business strategies. Therefore, integration solutions with their well-weathered strategic networks are no longer sufficient. Instead, there is need for more open business service ecosystems where previously unknown services and partnerships can be utilized. This trend is visible in many research and development activities in Asia and Europe, including projects like ECOLEAD [30], CrossWork [24], Pilarcos [22], and present FINeS cluster projects [23].

The ecosystem [42] is to be supported with infrastructure services to solve the evident problems of i) collaboration-governing contract management and ii) semantic and pragmatic interoperability. First, 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.

Second, the collaboration management goal is that in the future, individual users, enterprises or public organizations can easily compose new services from the open service markets, or establish temporary collaborations with complex peer relationships. Furthermore, these contract-governed collaborations can be managed by their partners. All this is supported by a global infrastructure with facilities for interoperability control and contract-based community management (establishment, control and breach recovery) among autonomous organizations; this infrastructure also takes responsibility for governing trust and privacy-preservation issues.

This paper focuses on the ecosystem aspects that are needed for supporting the creation of trust relationships to previously unknown partners, and for reacting to encountered breaches of trust during collaborations. All collaboration builds on trust: relying on an autonomous enterprise partner for a joint venture creates uncertainty and risks, which must be balanced against the expectation of an even greater benefit.

We define trust as the extent to which an actor is willing to participate in a given action with a given partner, considering the risks and incentives involved.

Trust decisions evaluate the available information on risks and benefits to determine whether a collaboration should be joined or if an ongoing collaboration has become too risky to continue in. While the decision to first join a collaboration is important, it is not enough by itself: the business situation as well as the behavior of collaborators can change during the collaboration, creating a need to make new decisions to re-evaluate the situation whenever resources are being committed.

The trust decisions need to be adjustable to different and changing business situations. The collaboration commitment decisions as well as the decisions reacting to breaches must be relative to the business incentives and strategies, eagerness for risky business in hope of good return of investment, search for niche markets, robustness of business domain, and knowledge about increased technical threats against the enterprise's computing systems.

This paper proposes a trust management system where autonomous enterprises make automated, private trust and distrust decisions about their membership in each collaboration separately, while taking advantage of globally shared reputation of business peers in earlier collaborations.

Section 2 gives an overview of the Pilarcos approach to open service ecosystem support and the management of inter-enterprise collaborations. This is the context in which the trust and distrust decisions are used. Section 3 introduces the trust-related actions and processes in the ecosystem and in the collaboration lifecycle. It describes the enterprise assets as a basis for measuring the risks and level of trust required for a trust decision. It further explains how the decision is supported by reputation systems, which provide a history view to predict future behavior of an enterprise's business service. Section 4 analyses the strengths, weaknesses, opportunities and threats of the proposed system, and concludes with suggestions for future work.

2 Inter-enterprise Collaboration Management in Service Ecosystems

To survive and succeed in the modern social and networked business environment, enterprises must be able to participate in multiple business networks simultaneously, be quick in adopting new kinds of well-crafted business models, and establish new collaborations swiftly. Acquiring this kind of flexibility sets two requirements: first, routine decision-making on committing to a collaboration must be automated (similarly to automated broker agents dealing

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

on stock exchanges) and second, situational information in the large service ecosystem (availability of services, reputation of partners, alliances, risk involved in the business model) must be made available to the automation tools.

The Pilarcos architecture views inter-enterprise collaboration as a loosely-coupled, dynamic constellation of business services. The constellation is governed by an eContract that captures the business network model describing the roles and interactions of the collaboration, the member services, and policies governing the joint behavior [19], [21].

The Pilarcos architecture for the open service ecosystem (Figure 1) includes

1. the participating enterprises, with their public business service portfolios exported [22];
2. business-domain governing consortia, with their public models of business scenarios and business models expressed as exported business network models (comprising a set of business process descriptions and compulsory associations between roles in them, and governing policies about acceptable behavior) [17];
3. a joint ontology about vocabulary to be used for contract negotiation, commitment and control [25], [39]-[40];
4. legislative rules to define acceptable contracts [25];
5. technical rules to define conformance rules over all categories of metainformation held as collaboration and interoperability knowledge [41]-[42];
6. infrastructure services to support partner discovery and selection, contract negotiation and commitment to new collaborations, monitoring of contracted behavior of partners, and breach detection and recovery services; these services especially include trust aspects in decision-making on commitment and breaches [19], [21];
7. reputation information flow, collected from past collaborations [36], [38].

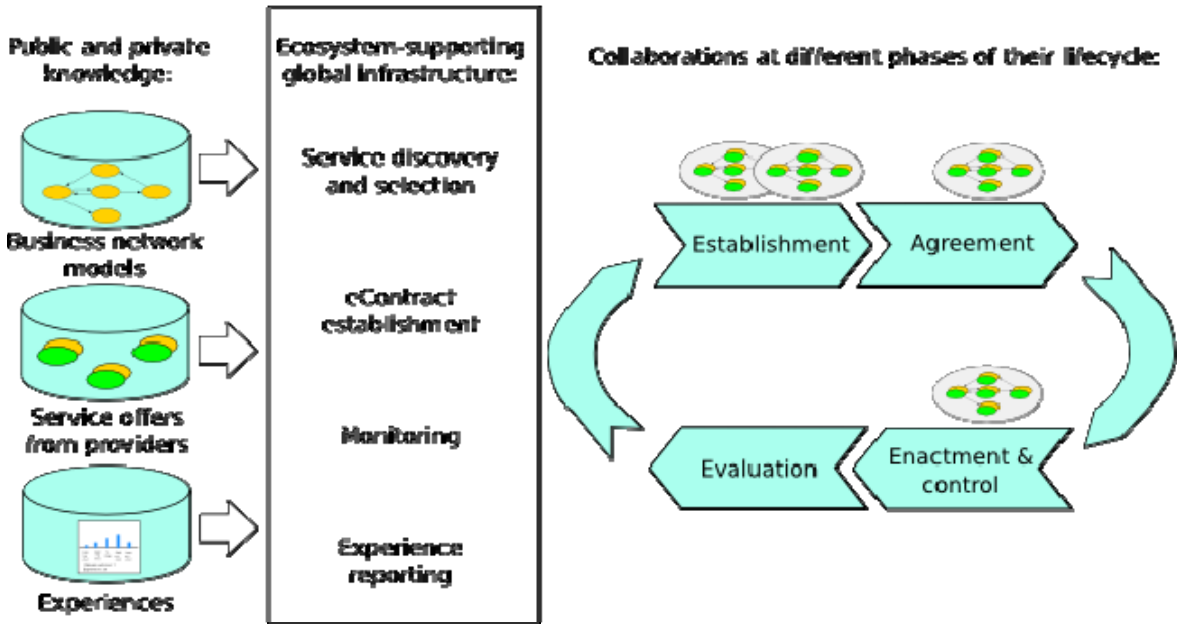

Figure 1: An Overview of the Open Service Ecosystem.

Figure 1 illustrates the ecosystem lifecycle. On the left, metainformation repositories and flows are shown to be created by the publishing and exporting processes denoted above as items 1 and 2. The repositories in particular contain public information about the available business network models, available services and reputation information about the available services. This information is stored in globally federated repositories, applying strictly specified structuring and conformance rules [18] created by the processes listed above as items 3, 4 and 5. The information is in turn utilized by the ecosystem infrastructure functions listed as item 6, e.g. service discovery and

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

selection, eContracting functions, monitoring of business services and reporting of experience on the services when a collaboration terminates. These functions are further described below.

On the right, the lifecycle of independent collaborations is shown to flow from establishment to evaluation at the dissolution phase. The infrastructure functions provide support for the four phases of the collaboration: establishment, agreement, enactment and control, and evaluation.

Unlike many existing proposals [30], [27], [51], collaboration management on the open service market does not rely on centralized control of the entire collaboration: participants remain autonomous and independent of the initiator of the collaboration. In addition, some of the management support needed can be offered as services by specialized third parties rather than requiring one ultimately trusted actor to rule over everything.

Service discovery and selection supports the collaboration establishment phase. It is based on public business network models describing the collaborations, and public service offers made by service providers [19], [40]. The business network models capture the best practices of a given field, and they are built from formally defined service types. The task of producing these models and types naturally falls to consortia and standardization bodies.

Service selection includes automated static interoperability checking, which ensures that the service offers fit the model of the collaboration, and have terms that are compatible with other offers being selected into the proposed business network. As service discovery and selection is separate from contract negotiations, it can be done without access to sensitive information; this makes it possible to have this task implemented as a third-party service [19].

Automated *eContract establishment* supports the agreement phase of the collaboration [19]. The business network model and the proposed service offers to populate the roles in it are processed by an automated contract negotiation infrastructure, which is controlled locally by each collaboration partner. Contracts are based on templates specific to the collaboration model, and the terms of service provision given in service offers form the basis of negotiations. The negotiated eContract includes a model of the business process of the collaboration, as well as the finalized terms of service in the form of accepted service offers.

Monitoring supports the enactment and control phase of the collaboration in particular [20]. It is done by each collaborator to protect local resources, keep track of the progress of the collaboration, and to ensure that partners follow the collaboration model. The business network model and service provision terms set by the negotiated eContract form the specification of correct behavior in the collaboration, which becomes relatively straightforward to monitor.

