

Reflective federation of enterprises in open service ecosystem

Lea Kutvonen

*Department of Computer Science
University of Helsinki, Finland
Email: lea.kutvonen@cs.helsinki.fi*

Abstract—The present trend of enterprise computing is towards networked business, and thus there is a high demand on a new layer of global infrastructure that facilitates inter-enterprise collaboration management and provides utilities for interoperability. This paper discusses the Pilarcos open ecosystem architecture; and especially, its reflective infrastructure that supports runtime establishment of collaborations, and consistent evolution of the ecosystem itself.

Keywords—open business service ecosystem; enhancement and application of ODP concepts; reflective infrastructure

I. INTRODUCTION

The present trend of enterprise computing is towards networked business, and therefore solutions that capture both business needs and computing solutions are needed. Furthermore, automating the collaboration life-cycle management but preserving independence of collaborating parties is needed.

A common way of describing the interplay of business and computing solutions is through enterprise architecture work. The enterprise architecture frameworks, such as Zachman [1] or TOGAF [2], establish methodologies for modeling enterprise systems joining these aspects, and provide support for decision-making when changes either to the computing part or the business part are expected. However, these frameworks do not prescribe any runtime infrastructure for managing the alignment or the changes. Moreover, traditional applications of these frameworks do not take into consideration the needs of (dynamically changing) collaboration partners.

Later, collaboration support infrastructures, models, and breeding environments for virtual enterprises (or for business networks, as we call them in this paper) have been introduced for solving the rising needs. Examples include ECOLEAD [3], Athena [4], and CrossWork [5]. Besides service-oriented architecture basis, these projects also address the system and service software engineering aspects in such environment, based on model-driven techniques. Most solutions focus on establishing collaborations and ensuring that the collaboration is interoperable in terms of technical messaging and shared semantics of the exchanged information and service calls.

In this paper, we have however considered an open ecosystem in which business networks, i.e. inter-enterprise

collaborations are designed, established, controlled, terminated, and evaluated – covering the full life-cycle of collaborations. This Pilarcos ecosystem [6] is supported by a new layer of global infrastructure that facilitates collaboration management and provides utilities for interoperability.

We have adopted a reflective and federative way of supporting the evolution of the ecosystem, thus keeping the change process scalable. Moreover, the ecosystem architecture must utilise management processes where ownership and decision-making authority is not guaranteed, because enterprises do not own all relevant artefacts and activities. We have adopted a multi-agent system pattern to be combined with the reflective model to overcome this difficulty.

A reflective system is a system that can reason about its own structure, behaviour and state, and furthermore, can act upon itself. For reasoning and action triggering, the system includes a causally connected metalevel model of its own characteristics, making it able to detect changes in itself or its environment, make decision based on that information, and change its own status or create interaction with its environment. To quote the original definition [7]: "We define computational reflection to be the behavior exhibited by a reflective system, where a reflective system is a computational system which is about itself in a causally connected way". The concept has successfully been applied on context sensitive and adaptive applications, or rather, on the middleware level that supports such applications: reflective middleware [8].

A federative system is a system that spans several administrative domains, each of which has a controller to manage the relevant activities within its own domain. To quote the ODP reference model [9]: "<X> federation as a community of <X> domains where there is a shared objective" [clauses 5.1.2, 5.1.1], and "<X> domain is a set of objects with a shared controlling object over the characteristic feature X" [clause 10.3]. As the Pilarcos architecture has been developed in alignment of the concepts defined in ODP reference model [9], [10], the relationship of further concepts (such as community, contract and liaison, type and template, service and policies) have been described in earlier papers [11], [12].

This paper illustrates how the use of runtime reflection concepts in the open ecosystem infrastructure of Pilarcos provides safe, consistent and evolvable support of business

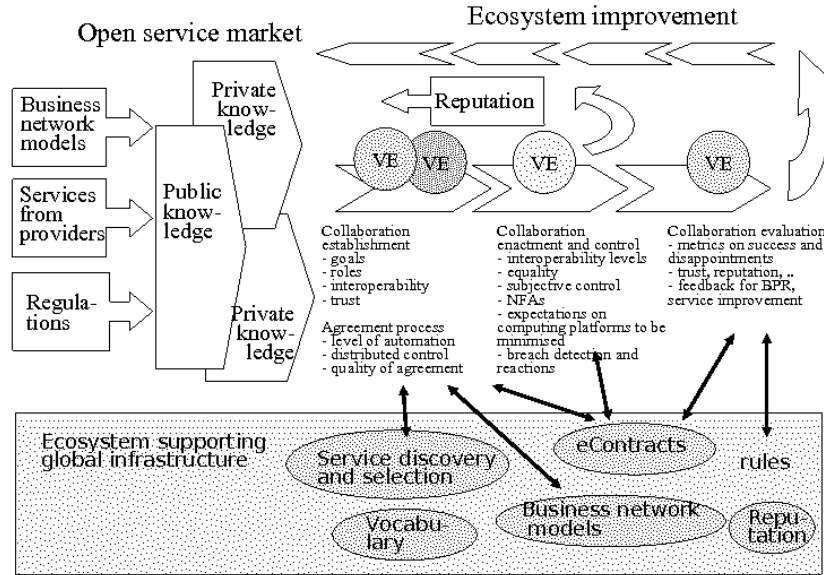


Figure 1. Overview of the open business service ecosystem. VE denotes virtual enterprise, i.e. business network.

networks and the whole of the ecosystem, still allowing each ecosystem and collaboration member to adapt it according to their own needs.

Section 2 provides an overall view of the service ecosystem, while Section 3 introduces it from the reflective system model perspective. Section 4 concludes by discussing related work and achievements.

II. OPEN BUSINESS SERVICE ECOSYSTEM

In biology, an ecosystem means an environment where flora and fauna lives and dies, utilises resources and interact with each other, and eventually, new species can develop. This behaviour is restricted by laws of nature, or by for example human interventions.

Similarly, inter-enterprise collaborations that are governed by multi-party contracts can be bread, run their natural life-cycle, be terminated, and leave experience knowledge behind. We call such an environment *open business service ecosystem* since we choose to focus on service-oriented computing and keep business services aligned with their software counterparts, and vice versa.

Figure 1 illustrates the ecosystem structure. On the left side, metainformation is brought to the system by designers and analyzers concerning

- available services from service providers (enterprises including public and private sector providers),
- publicly known business scenario descriptions that we call business network models (compositions of sets of business process descriptions, with required associations between roles to be simultaneously occupied);

- regulations for conducting collaboration at administrative domains;
- reputation information that allows terminating collaborations to report failures and successes in the ecosystem to advise further collaboration decisions.

The bottom part represents the global, federated infrastructure services that provide services for business service and partner detection and selection, econtract establishment, monitoring and breach detection, and reputation management facilities to support trust management with. The top part represents the life-cycle of each collaboration (virtual enterprise) from negotiation to termination phase.

The different ecosystem building projects we have studied differ in automation and control aspects of negotiation, operational time monitoring, and feedback generation for private and public collaboration and interoperability knowledge bases.

In Pilarcos, the negotiation phase is "democratic" as all potential partners can participate in the business network model definition process, and the service discovery process. Only when the final decision-making phase including final commitment to the formulated contract is performed, the decisions are made at each partner according to private rules. In contrast, ECOLEAD and CrossWork allow a coordinator to do the selection, and the selected partners start negotiating about the roles and interaction patterns suitable for the business case at hand and the competencies of the partners. In this phase, Pilarcos utilises more automation, while in the mentioned comparison architectures, the role

of infrastructure is to support human decision-making.

In Pilarcos, the regulations governing collaborative behaviour are considered to be independently declared at each administrative domain. Therefore, collaborating partners from different domains may have contractual discrepancies with each other, although they do all follow their local regulations. Because regulatory systems are changing and international collaborations cause mismatches, it is important that the runtime support system is able to detect breaches from each partners point of view separately.