There are two practical limitations to how powerful monitoring can be made in inter-enterprise collaboration. First, given the autonomy of participants, there can be no all-seeing trusted third party, monitoring all that goes on in the collaboration. Second, monitoring must generally be non-invasive, i.e. it can only observe the messages sent by the technical service, not the internal state of the service.

As a result, business network models must specifically support monitoring by demanding sufficient information to be included in the messages sent between services. If a specific collaboration type requires a notary to act as a witness to specific activities, for example, it must be included as an actor in the business network model, and the model must direct any message traffic in need of observing through the notary actor. The strict separation of policy monitoring and enforcement from the technical service application is a necessity, as the service can participate simultaneously in multiple collaborations with different shared policies. On the other hand, message-based monitoring also avoids the high costs of inserting sensors into the code of legacy applications [49].

Finally, while it is relatively straightforward to monitor for contract and policy compliance, policy-abiding but “suspicious” behavior is more difficult to capture. Anomaly detection solutions suffer from false alarms and therefore would be best used under human supervision [49].

Experience reporting supports the evaluation phase of the collaboration, and connects to the monitoring service during the enactment of the collaboration [36], [38]. Experience reporting forms the core of social control [31] in the open service ecosystem. As contract violations are detected by monitors, they are published to other actors as well: it is important to create a direct reputation impact to privacy and data security violations in order to limit the damage that misbehaving actors can achieve in other collaborations.

The storage, processing and reporting of globally shared reputation information is a challenging problem, as it requires support for evaluating the credibility of experience reports [37]. False reports do not only affect the targeted service, but also inhibit the other actors’ ability to assess its behavior, reducing the social control impact of reputation [45].

Current approaches to inter-enterprise collaboration differ in the format of the breeding environment where business networks are built up. Most breeding environments (ECOLEAD, CrossWork) are based on strategic network membership, for which partners must be assessed for their trustworthiness. After that, all potential partners are considered trustworthy, the business network model is negotiated with a selected set of partners, and the joint model

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

is expected to guarantee interoperability. Within the trusted breeding environment, the partners authenticate each other through cryptographical certificates or similar security tokens [51], [29], which forms the extent of automated trust management in these systems.

However, gaining membership in the trusted virtual breeding environment is no real guarantee that the enterprise partner will behave well and follow shared norms in the future. Causes for misbehavior are numerous: besides the partner being downright malicious, their service can be buggy, overloaded or subverted by an outside attacker. Participating in multiple collaborations may leave them with conflicts of interest either between collaborations or between a collaboration and local policy, and in the end they may well wish to optimize their resource use by simply bending the contract a little. These are all familiar problems from the brick-and-mortar enterprise tradition, and they will follow us into the ecosystem of modern inter-enterprise collaborations.

The Pilarcos approach reaches further by separation of business network model design and collaboration partnership, dynamic evaluation of conformance to the contract during business network operation, and especially, including distrust of collaboration partners during the collaboration as well.

3 Trust Challenges and Concepts

Above, we have emphasized the importance of decision-making at specific places in the service ecosystem and especially in the collaboration lifecycle. At the enterprise level, the trust decisions hook into business decisions weighing collaboration risks and incentives against each other. For this weighing, we use an asset-aware risk vector, which extends the expressive power of the model beyond what is found in other related models. At the ecosystem level, reputation-based trust decisions have an essential role in keeping the ecosystem scalable, utilizing the same mechanisms as social groups [7].

3.1 Trust-related Actions in the Collaboration Lifecycle

In the collaboration lifecycle, two trust-related decision-making points exist to protect the assets of the enterprise:

- First, at the point of commitment on the collaboration contract, after the suggested contract has been formed based on publicly available metainformation.
- Second, at the distrust analysis agent that is part of the collaboration monitoring mechanism. While the collaboration is operational, monitors notify when the expected interaction sequence to and from an involved enterprise is violated. The distrust analysis agent participates in deciding whether the violation is relevant and whether the trust decision needs to be changed.

In the commitment step, the decision-making engine evaluates whether the benefits of participation in the collaboration outweigh the risks. This evaluation can change during the collaboration based on new experience information. Borderline and unclear cases cannot be determined by automation; they are instead forwarded to a human user. This distinction is guided by a metapolicy defining what kind of situations can be considered routine, with clear outcomes [36].

For the monitoring step [21], [25], behavior expectations are collected from the enterprise's local policies and those from the collaboration contract. These are transformed to simple monitor rules, to which all inter-enterprise message exchange must conform. When a deviation is detected, a decision must be taken whether the commitment-time trust assertion has been violated. As reactions to trust breaches, the enterprise can decide to withdraw itself from the collaboration, require renegotiation of the collaboration contract, and create negative experience information both for its local use and potentially also for the ecosystem to distribute. For performance reasons, the interaction sequence is based on tasks containing several services and conversations, not on single message exchanges. The monitoring is private for each enterprise, so it can reflect very different degrees of trusting attitude towards partners and different levels of resource consumption for monitoring. Monitoring protects the enterprise from unexpected external requests, as well as from unwanted requests going out from the enterprise.

In both decision-making places, the decisions are guided by private policies: the decision is always subjective to the authority holding the power of making the decision. A trust decision is made weighing the risk estimation of the situation against the measure for risk tolerance in that situation. Trust is situational, depending on the specific action being decided on. It is also subjective, meaning that the experience information and strategic valuations behind it are local to the trustor, as are the private policies directing the process. The trust information model consists of four central factors: risk, tolerance, reputation and importance. Its metrics are based on the effects of actions to the assets of the enterprise.

In comparison to related work, the Pilarcos trust system combines the strengths of reputation-based (e.g. [12], [47], [43], [14]-[15]) and policy-based (e.g. [29], [1], [4], [48], [5]) approaches to trust management. On one hand, it uses up-to-date reputation information as a basis for risk and benefit estimation, and on the other hand, it combines this information with local business rules and valuations to support flexible decision policies. A central aim of the design has been to separate the policies directing the use of gathered information for trust decisions, and the policies

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

directing the information gathering itself. A further overview on related work is provided in our surveys on trust management [35], trust models [50] and reputation systems [37].

In the next sections, the following essential aspects of the model and prototype implementation are discussed:

- assets
- trust-decision computation
- adjusting the decision by private policies.

3.2 Assets

The goal of the Pilarcos trust management system is to protect enterprise assets: to avoid actions which have a negative effect on them, and to ensure that actions with positive effects are taken whenever possible. Reflecting this, the trust information model expresses risks, risk tolerance, experiences and business importance all through the expected, acceptable, observed and definite effects of actions on assets.

The model defines a set of four assets: monetary, reputation, control and satisfaction (figure 2). The division to multiple assets provides support for policies protecting intangible assets and valuations, such as the privacy of data and the reputation and market share of the enterprise. On the other hand, not leaving assets to be defined by each enterprise separately makes it simpler to share experiences based on the asset model.

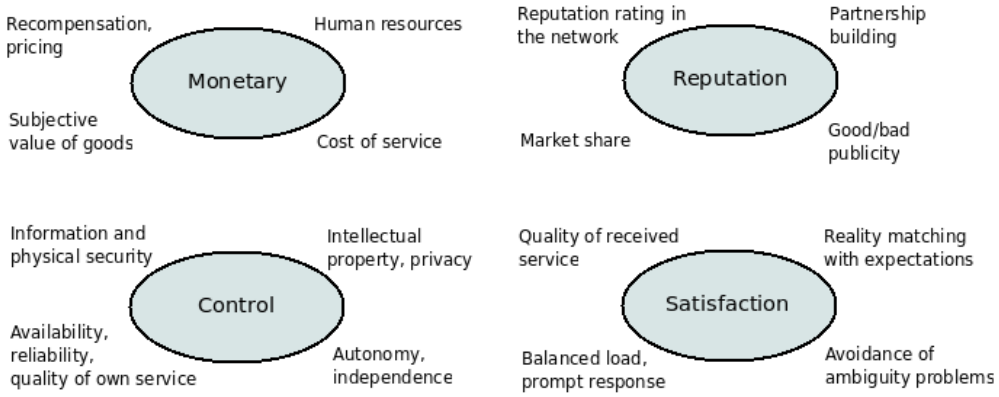

Figure 2: The Four Standard Assets.

The monetary asset represents all assets that are straightforward to measure in monetary terms. Different kinds of resources are committed and gained during collaborations, such as payment for services, storage or computational capacity, staff effort, and produced or consumed goods. The asset forms the basis of decision-making in many situations, and is also often the simplest to measure.

The reputation asset represents the good name, market share and the more technical reputation system status of the enterprise. The attitude of partners and customers towards the enterprise determines its success in initiating and joining collaborations in the future. For this purpose, the enterprise may prefer to avoid participating in seedy projects that threaten to smear its reputation, or to invest in efforts that are likely to bring it more customers.

The control asset encompasses the security of the enterprise, as well as its privacy and autonomy. Loss of control threatens the continued operation of the enterprise, through indirect monetary losses, sensitive information leaks, lawsuits or a loss of freedom to make its own decisions. In decision-making, protecting the control asset entails e.g. being critical of collaboration models which are known to force bad security practices or weakened privacy on the participants, and not agreeing to overly binding contracts. If a contract defines stiff penalty clauses for pulling out from a collaboration session, for example, the threat of penalties will in practice force trust decisions within the collaboration to be positive, even if the partner is clearly misbehaving. This loss of autonomy must be taken into account when deciding to join the collaboration.

The satisfaction asset represents the fulfillment of expectations, i.e. the enterprise gaining what it was trying to gain from a collaboration. Satisfaction measurement forms the basis of most single-asset reputation systems. A contractual breach, low quality of service or general partner incompetence does not always incur notable financial losses, but it has a clear influence on how willing the enterprise is to collaborate with such a partner in the future.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

Satisfaction can also be protected: effort invested in more specific contractual terms and tighter monitoring discourage misbehavior, while choosing partners based on high-quality reputation information rewards good service.

In formal terms, A is the set of guarded assets, represented by integers $0..|A| -1$. When the standard assets of monetary, reputation, control and satisfaction are used, $|A| = 4$. J represents the set of possible outcomes of actions on assets as integers 0...5: 0 for unknown effect, 1 for major negative effect, 2 for minor negative effect, 3 for no effect, 4 for minor positive effect, and 5 for major positive effect. $|J| = 6$. "No effect" differs from "unknown effect". For example, not losing or gaining money would represent a lack of effect, while an experience with a delayed payment on its way might be included as an unknown outcome in decision-making. The trustor defines how a series of events reflects on its assets; the choices on what constitutes a minor or major effect are inevitably subjective.

In comparison to related work, the prevalent approach of for example Bayesian reputation systems ([12], [47], [28], [26]) is to model outcomes only as binary cooperation and defection, i.e. whether the contract was followed or not. This corresponds with the satisfaction asset in our model. Extensions to the limitation of two outcomes have been proposed [11], [32]. However, semantically they still do not measure effects on assets in the sense we do, but either a set of different binary events, or the range of success. The commercial reputation system of the eBay marketplace (Site 1) directly measures satisfaction, on a three-step (positive, neutral, negative) scale.

The SECURE trust model [2] is asset-aware, but as a design choice, all risks are expressed through a single asset. The assumption is that in the target environment, all assets of the private human user can be translated into monetary terms. We find that in the enterprise context, this simplification is not viable.

3.3 Computing Multi-dimensional Trust Decisions

For the trust decisions, two computing tracks are required: calculating a risk estimate vector and a risk tolerance function for the situation. For the actual trust decision, the system then checks that the vector falls within the area of tolerable risk for positive trust decision. The risk tolerance function may also indicate the domain of a gray area, which means that the decision must be forwarded for human decision-making.

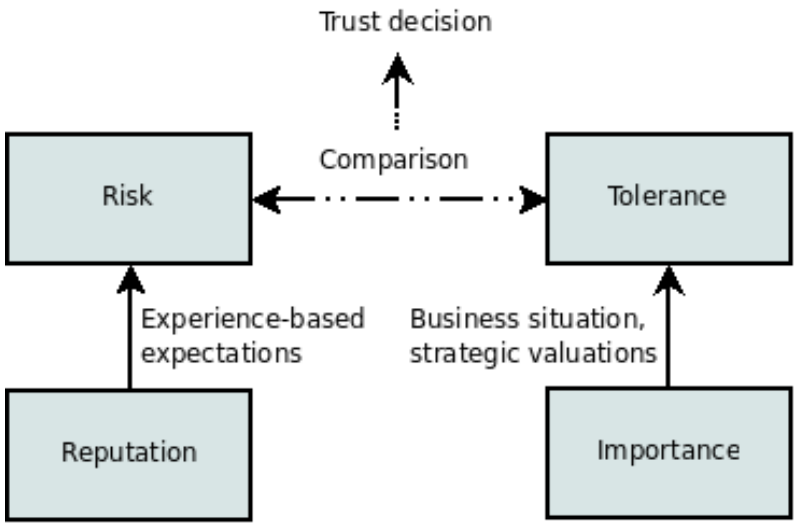

Figure 3: The Factors of a Trust Decision.

The risk estimate is built on a view of the reputation of the trustee, where reputation represents earlier real-world experiences. The risk tolerance builds on the importance of allowing the guarded action because of the business values supporting the involvement in the collaboration. The interrelationships of the factors are presented in figure 3. In the following, both computing tracks are described reflecting their relationship to the assets. First, we present the risk and reputation and the risk tolerance and importance factor pairs, then outline the decision-making step using this information. Presenting the algorithm for calculating the risk vector requires that we also define the structure of the reputation information it is based on. The combination of these two multifaceted factors make its evaluation notably more complex than that of importance and risk tolerance. Once both calculation tracks have been completed, the remaining decision-making step becomes a simple evaluation.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

3.3.1 Risk Vector Computation

The *risk* predicts the trustee's future capability or willingness to correctly deliver the contracted service. The risk value represents the costs of failure or the gained benefits of success as seen by the trusting enterprise. The value always depends on the service in question; the trust model is about trusting a service provided by an enterprise, not about trusting the enterprise as such.

The risk value is therefore an expectation value, where the probabilities of an event are calculated from earlier reputation values, and the cost-benefit factor is dependent on the action, its effects on the assets, and the situation. The action effects can be determined from the design of the business network model. The situational knowledge is covered by the adjustment policies discussed in Section 3.4.

More formally, the risk R of an action contains $|A|$ vectors r_a, one for each asset.

$$R = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{|A|-1}), \text{ where } \mathbf{r}_a = (\mathbf{p}_a, |E|, c, q_a)$$

The vectors store the probabilities of different known outcomes for each asset, and three different measures of the amount and quality of the information used to produce the evaluation. The risk evaluation is specific to a given trustor, trustee and action; we omit references to these three parameters in the formalism for readability.

The first term $\mathbf{p}_a$ is a vector that represents the probabilities of different outcomes: $\mathbf{p}_a = (p_{a,1}, \ldots, p_{a,|J|-1})$, where each $p_{a,j}$ is the calculated probability of outcome j happening for asset a; J is the set of outcome categories {0,1,2,3,4,5} presented in Section 3.2. The latter indexes begin from 1; the probability of an unknown outcome is not considered, but information about unknown outcomes in the past is represented through another term, q_a.

E represents the group of all experiences the trustor has on the trustee in its reputation view, and the number of experiences, $|E|$, measures the amount of reputation information behind the risk estimation. The third term, c, is the combined credibility of local reputation and external, third-party reputation information at the time of evaluation. The last measure, q_a, is the number of experiences in E where the outcome was unknown for asset a: the higher this value is in relation to $|E|$, the lower the certainty of the risk analysis. We return to these quantity and quality measures in transforming reputation information into a risk analysis.

Reputation represents the current view of a trustor's trustworthiness formed from local experience and shared third-party experience. The reputation views building on these two very different sources are stored separately up until the moment of a trust decision.

Local reputation consolidates single experiences gathered by the trustor. These, in turn, are formed by analyzing the output of the Pilarcos monitors [20]. The monitors are not aware of the particular assets being protected in the system, but only detect noteworthy events in the system connected to a business process; the trust management system needs a policy for translating the events into outcomes. An example event could be "product order", with the price or value of the product as a parameter. Further parameters would be the trustee's identifier and the identifier for the action whose business process the event connects to.

External reputation (figure 4) is information received from third parties through a number of reputation networks. All the external reputation views are normalized to the local reputation representation format; the transformation logic is manually designed at the time of accepting a reputation source to be considered in the decision-making process. Also, the source's credibility is given a weight value. In principle, a dynamic credibility learning algorithm is utilized.

In formal terms, the reputation U of a trustee as viewed by a given trustor consists of local reputation U_{local} , and a subjective evaluation of third-party reputation information U_{ext}. The structure of both halves is the same: each contains $|A|$ vectors and a credibility score c^{local} or c^{ext} in the range [0..1].

The credibility value c^{local} of local reputation is 1, while the third-party reputation credibility value c^{ext} is set based on the trustor's analysis of the combined credibility of the reputation system the information comes from, and the credibility of the sources providing reputation information in that network. To avoid repetition, we present the two symmetric reputation structures as generic variables that are equal for both halves, denoting this with an asterisk (*). For example, U* represents both U^{local} and U^{ext}; the formulae are the same for both halves, while the values are different.

Both types of reputation consist of $|A|$ vectors $\mathbf{u}_a^*$, one for each asset:

$$U^* = (\mathbf{u}_0^*, \mathbf{u}_1^*, \ldots, \mathbf{u}_{|A|-1}^*), \text{ where } \mathbf{u}_a^* = (u_{a,0}^*, u_{a,1}^*, \ldots, u_{a,|J|-1}^*)$$

The counters $u_{a,j}^*$ express the number of experiences of outcome j in J for asset a, with $j = 0$ representing an unknown effect.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

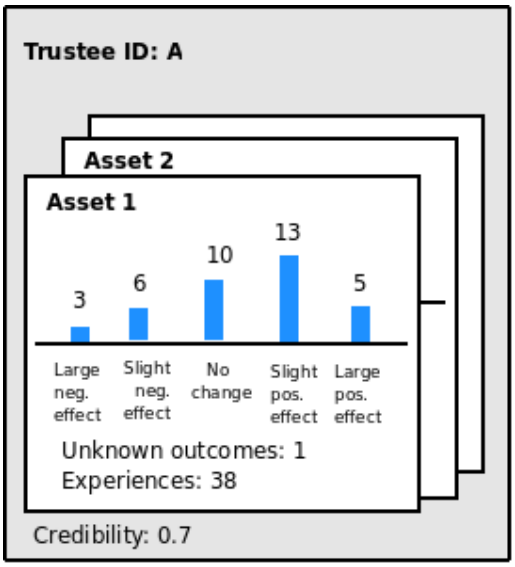

Figure 4: A Sample External Reputation View.