Furthermore, an important feedback aspect in the Pilarcos ecosystem is that of reputation information generation. The reputation-based trust management concept facilitates the scalability of the ecosystem. Interestingly, we can here rely on social ecosystem studies [13]: the number of potential partners in the ecosystem is very limited if there are no established behavior norms, and only slightly higher if misbehaviour is sanctioned. However, if also leaving misbehaviour unreported is considered as misbehaviour, 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, it emulates the social or legal system pressure of business domain. This functionality is much missing from other approaches.

The main five challenges in Pilarcos ecosystem have been as follows:

- support functions to match the management needs; including service discovery and selection, eContracting, and breach management [14];
- support of ecosystem evolution by a coherent knowledge base; this includes the control of collaboration types and available services and interoperability management information [15], [16], [17];
- support of regulation of collaborations in such a way that only acceptable collaboration types are allowed, and by controlling the behaviour of services through contracts and enterprise policies [16];
- support of private decision-making on collaborations; this includes enterprises' expert system for making decisions related to contracts, breaches and trust [18];
- global software application system production; this involves production methods for service software and coordination models, and definition of open service ecosystem quality requirements for software.

III. REFLECTIVE ACTIVITIES

The Pilarcos architecture includes features of self-adaptive systems and complex adaptive systems.

A self-adaptive system evaluates its own behaviour and takes corrective actions when the evaluation indicates the system is not behaving according to expectations.

A complex adaptive system is a dynamic network of agents acting in parallel, reacting to each others actions. The control of the system is decentralised, and the coherence of behaviour in the system rises from competition and collaboration of the agents. Figure 2 shows how the complex system interacts with its changing environment and emerges by learning based on feedback.

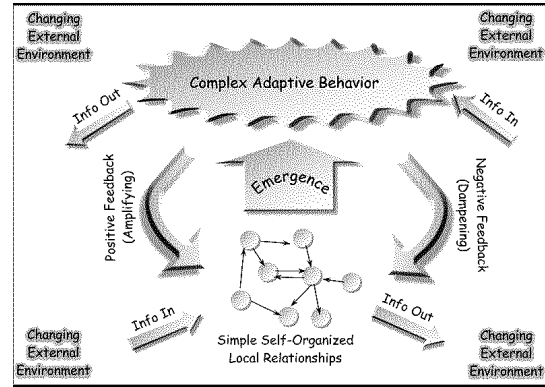


Figure 2. Overview of complex adaptive system [19].

In the Pilarcos architecture, the foci of reflective activities are i) the ecosystem itself, ii) each business network, and iii) each enterprise as the ecosystem members.

A. Ecosystem reflection

The ecosystem purpose is to enable business networks to be established, operated, terminated and experience information to be gathered for evolving the system further. Therefore, the ecosystem core is the infrastructure services for service selection, eContract management, breach management, and reputation-based trust system, utilising members, and regulations on behaviour in the ecosystem. Each member of the ecosystem is represented by a private agent that is capable of interacting with the infrastructure services and other similar agents.

The infrastructure services hold the metainformation in public repositories that are federated for scalability and for maintaining independent but overlapping markets. The ecosystem life-cycle requires coherent models and metainformation to be fed into the infrastructure knowledge repositories, describing business network models, available services, reputation, and furthermore, service type information to form a shared ontology or vocabulary for the other categories of items. The reflective activities of business networks and enterprises can only succeed, if the models available conform to the Pilarcos requirements and are consistently related to each other.

The ecosystem repositories (service offer repository, service type repository, business network model repository) are governed by four ontology metamodels [16], [17]:

- domain ontology metamodel that defines basic ecosystem concepts like collaboration, service, and contract;

- methodology metamodel that defines phases of service engineering processes and especially the produced artefacts;
- domain reference metamodel that captures the infrastructure elements by defining the operational time support functions and artefacts manipulated;
- knowledge management metamodel that defines language on storing each knowledge item and relationship.

The ontology metamodels are interlinked so that a basic concept can be connected to its representation format in a methodology and a operational time infrastructure function, and has a technical storage representation. Thus, the design and production time artefacts become also artefacts at the operational time.