The set of experiences, E^*, is an abstract group

$$E^* = \{ \mathbf{e}_k^* : e_{k,a}^* = \text{outcome value } j \text{ in } J, \text{ for all } a \text{ in } A \}.$$

That is, an experience consists of the effects of one action expressed for each asset. Each experience $\mathbf{e}_k^*$ also belongs to one epoch. The index k ranges from 0 to $|E^*|$-1.

Given the set of experiences, the reputation counters $u_{a,j}^*$ can be expressed as the size of the subgroup of E where the experiences had outcome j for asset a, that is: $u_{a,j}^* = |E_{a,j}^*|$, *where* $E_{a,j}^* = \{ \mathbf{e}_k^* \text{ in } E^* : e_{k,a}^* = j \}$. The value $u_{a,0}^*$ is particularly interesting, as it expresses the number of experiences with unknown values for the asset: we name it q_a^*. When compared to the total number of experiences, $|E^*|$, it provides us with a measure on the quality of the information.

Reputation is transformed into risk in three phases: 1) merging the local and external reputation views together with a weighted sum, 2) scaling the experience counters representing known effects to proportions in the range [0..1] and 3) recalculating a joint credibility and information content score for the result, that is, the variables c, $|E|$ and q_a that appear in the risk vectors.

In the first phase, the local and external reputation views are merged based on the amount of information available in either, and the credibility attached to the views. Local reputation is more credible than external, but there is usually less local information available, and both should be reflected on the weight given to local reputation. We define two functions, μ^{local} and μ^{ext}, to determine the weights for both local and external reputation values. They use the corresponding credibility value c^*, amount of experience $|E^*|$ and a vector $\mathbf{q}^*$ of the number of experiences where the effects are unknown for different assets: $\mathbf{q}^* = (q_0^*, q_1^*, ..., q_{|A|-1}^*) = (u_{0,0}^*, u_{1,0}^*, ..., u_{|A|-1,0}^*)$. The multipliers produced by the μ^* functions add up to 1; the specific declaration of the μ^* functions depends on local calibration.

$$\mu^{local}(c^{local}, E^{local}, \mathbf{q}^{local}) + \mu^{ext}(c^{ext}, E^{ext}, \mathbf{q}^{ext}) = 1.$$

For improved quality of reputation information, we define epochs for the reputation data flows. It is necessary to discount old information in favor of new [13] in order to react to changes in the trustee's behavior. However, the prevalent method of time-based downgrading of reputation history is unoptimal, as it allows a flood of positive experiences from low-value transactions to erase the reputation impact from high-value transgressions.

A trust epoch, in contrast, is defined as a relatively consistent sequence of reputation information. A new epoch is started when there is sufficient evidence for the trustee's behavior having changed, but the information in old epochs is still stored. The weight given to the current epoch determines the speed with which the system reacts to changes in behavior. The number of epochs also measures the consistency of the trustee's behavior.

The algorithm for detecting this change is an interesting research topic. Epoch changes are subjective, as is the reputation information they are locally detected from. The change can be detected from local or sufficiently credible external experience, and it applies to both experience types simultaneously.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

The most suitable policy for epoch changes varies based on the characteristics of the ecosystem. We present here two example policies to achieve two different goals:

Load balancing: a service provider usually provides good service, but occasionally the quality of service varies depending on the amount of incoming requests. The first example policy should quickly react to a drop in the quality of service, as it also indicates a need for load balancing.

Oscillation detection: a service provider oscillates between good and malicious behavior: first it collects good reputation, then it cheats in as many service transactions as it can to maximize its gains before its reputation has been lost. The second example policy should quickly react to this kind of change for the worse, but also be forgiving once the service returns to normality.

In the load balancing example, we can apply a simple dynamically learning algorithm: a window of n previous experiences is stored by the epoch change detector, and whenever a new experience falls outside the values present in the existing filled window, a new epoch is created. As normal service quality is indicated by the vast majority of experiences, the window is typically filled with such experiences. At the first drop in reputation, a new epoch and a new, empty learning window are created. While the disturbance goes on, the window is filling up with negative (or less positive) experiences. During this learning phase, when the epoch contains less than n experiences, new epochs are not created.

If the window (n) is set to be shorter than a typical disturbance, it will be full of negative experiences by the time the service returns to normal load, and a new epoch is started when the first positive experience arrives. This leads to a swift return to the service provider when it is no longer overloaded. For a more pessimistic, slow recovery, the window (n) can be chosen to be longer than a typical disturbance, which means that reputation is slowly gained back within the newest epoch. Again, once the window fills up with normal experiences, the first sign of a negative experience causes a new epoch to be started. A limitation of this policy is that if the experiences indicating normal or overloaded states have some natural variation, new epochs may be created too easily.

In the oscillation detection example, the difference between good and malicious behavior is simple to observe, as the experiences will be polarized: positive or negative. In order to allow a greater variation in behavior than the learning window policy, we apply a static, specification-based epoch detection algorithm. We first define two behavior profiles: “good” and “evil”. The good profile covers positive experiences, the evil profile negative. Neutral experiences, or those representing unknown outcomes, fall in neither category.

Given these profiles, we define each ongoing epoch to be either good or evil, and the epoch changes if an incoming experience matches the opposite profile rather than the current one. Neutral or unknown outcomes do not change the epoch, as they match neither. Again, the ongoing epoch can in principle be given full weight in decision-making. On the other hand, the attacker may respond by oscillating on every service request: cooperate, cheat, cooperate and cheat. To withstand this kind of behavior, the number of epochs or the number of experiences in the current epoch should play a part in choosing a better weight division between the current and previous epochs, or indicate that the decision should really be delegated to a human user due to high uncertainty in the reputation information.

Trust epochs are a novel concept, with no comparable approaches appearing in related work. The typical approach of time-based discounting of old reputation information can be implemented e.g. through adding new experiences to weighted sums, where the weight given to the existing data determines the speed of forgetting old information [12]. Algorithms specializing on oscillation detection have been proposed by e.g. TrustGuard [46], which includes the equivalent of mathematical derivative of the partner’s reputation as a factor in trust decisions. To continue our epoch work, we plan to run simulation experiments to compare the performance of different epoch detection policies, including some more sophisticated learning algorithms.

In the formal model, to support more than one epoch, the μ^* values can be divided beyond the two groups. Merging follows the same pattern, with each *-marked variable appearing separately for each epoch as well.

Using the μ^* functions, we merge the experiences into a temporary $U^{merged} = (\mathbf{u}_0^{merged}, \mathbf{u}_1^{merged}, ..., \mathbf{u}_{|A|-1}^{merged})$, where each vector $\mathbf{u}_a^{merged}$ contains six combined counters $u_{a,j}^{merged}$: the weighed sum of the local and external respective counters.

Note that unlike the values in $\mathbf{u}_a^{local}$ and $\mathbf{u}_a^{ext}$, these merged values are no longer integers, but real numbers. For all a in A, j in J, we have: $u_{a,j}^{merged} = \sum \mu^*(c^*, E^*, \mathbf{q}^*) * u_{a,j}{}^*$.

In the second phase of risk calculation, we scale the experience counters except the unknowns to the range [0..1] to represent probability. To achieve this, we sum the values of known effects ($j \neq 0$) and divide each value by the sum. As a result, we get the $p_{a,j}$ values mentioned earlier in the risk representation. For all j in $J \setminus \{0\}$, $p_{a,j} = u_{a,j}^{merged} / N_a$, where N_a is the sum of $u_{a,j}^{merged}$ over all j in J.

In the third phase, we calculate combined measures of the quality of information: c, $|E|$, and the vector of $|A|$ different q_a values. The combined credibility c is determined by a μ-weighted average of the local and external credibilities. It

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

depicts the weight given to each half in the probabilities as well. $c = \sum \mu^* (c^*, E^*, \mathbf{q}^*) * c^*$.

To calculate the total number of experiences, $|E|$, we sum the number of local and external experiences: $|E|=|E^{local}| + |E^{ext}|$. Although it is clear that not all experiences have been given equal weight in the evaluation, this measure gives an indication of how much information there is available on the actor overall. The combined number of experiences for each asset where the effect was unknown, q_a, is captured by adding the values of the previously calculated $\mathbf{q}^*$ vectors, for all a in A:

$$q_a = q_a^{local} + q_a^{ext} = u_{a,0}^{local} + u_{a,0}^{ext}$$

Again, not all unknowns weigh equally in the probability calculations, so we could consider a μ-weighted average here similarly to the calculation of the credibility value c. On the other hand, the true total number of unknowns is a more useful value to use together with the amount of total experience, $|E|$, as $q_a / |E|$ gives the proportion of uncertain values.

3.3.2 Risk Tolerance Vector Computation

A trustor's *risk tolerance* is determined by the situation calling for a trust decision, independent of the trustee's behavior. While the risk evaluation changes constantly based on new experiences, risk tolerance is only changed when changes in the surrounding business situation call for it.

Risk tolerance depends on the business importance of the action, and local policy expressing the trustor's general risk attitude, encompassing the tolerance of both certain probabilities of various outcomes, and the uncertainty in the information. Tolerance is expressed as a set of constraints for the risk evaluation; if the acceptance constraints are met, the trust decision is positive. The constraints are asset-specific, and can give upper or lower bounds either to probabilities of particular outcomes, the sum of probabilities of a set of outcomes, or the measures of uncertainty. The bounds can be absolute or relative, containing comparisons between probabilities: for example the probability of monetary gain can be required to be larger than the probability of loss.

A trustor's risk attitude determines how risk-averse or risk-seeking the trustor is. A risk-averse trustor will require that an action have high importance to balance for the risk a positive decision would cause, while a risk-seeking trustor can accept a higher risk in relation to the baseline set by the action's importance.

Building a configuration system to help a trustor express their risk attitude through these formulae is an important item of future work. The aim is to bring the level of expression for configurations as close to the business processes and the language of the decision-makers as possible, and minimizing configuration work that requires expensive consultation.

The importance factor expresses the business value of the action, and the cost of a negative trust decision. The costs and benefits do not depend on the expected behavior of the trustee. For example, a negative trust decision blocking an action may result in compensation clauses being activated in the contract between the trustor and trustee. The required compensation may still be small enough that blocking the action is preferable to risking that the trustee causes greater losses by defection.

Importance is expressed in the form of gains or losses to each asset caused by approving the action. Importance information covers the investment required by the action and the guaranteed return of investment, when for example a certain group of actions are considered to be so valuable that requests for them get high priority. To a bank, for example, a cheap loan may be a strategic way to attract customers to move all their banking services to it. Importance should also capture a lack of real choice, should it occur, and more generally the perceived cost of denying service to the trustee. The valuations considered may include the interests of the surrounding business network, adjusted based on how much weight the trustor decides to place on them.

The risk tolerance T of an action, given a particular trustor and trustee, consists of two vectors of $|A|$ functions $f_{x,a}$, one for each asset. The first vector represents the value bounds for automatically accepting an action, while the second vector represents the value bounds for automatically rejecting an action.

$$T = \{ (f_{faccept,0}, f_{accept,1}, \ldots, f_{accept,\,|A|-1}), (f_{reject,0}, f_{reject,1}, \ldots, f_{reject,|A|-1}) \}$$

The functions represent the acceptable limits for the risk values in the risk vectors $\mathbf{r}_a$: they evaluate whether the values are within bounds or not. For all a in A, $f_{a,x}(\mathbf{r}_a)$ is 1 if the values of $\mathbf{r}_a$ are within the bounds, or 0 otherwise.

Risk tolerance depends on the importance of an action. The importance factor I contains $|A|$ values v_a, one for each asset.

$$I = (v_0, v_1, \ldots, v_{|A|-1})$$

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

The values express the known effects a positive trust decision has on different assets: for all *a* in *A*, v_a = an effect value *j* in *J* \ {0}. There are no unknown effects (*j* = 0) for importance: it depicts only those assumed effects and valuations in the enterprise that affect decision-making.

Both importance and risk tolerance depend on the trustor, trustee and action. Risk tolerance is evaluated based on the importance value; each trustor determines the exact evaluation function $F_T(I)$ that produces the risk tolerance T.

3.3.3 Trust Decision

When a trust decision is needed, the two computation tracks described above are executed. With a risk evaluation generated from reputation values, and risk tolerance functions derived from importance values, the actual trust decision is straightforward. The evaluation result of a vector $\mathbf{f}_x$ is a match if the risk tolerance functions within it evaluate to 1, i.e. $f_{x,a}(\mathbf{r}_a) = 1$ for all assets *a* in *A*, and a mismatch otherwise. The action in turn is automatically accepted if the first vector $\mathbf{f}_{accept}$ is matched, automatically rejected if the second vector $\mathbf{f}_{reject}$ is matched, and delegated to a human user otherwise. The division of the risk domain into separate areas for positive ($\mathbf{f}_{accept}$), negative ($\mathbf{f}_{reject}$) and uncertain trust decisions is depicted in figure 5.

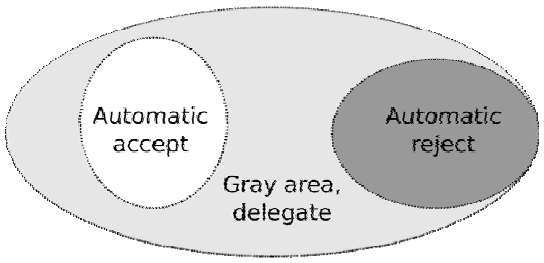

Figure 5: Tolerance Functions Divide the Risk Value Domain into Three Areas.

3.4 Adjusting Trust Decisions for Different Business Situations

The automation of trust decisions needs to be adjusted appropriately to create suitable decision sets for changing business situations. The risks involved in adopting the Pilarcos style of operation include making the wrong automated decisions, or reacting too quickly or slowly to changed reputation information. Risks may also be introduced by creating vulnerabilities in the ecosystem infrastructure.

Changes in business situations to be considered, and discussed below, include

1. changes of business network models or service offers available on the market;
2. attacks towards the enterprise either as business takeover trials through competition, fraudulent reputation manipulation, or security / denial of service attacks towards the enterprise's computing system;
3. privacy threats caused by exposing information or metainformation even within the collaboration, as the partners are capable of retaining the information and allowing it to be forwarded later;
4. inability to fully rely on the ecosystem infrastructure consistency and the trustworthiness of its support elements; and
5. fraudulent behavior by a partner or partners in the ecosystem by first collecting good reputation, then collecting monetary benefits through misbehavior that ruins its reputation.

The mechanisms to either benefit from new situations or to protect the enterprise against threats fall into three categories: metapolicies that guide the decision-making in commitment and distrust decisions, adjustable thresholds for different types of operational situations for positive and negative routine decisions, and finally, building systemic trust towards the infrastructure.

A metapolicy is a policy about when and how a decision can be made by an automated decision-making system according to its internal rules. We have identified four metapolicy categories:

- Strategic orientation of the enterprise. These metapolicies determine whether the activity governed by decision policy is in alignment with the strategic orientation of the enterprise.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

- The trust propensity, i.e. trusting attitude, of the enterprise. These metapolicies determine whether the trust decisions should generally allow optimistic experimentation with new partners, or be more careful and limit collaborations to well-known, strategic partners with high reputation.

- Trust in the elements added by automation. These metapolicies determine whether there is enough systemic trust in the automated decision tools and components supporting them to delegate a given decision. If systemic trust is insufficient, the decision should be made by an actor with higher authorization, a human user.

- The correctness and quality of metadata used by the decision-making system. These metapolicies determine whether the metainformation fed to the decision tools is credible and trustworthy enough to base decisions on. Especially, awareness of the varying credibility of reputation information available for decision-making is critical.

- Privacy policies to determine when information flow cannot be allowed and a collaboration contract must be rejected or breached.

Reflecting the change of market situation (enumerated item 1) requires re-evaluation of how competitive power is best created for the enterprise, and causes changes in the strategic orientation of the enterprise. It is important that such changes in enterprise policy will be briskly represented in the Pilarcos decision-making services. The set of metapolicies can be adjusted to consider all potential business network models, all potential partners, semi-open sets of either, or even, a selected set of business network models or strategic network partners. Thus, the wrong kind of collaborations can be directly discarded without wasting resources for evaluating trust for strategically unworthy collaborations. It should be noted that the metapolicy on strategic orientation enables each enterprise to decide for themselves whether they utilize the ecosystem in the traditional way, relying on the strategic network only, or in a very opportunistic way.

For business level attack scenarios (item 2), lowering the credibility weights for external reputation information sources and raising the thresholds for trusting new collaborations give a reasonable starting point. We have implemented a set of context filters for system administrators to choose from, temporarily adjusting all technical policies to follow this advice by means of additional weight factors, leading to less trusting decisions. For example, the risk tolerance for all non-urgent actions can be drastically reduced, or all trust decisions concerning new actors, with reasonably little reputation information, can be delegated to human users.

Privacy metapolicy governs privacy-affecting activities in collaborations (item 3). From a design point of view, it would be tempting to treat privacy policies as normal policies governing each service or information element, but the nature of privacy preservation is to veto otherwise acceptable actions. Therefore, we raise the privacy policies to the level of metapolicies. For example, it may be the case that the suggested collaboration is interesting, acceptable, and considered on the general level to be trustworthy. However, in the processing it may happen that a service request triggers the need for passing classified documents as part of the service. In this case, it is essential that the privacy classification of information overrides any collaboration agreements, and the individual action of serving a single service request is escalated to human decision-makers. The privacy policies must be attached to all metainformation in addition to the normal payload data.