In terms of a complex reflective system, the ecosystem can be viewed as follows. First, the ontologies built allow the system to decide when an addition to its knowledge base is acceptable or not, either accepting evolution or rejecting it. Interesting activities are those where designers try to modify the knowledge base by auditing new business network models, new service types or new service offers. The ecosystem will check the proposed models against its knowledge base, and may decide to reject the proposition because of discrepancies with existing models, incomplete modeling, or lacking traceability of the publisher of the model. In practice, when designers and analyzers try to enter information items and models to the repositories, the corresponding metamodel layers will be used to study, whether the suggestion is correct. If not, the designers can be given advice on how to improve the proposal. The designers may also study existing artefacts in the repositories, assess their relationships and create relationship links between the artefacts. For these relationships, similar modeling layers exists, and thus, at this phase too, advice on analysts may be provided.

Second, the ontologies can be grown on the metamodel level as the ecosystem evolves. The introduction of new concepts requires definition of new language concepts and their relationships to the already existing concepts on the same meta-level. The upper metamodel levels provide sufficient rules for allowing this kind of change in the repository content rules.

Third, the ecosystem members are expected to monitor the successes and failures of business networks they participate and share experience on collaborating with each other. This information is flooded through the ecosystem, which provides the expected feedback loops for learning about evolution needs.

Finally, the knowledge base must be technically distributed for two reasons: technical scalability and need for separate alliances at the business level. Thus the ecosystem must be able to adopt new repository and infrastructure function providers, and furthermore, support member enterprises in choosing which providers to trust for these basic services.

Semantically the content of the repositories can be federated based on the coherence of uppermost concept models across the repositories.

B. Contract-based governance of business networks

A business network is an identified, isolated collaboration that governed by eContract [14]. The eContract is structured according to the roles defined in the selected business network model. The services playing parts in the collaboration are picked from the service offers available as directed by the initiator of the collaboration and as dictated in the business network model selected by the initiator.

The service selection is supported by a populator function that ensures the suggested (or proposed) set of services match to the roles for their service types, are not denied to work together by regulations, and are interoperable on technical, semantic and pragmatic levels. The populator matches all roles simultaneously, instead of using a client-server type pairwise matching.

This initial selection is based on public information, and therefore, private decision-making is incorporated to a negotiation phase. In the negotiation phase each suggested collaborating party can agree to join the collaboration, or refrain. Population and negotiation phases are repeated as necessary.

At the end of a successful negotiation phase, the eContract is formed. The eContract captures the business network model, players in each role, requirements for communication channels between services, and requirements for nonfunctional properties of the collaboration as well as policies providing invariants the collaborative behaviour should hold.

During the operation of the business network each member service monitors the acceptability of the behaviour (messaging) it can directly assess. The assessment is based on the eContract contents and local enterprise policies. The rules derived from these two sources can be contradictory. At the negotiation phase only those policies are checked that are explicated both in the eContract and in the enterprise policies; moreover, the enterprise policies can change during the collaboration without consulting the collaboration peers.

When an essential breach against the eContract is detected by one of the collaboration members, the eContract agent is notified. The eContract agent then starts recovery steps, for example, terminating the collaboration, or changing the faulty member to a new one. The recovery capabilities are dependent on the business network model, and therefore, the breach recovery process is defined as part of the business network model itself.

As a collaboration terminates, the members are expected to provide feedback information. Especially experience information is necessary to produce reputation information flow.

In the construction of the infrastructure model for the open business service ecosystem, the reflection model has been

used together with a multi-agent system model. Figure 3 illustrates the situation as follows.

Here, the topmost model "system model" refers to the business network system model embedded to an eContract after it has been established and committed to by all parties. This model, i.e. contract, can be changed by a) human administrators at the involved enterprises requesting so through their expert systems, or b) asynchronous event arisen by the underlying system, i.e. any of the services themselves detecting a situation requiring attention. An attention requiring event is such that does not conform to the criteria set in the contract (either by a service itself detecting a failure or someone else in the business network detecting a breach). This event can be handled by the contract (agent) automatically or the decision can be forwarded for human intervention (necessary to control the amount of automation in presence of business and legal consequences of actions).

Viewing the managed system itself, the situation becomes more complicated, as the business services involved are not under the control of a single authority, but belong to separate, independent administrative domains. Thus, the willingness to perform actions, capability and implementation of the actions, and representation of the information related to the expected actions may differ significantly. We do not suggest harmonisation of the system and platform level, since one of the main goals of the architecture is to allow evolution - also on this technical level.