To manage the distrust of the ecosystem infrastructure weak points (item 4), we have introduced a metapolicy for guiding their use. Metapolicies directing the use of automatically used system-level services addresses a new problem created by Pilarcos-like architectures. The Pilarcos infrastructure allows relaxed matching of service interfaces, and thus, supports automatic configuration of communication channels. The type repository [40], in which interface descriptions and their relationships are stored, also provides references to modifier-interceptors to be placed in the communication channel to for example transform euros to dollars. However, the type repositories may be external to the enterprises, or use externally provided modifier-interceptors; therefore, the trustworthiness of the collaboration can be undermined by that small helping device. These metapolicies should be able to identify which type repositories or which interceptors can be freely used and which should be rejected.

The metadata quality metapolicies reflect the need for suspecting the quality of information in the infrastructure repositories (item 4) and in the reputation information collection process. Local reputation, collected through local monitors, is reliable and high quality, but expensive to gather, as it requires taking the risk of collaborating with the target actor. On the other hand, external reputation, gathered through agents operating in global reputation networks, is less expensive to gather but more unreliable, and more likely to contain errors. The relative weights given to local and external reputation in a risk evaluation are determined by the amount, certainty and credibility of each type of reputation information [37]. The weight is increased as the amount of cases seen with a definite outcome increases (uncertain results are noted as a separate category of outcomes), and the credibility of reputation information providers is followed as their reputation in that role, as seen by their peers. In addition, if a reputation system distributing shared reputation information does not support rigorous source credibility evaluation or distorts information passed through it, its own credibility is set to be low.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

The final change in the business scenario, fraudulent use of collected good reputation (item 5), cannot be solved by metapolicies alone. More detailed trust decision policies are required. Different policies are best suited for different problem situations. What the Pilarcos system can provide is a set of trust decision policies from which the best can be chosen, either manually or by the recommendation of a learning algorithm.

There are three important concerns related to this kind of misuse of collected reputation: First, since reputation is used as a medium of punishing misbehavior, good reputation must be slowly gained but quickly lost. Second, as all transactions do not require equal investment, experiences from low-value actions should preferably be separated from high-value ones when making decisions as well: in eBay terms, actors should not be able to sell a few toothpicks to gain a reputation as a trustworthy car seller. Finally, differentiating between experiences should not lead to information sparsity, i.e. lack of suitable information for making a well-founded decision.

In the trust decision algorithm, a changeable risk tolerance function is used to determine how sharply the decision changes when the incoming reputation information changes. Due to this modular design, the function can be changed at runtime, and provides a good environment for testing how different algorithms protect from these kinds of attacks.

We have simulated the effect of a set of simple risk tolerance formulae:

A:Basic A basic "must not have had more negative experiences than positive experiences" policy: no difference is made between minor and major effects. This policy provides a baseline for comparisons.

B:Pessimistic A more strict policy, "must have had 3 positive experiences per each negative experience". Again, no difference is made between minor and major effects. This policy reflects the need for quickly losing reputation due to misbehavior: three good experiences are needed to cover for each negative experience. The multiplier has been chosen to illustrate the difference already with a small number of experiences.

C:Separative Separating minor and major experiences into their own classes: the number of major positive experiences must be at least equal to the number of major negative experiences, and the same for minor positive and minor negative experiences. The different effect classes are not translated into each other at all. This policy reflects the second concern of differentiating between high- and low-value transactions.

D:Separative-pessimistic Like C, but with triple weight given to negative experiences. This policy combines the first two concerns.

E:Sharp Like A, but with triple weight given to experiences with major positive or negative effects, as opposed to minor effects. This policy is a tradeoff between the third and second concerns: the separative policy may lead to information sparsity, so instead of completely separating the two experience classes, it can transform experiences from one class to the other with a multiplier.

F:Sharp-pessimistic Like B, but with triple weight given to experiences with major positive or negative effects, as opposed to minor effects. This policy reflects the first concern as well as balancing between the second and third concerns.

Our simulations demonstrate that against actively malicious behavior, pessimistic policies are most efficient. Multipliers should be chosen so that expected and acceptable fluctuations in quality of service will not trigger negative trust decisions, but that the overall benefit from opportunistic malicious behavior is too low to be attractive. In a simple collaboration setting with no real attackers, the basic policy is sufficient to tell apart working and broken services, especially when combined with epoch detection. The separative policy performs reasonably well when actions are clearly divided into two categories, but could separate on a different level: not allow a high-value action if the high-value experiences do not support the decision, but still allow low-value actions if the low-value experiences are good. The sharp approach is more forgiving in this sense.

Erroneous reputation information is a major issue for reputation systems. As participation in collaborations depends on a good reputation, actors have a strong motivation to defame competitors or to artificially increase their own reputation. The spreading of misinformation must be punished by a loss of credibility as an information source. False reputation information can be detected e.g. through comparisons to first-hand experiences [47] or comparisons between multiple external sources [8]. In addition, other information on the source, such as its reputation as a service provider [14] or its social relations [43], [10] can also be utilized to estimate its credibility [13].

We have simulated policies on integrating new external reputation information into the system, to handle sources with different credibility. A central concern here is that we want to protect the system against false reputation information, but occasionally the more vulnerable, low-credibility sources are the only up-to-date information source available. It is necessary to quickly react to misbehavior indicated in third-party information to avoid first-hand losses. We survey different approaches to evaluating the credibility of external information sources in earlier work [37]; overcoming the semantic challenges of combining information from multiple sources and different formats is an important research topic as well [9], [16].

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

Finally, fraudulent behavior of partners must be detected also during the collaboration operation, and it should increase distrust in the culprit. For this purpose, the monitoring mechanism is necessary. Each monitor only holds and runs with a set of unrelated simple rules that have been inherited from the local policies and those within the collaboration contract. The policies have initially been created with the involved semantics ranging from business model, technical model, to information model criteria and invariants. However, the monitor is not able to reflect the ruleset back to these principles, but must notify each breach to a more intelligent agent with less strict performance requirements. This agent is then able to decide how relevant the breach is, and whether a distrust situation has been created where the collaboration must be altered or terminated. It should be noted that the policies and metapolicies governing this decision may have changed from the values in use when the collaboration was established. Moreover, the reputation flow may have provided quite a different view of the partners while the collaboration has been active.

In the long run, misbehavior must also be sanctioned by law in order to provide a final deterrent. This requires a new level of legal support for inter-enterprise collaboration, which we will return to in Section 4.

3.5 Reputation Loop for Maintaining the Ecosystem

On the ecosystem level, the essential effect is that a social control loop has been created. The reputation-based trust management concept facilitates the scalability of the ecosystem. Interestingly, we can here rely on social ecosystem studies [7]: the number of potential partners in the ecosystem is limited to very small numbers if there are no established behavior norms, and only slightly higher numbers if there are sanctions for misbehaving. However, if also leaving misbehavior unreported is considered misbehavior in itself, an increasingly large ecosystem can be kept alive. The reputation production mechanism together with the negotiation step where partners can reflect the collaboration suitability for their strategies, their resources, and the potential risk predicted with reputation information, creates a cycle that has this necessary control function. Simply put, it emulates the social or legal system pressure of the business domain. This functionality is sorely missing from other ecosystem approaches.

In the business context, distributed word-of-mouth should be complemented with some centralized information sources. Examples of reputation sources in the current environment include blacklists published by consumer protection authorities or certification schemes (e.g. (Site 2–4)), credit rating information (e.g. (Site 5)), and even stock market fluctuations. Depending on the business domain, different models for spreading reputation information can be adopted [3].

Although the open service ecosystem cannot rely on centralized control, there is no problem with providing reputation information gathering and analysis as third party services: a reputation service does not need to be trusted by all actors, just those for whom the tradeoff between control and investment makes sense, and who find the results relevant to their own sphere of collaborations. Global convergence to a single system is a highly unlikely, and even undesirable phenomenon.

In order for the social pressure from reputation to have any effect, persistent digital identities are needed for the services [34]. On the other hand, strong identity management is already necessary to enter into legally binding contracts, and the creation of new legal entities is controlled. As creating a legal entity capable of signing contracts carries more cost than generating a simple fake service, this also protects against experience distortion by a group of generated drone services, also known as the Sybil attack [6].

4 Discussion

We have proposed a trust management system where autonomous actors make automated local trust decisions based on private policy, while taking advantage of globally shared experiences of the behavior of business peers. The trust decisions can easily be adjusted to different and changing business situations through policy configurations. The trust information model presented extends the currently prevalent models in two directions: first, the narrow single-unit scale for reputation is replaced with an asset-aware model, which provides more expressive power in order to support meaningful risk estimations. Second, the reputation factor is complemented with explicit trustor-dependent risk, importance, risk tolerance and context factors in order to support flexible decision policies. While these extensions also add complexity to the system, they can be used only where needed: the system gracefully degrades to a straightforward reputation-based decision system if the decision-making environment is simple enough to not have use for the full information model.

The open service ecosystem entails an opportunity for creating open marketplaces of services, which adapt quickly to new business needs and allow opportunistic collaboration with new partners. As the ecosystems are not statically fixed to a single business domain or a given set of strategic partners, they can quickly adopt new business models and locate the best service offers available. The ecosystem is also inherently more democratic than the traditional initiator-centric model, which requires all participants to both trust a single hub actor and to modify their systems to be interoperable with those of the initiator of the collaboration.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

However, there are threats and weaknesses to consider in this setting as well, on three different areas: risk of high cost while adopting and managing the ecosystem, infrastructure-level threats introduced by the new global support services, and the current lack of legislative support.

The maintenance costs of the open service ecosystem fall into two categories: the initial investment needed to build the ecosystem, and the administrative costs of its operation.

The initial investment for adopting the ecosystem infrastructure support facilities is high. Setting up the infrastructure requires enterprise architecture changes, and educating personnel in model-driven, service-oriented technology. However, applying the infrastructure does promise significant cost reductions later in the ecosystem lifecycle. The cost reduction is based on the service-oriented modularization of the system and usage of model-based control of collaborations [17]. Thus the ecosystem change management is based on a stable infrastructure layer that can govern collaborations using metainformation and models about the new services and collaboration scenarios.

The administrative costs for supporting the automation of management tasks have also been suspected to be very high: Business network models, service types and service offers must be produced and exported. Further, trust information must be annotated into relevant models of e.g. inter-enterprise collaboration and enterprise risk management, in order to make the information available to automated processing.

These actions are, however, necessary elements of normal design and management processes in enterprises. On these areas, the Pilarcos ecosystem architecture provides an opportunity to move from low-level, technology-dependent expressions to higher-level conceptual work at the business level. For example, rather than setting up very detailed monitoring rules and trust decision policies, we expect that policy-setters will use a high-level policy language to express their goals, and these will then be refined to the lower-level policies that are automatically enforced. Solhaug discusses the user-friendliness of policy languages and policy refinement in more length in his work [44].

Producing the information relevant to trust management does not require a separate modeling round within each enterprise. The business network modeling must be done for contracting and collaboration enactment purposes, and analyzing the risks for any enterprise activities can be based on the same models. Modeling information relevant to trust decisions as a part of, for example, business network modeling makes it possible to enrich the trust decisions without having to manually feed in the same information in multiple places.

Although trust management adds some configuration needs, it also brings opportunities for automatically adapting to changing business situations.

The introduction of new infrastructure-level services will also introduce threats for the ecosystem. For example, a business network model may force poor security practices on the participants, or the third party service in charge of service discovery may provide biased proposals for partners. Reputation sharing introduces an enterprise privacy threat in itself: the organizations will not have full control over what is said about them or their services. This has been suspected to lead to uncontrolled defamation hurting the business on one hand, and the threat of libel lawsuits inhibiting the sharing of negative experiences, on the other. We are currently developing tools and algorithms required for controlling these aspects.

Legislation currently provides no clear rules on the responsibility of software agents. Therefore, agreements made through automated contract negotiations may not yet be legally binding. This problem applies to all ecosystems which are not based on formal blanket agreements between all members of the ecosystem.

First, legislation must be modified to support computational agents making legally binding contractual commitments. It is technically fully feasible to automate contract negotiations based on contract templates, where the terms of service provision are adjusted to fit all members of the proposed collaboration. The resulting contracts must become legally binding, despite having been finalized and enacted by agents. The combination of standardization and legislation should eliminate any need for manual pairwise signing of pre-contracts between all potential partners in the market.

As noted in Section 3.5, the creation of legally acknowledged identities for the business service agents will not only allow them to sign binding contracts. It will also provide a system of persistent identities necessary for accumulating reputation information.

Second, legal support is also needed for the partially as well as fully automated exchange of experience information on these agents. As these experiences form a reputation for services, even valid negative experience reports cause an opening for defamation charges, vengeful business tactics and even retaliatory negative reports [33]. On the other hand, the existing services for providing e.g. credit ratings and consumer protection blacklists have not been toppled by the threat of lawsuits, and anonymous reputation systems have been proposed for sharing experiences in sensitive situations [37].

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

In the end, legislation cannot solve all the problems that reputation systems suffer from, similarly to how technology cannot solve all social problems. It is important to apply social means to a social problem: the spreading of false reputation information must be punished by loss of reputation, in order to create social pressure to not only follow rules, but also to altruistically punish those who do not follow the rules [7]. In our system, this kind of reputation as an information source is covered by the credibility measure presented in Section 3.4.

In summary, the key strengths of the open service ecosystem lie in its openness and flexibility: the ecosystem itself is open both to new business models and to new actors, and it is scalable. There is no single reputation information source but multiple, and each enterprise can choose which reputation source to connect to.

The Pilarcos infrastructure allows opportunistic behavior and experimenting on collaborations with new partners made available in the ecosystem. On the other hand, it also allows collaborations to be set up based on traditional strategic networks as well; policies approving only specific well-known partners are by no means prohibited. The ease of setting up new collaborations supports both niche market exploitation for large companies, and alternative company liaisons for subcontractors, which have been traditionally bound to their contractor.

The impact of the trust management system can be seen on two levels:

- On the business ecosystem level, social control through reputation keeps the market functional despite the presence of misbehaving peers.
- On the single business level, trust decisions allow the enterprise to control its own risk-taking, while shared reputation information in particular helps it learn from others' mistakes as well in order to avoid misbehaving partners.

The sharing and use of experience information introduces social control into inter-enterprise collaboration in the open service ecosystem. Shared experiences form a computational equivalent of reputation: those who are caught misbehaving suffer damage to their reputation, while those who correctly report this misbehavior gain positive reputation. The reputation damage, in turn, warns off other actors not to collaborate with the misbehaver, which limits the overall damage they can cause.

Acknowledgments

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

Websites List

Site 1: eBay
http://www.ebay.com/

Site 2: Better Business Bureau
http://www.bbb.org/

Site 3: National Consumer Agency of Ireland
http://www.consumerconnect.ie/

Site 4: Trustmark Tradesman Certification
http://www.trustmark.org.uk/

Site 5: Standard & Poor's website
http://www.standardandpoors.com/

References

[1] M. Blaze, J. Feigenbaum, and J. Lacy, Decentralized trust management, in Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California. IEEE, 1996, pp. 164-173.
[2] V. Cahill, E. Gray, J-M. Seigneur, C.D. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. Di Marzo Serugendo, C. Bryce, M. Carbone, K. Krukow and M. Nielson. (2003, August). Using trust for secure collaboration in uncertain environments, Pervasive Computing. [Online]. vol. 2, no. 3, pp. 52-61. Available: http://ieeexplore.ieee.org/iel5/7756/27556/01228527.pdf.
[3] D. Chadwick, Operational models for reputation servers, in Proceedings of the 3rd International Conference on Trust Management, Rocquencourt, France, 2005, pp. 108-115.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

[4] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. (1997, September) REFEREE: Trust management for Web applications, Computer Networks and ISDN Systems. [Online]. vol. 29, no. 8-13, pp. 953-964. Available: http://dx.doi.org/10.1016/S0169-7552(97)00009-3.
[5] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, The Ponder policy specification language, in Proceedings of the International Workshop on Policies for Distributed Systems and Networks, New York, 2001, pp. 18-38.
[6] J. R. Douceur, The Sybil attack, in Proceedings of the 1st International Workshop on Peer-to-Peer systems. Cambridge, MA, USA, 2002, pp. 101.
[7] E. Fehr and U. Fischbacher, The nature of human altruism, Nature, vol. 425, October 2003. [Online]. Available: http://dx.doi.org/10.1038/nature02043.
[8] A. Fernandes, E. Kotsovinos, S. Östring, and B. Dragovic, Pinocchio: Incentives for honest participation in distributed trust management, in Proceedings of the 2nd International Conference on Trust Management, Oxford, UK, 2004, pp. 64-77.
[9] N. Gal-Oz, T. Grinshpoun, E. Gudes, and I. Friese, TRIC: An infrastructure for trust and reputation across virtual communities, in Proceedings of the 5th International Conference on Internet and Web Applications and Services. Barcelona, Spain. IEEE, 2010, pp. 43-50.
[10] N. Gal-Oz, E. Gudes, and D. Hendler, A robust and knot-aware trust-based reputation model, in Trust Management II, vol. 263, (Y. Karabulut, J. Mitchell, P. Herrmann, and C. D. Jensen, Eds.). Pisa, Italy: Springer, 2008, pp. 167-182.
[11] A. Jøsang and J. Haller, Dirichlet reputation systems, in Proceedings of the 2nd International Conference on Availability, Reliability and Security. Vienna, Austria. IEEE Computer Society, 2007, pp. 112-119.
[12] A. Jøsang and R. Ismail, The Beta reputation system, in Proceedings of the 15h Bled Electronic Commerce Conference, Bled, Slovenia, 2002, pp 324-337.
[13] A. Jøsang, R. Ismail, and C. Boyd. (2007, March). A survey of trust and reputation systems for online service provision, Decision Support Systems: Emerging Issues in Collaborative Commerce. [Online]. vol. 43, no. 2, pp. 618-644. Available: http://dx.doi.org/10.1016/j.dss.2005.05.019.
[14] S. Kamvar, M. Schlosser, and H. Garcia-Molina, The EigenTrust algorithm for reputation management in P2P networks, in Proceedings of the Twelfth International World-Wide Web Conference, Budapest, Hungary, 2003, pp. 446-458.
[15] M. Kinateder and K. Rothermel, Architecture and algorithms for a distributed reputation system, in Proceedings of the 1st International Conference on Trust Management, Greece, 2003, pp. 1-16.
[16] M. Kinateder, E. Baschny, and K. Rothermel, Towards a generic trust model - comparison of various trust update algorithms, in Proceedings of the 3rd International Conference on Trust Management, Rocquencourt, France, 2005, pp. 177-192.
[17] L. Kutvonen, Automated management of interorganisational applications, in Proceedings of the Sixth international conference on Enterprise Distributed Object Computing (EDOC '02), Lausanne, Switzerland, 2002, pp. 27–38. [Online]. Available: http://www.cs.helsinki.fi/group/pilarcos/deliverables/kutvonen_management_edoc_2002.pdf.
[18] L. Kutvonen, Building B2B middleware - interoperability knowledge management issues, in Enterprise Interoperability II - New Challenges and Approaches. Funchal, Portugal: Springer, 2007, pp. 629-632.
[19] L. Kutvonen, J. Metso, and S. Ruohomaa. (2007, July). From trading to eCommunity management: Responding to social and contractual challenges, Information Systems Frontiers (ISF) - Special Issue on Enterprise Services Computing: Evolution and Challenges. [Online]. vol. 9, no. 2-3, pp. 181-194. Available: http://dx.doi.org/10.1007/s10796-007-9031-x.
[20] L. Kutvonen, J. Metso, and T. Ruokolainen, Inter-enterprise collaboration management in dynamic business networks, in Proceedings of Confederated International Conferences on the Move to Meaningful Internet Systems. Agia Napa, Cyprus. 2005, pp. 593-611.
[21] L. Kutvonen, T. Ruokolainen, and J. Metso. (2002, 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. [Online]. vol. 3, no. 1, pp. 1-21. Available: http://www.idea-group.com/articles/details.asp?id=6597.
[22] L. Kutvonen, T. Ruokolainen, S. Ruohomaa, and J. Metso, Service-oriented middleware for managing inter-enterprise collaborations, in Global Implications of Modern Enterprise Information Systems: Technologies and Applications, A. Gunasekaran, Ed. Hershey, PA: IGI Global, 2008, pp. 209-241.
[23] M.-S. Li, M. Kürümlüoglu, M. Mazura, and R. van den Berg, Future Internet Enterprise Systems (FInES) cluster position paper, EU FP7 FInES cluster, Technical Report, 2009. [Online]. Available: http://cordis.europa.eu/fp7/ict/enet/fines-positionpaper en.html.
[24] N. Mehandiev and P. Grefen, Eds. Dynamic Business Process Formation for Instant Virtual Enterprises. New York: Springer, 2010.
[25] J. Metso and L. Kutvonen, Managing virtual organizations with contracts, in Electronic proceedings of the Workshop on Contract Architectures and Languages (CoALa2005), Enschede, The Netherlands, 2005.
[26] L. Mui, M. Mohtashemi, and A. Halberstadt, A computational model of trust and reputation, in Proceedings of the 35th Annual Hawaii International Conference on System Sciences. Waikoloa, Hawaii. IEEE Computer Society, 2002, pp. 188.
[27] T. J. Norman, A. D. Preece, S. Chalmers, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, W. A. Gray, and N. J. Fiddian. (2004, April). Agent-based formation of virtual organisations, Knowledge-Based Systems. [Online]. vol. 17, no. 2-4, pp. 103–111. Available: http://dx.doi.org/10.1016/j.knosys.2004.03.005.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 118-136

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200008

[28] P. Nurmi, A Bayesian framework for online reputation systems, in Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services. Guadeloupe, French Caribbean. IEEE Computer Society, 2006, pp. 121.
[29] OASIS Web Service Secure Exchange TC, WS-Trust 1.3 OASIS Standard, OASIS, March 2007. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.pdf.
[30] R. J. Rabelo, S. Gusmeroli, C. Arana, and T. Nagellen, The ECOLEAD ICT infrastructure for collaborative networked organizations, in Network-Centric Collaboration and Supporting Frameworks, vol. 224, (L. M. Camarinba-Matos, H. Afsarmanesb and Martin Ollus, Eds.). New York: Springer, 2006, pp. 451-460.
[31] L. Rasmusson and S. Jansson, Simulated social control for secure Internet commerce, in Proceedings of the Workshop on New Security Paradigms, California, 1996, pp. 18-25.
[32] S. Reece, A. Rogers, S. Roberts, and N. R. Jennings, Rumours and reputation: evaluating multi-dimensional trust within a decentralized reputation system, in Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-agent Systems, Honolulu, Hawaii, 2007, pp. 1063-1070.
[33] P. Resnick and R. Zeckhauser, Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system, in The Economics of the Internet and E-Commerce, vol. 11, M. R. Baye, Ed. Amsterdam: Elsevier Science, 2002, pp. 127-157.
[34] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. (2000, December). Reputation systems, Communications of the ACM. [Online]. vol. 43, no. 12, pp. 45-48. Available: http://doi.acm.org/10.1145/355112.355122.
[35] S. Ruohomaa and L. Kutvonen, Trust management survey, in Proceedings of the 3rd International Conference on Trust Management, Rocquencourt, France, 2005, pp. 77-92.
[36] S. Ruohomaa and L. Kutvonen, Making multi-dimensional trust decisions on inter-enterprise collaborations, in Proceedings of the 3th International Conference on Availability, Security and Reliability. Barcelona, Spain. IEEE Computer Society, 2008, pp. 873-880.
[37] S. Ruohomaa, L. Kutvonen, and E. Koutrouli, Reputation management survey, in Proceedings of the 2nd International Conference on Availability, Reliability and Security. Vienna, Austria: IEEE Computer Society, 2007, pp. 103-111.
[38] S. Ruohomaa, L. Viljanen, and L. Kutvonen, Guarding enterprise collaborations with trust decisions—the TuBE approach, in Proceedings of the Workshops and the Doctoral Symposium of the Second IFAC/IFIP I-ESA International Conference, Bordeaux, France, 2006, pp. 237–248.
[39] T. Ruokolainen and L. Kutvonen, Addressing autonomy and interoperability in breeding environments, in Network-Centric Collaboration and Supporting Frameworks, vol. 224, (L. Camarinha-Matos, H. Afsarmanesh and M. Ollus, Eds.). Boston: Springer, 2006, pp. 481-488.
[40] T. Ruokolainen and L. Kutvonen, Service typing in collaborative systems, in Enterprise Interoperability: New Challenges and Approaches (G. Doumeingts, J. Müller, G. Morel, and B. Vallespir, Eds.). London: Springer, 2007, pp. 343–354.
[41] T. Ruokolainen and L. Kutvonen, Managing interoperability knowledge in open service ecosystems, in Enterprise Distributed Object Computing Conference Workshops, V. Tosic, Ed. Hershey, PA: IGI Global, 2009, pp. 20-211.
[42] T. Ruokolainen and L. Kutvonen, Managing non-functional properties of inter-enterprise business service delivery, in Service-Oriented Computing - ICSOC 2007 Workshops, vol. 4907, (Nitto and M. Ripeanu, Eds.). Heidelberg, Berlin: Springer, 2009, pp. 90-92.
[43] J. Sabater and C. Sierra, Reputation and social network analysis in multi-agent systems, in Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems, Bologna, Italy, 2002, pp. 475-482.
[44] B. Solhaug, Policy specification using sequence diagrams. Applied to trust management, Ph.D. dissertation, University of Bergen, Norway, 2009.
[45] D. J. Solove (2006, January), A taxonomy of privacy, University of Pennsylvania Law Review. [Online]. vol. 154, no. 3, pp. 477-560. Available: http://ssrn.com/abstract=667622.
[46] M. Srivatsa, L. Xiong, and L. Liu, TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks, in Proceedings of the 14th International Conference on the World Wide Web. New York, 2005, pp. 422-431.
[47] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. (2006, March). TRAVOS: Trust and reputation in the context of inaccurate reputation sources, Autonomous Agents and Multi-agent Systems. [Online]. vol. 12, no. 2, pp. 183-198. [Online]. Available: http://www.springerlink.com/content/2h56k13n37qk0274/.
[48] A. Uszok, J. M. Bradshaw, and R. Jeffers, KAoS: A policy and domain services framework for grid computing and Semantic Web services, in Proceedings of 2nd International Conference on Trust Management, Oxford, UK, 2004, pp. 16-26.
[49] L. Viljanen, A survey on application level intrusion detection, University of Helsinki, Department of Computer Science, Techical Report, 2005.
[50] L. Viljanen, Towards an ontology of trust, in Proceedings of the 2nd International Conference on Trust, Privacy, and Security in Digital Business, Copenhagen, Denmark, 2005, pp. 175–184.
[51] M. Wilson, A. Arenas, D. Chadwick, T. Dimitrakos, J. Doser, P. Giambiagi, D. Golby, C. Geuer-Pollman, J. Haller, K. Stølen, T. Mahler, L. Martino, X. Parent, S. Ristol, J. Sairamesh, L. Schubert, and N. Tuptuk, The TrustCoM approach to enforcing agreements between interoperating enterprises, in Proceedings of Interoperability for Enterprise Software and Applications Conference. Bordeaux, France: Springer-Verlag, 2006, pp. 365-376.