We can require the causal connection to be held between the eContract and the business network structures and criteria, but cannot assume that the causal connection would be completely unified. Therefore, the solution has two parts. First, for each role in the contract there is a causal connection with the supporting business service at the providing enterprise. Second, the contract agent forms a shared-vocabulary messaging environment for the agents representing participating enterprises. These enterprise agents are responsible of keeping their local services in synchrony with the committed contract status; here we use the reflective model again to allow a variety of techniques to be utilised locally. Between themselves the enterprise agents need to use protocols familiar from multi-agent systems or speech act theories: definitions and declarations, requests, suggestions, commitments, and opinions.

It is inevitable that at some point of time the enterprise agents either fail to manage their local services as promised, do not agree on the contract status or changes to it, or cannot find a way to resolve a breach detected in the underlying system. This is normal in business, so it should not be considered as faulty behaviour of the supporting software either.

The remaining block of design is that of monitoring collaboration behaviour. Each enterprise has their local policies and in addition, the contract brings in a set of policies and processes specific for the collaboration instance

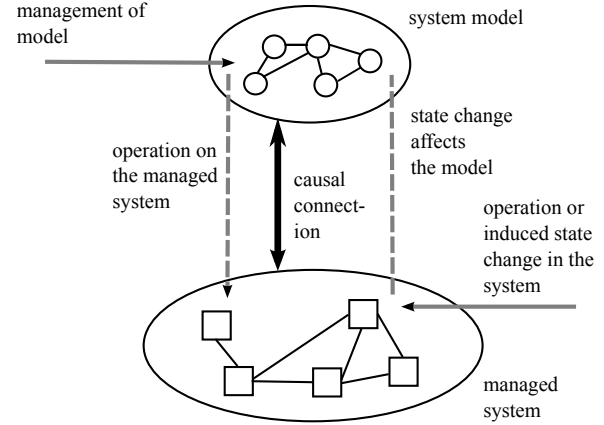


Figure 3. Overview of the reflective management by contracts

it governs. This metainformation can be transformed to simple monitoring rules that are executed by the underlying computing platform supporting a specific service. We might say that the reflective model is here utilised in the opposite direction: the running system is made to reflect the changing expectations towards it, and to trigger corrections to the model where the real system is not in synchrony with those expectations.

In respect of associations to ODP reference model, Pilarcos presents a federated system. The federation is formed by enterprise agents jointly governing the business network by sharing the eContract.

C. Member enterprise reflection

The final view to the complex system is from the enterprise point of view. In this paper just a few notes is needed, but fuller explanation is given elsewhere [18].

First, each enterprise stays independent, and can change its policies that direct it joining into collaborations any time based on its private strategies.

Second, the dynamic knowledge base supported jointly by the ecosystem is always available for each enterprise, so initiatives for business networks are easy to make or join. The utilities for interoperability and collaboration life-cycle management are responsibilities of the ecosystem infrastructure, so its quality, usability and trustworthiness must be separately considered. Although the service providers for infrastructure services can be rather varied, the market competition will soon pick out a few to this "trusted third party" role.

Third, the enterprises can detect breaches during the collaborations and cause sanctions or other breach recovery methods to be used without themselves supporting that kind of functionality.

In summary, the enterprises can utilise the ecosystem as a sensor for the changing market environment, and as an actor for feedback and evolution.

IV. DISCUSSION

The mission of the work on Pilarcos is to develop solutions for service interoperability and management of dynamically formed business service collaborations and peer communities.

According to our vision, in individual users, enterprises or public organizations can easily compose new services from open service markets, or establish temporary collaborations with complex peer relationships. Furthermore, these contract-governed collaborations can be managed by all involved parties. 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 organization; this infrastructure also takes responsibility of governing trust and privacy-preservation issues. The support environment is complemented with service-oriented software engineering practices that enable semantic and pragmatic interoperability management.

The commonly asked question is how Pilarcos relates to SOA, Web services, and virtual breeding environments, and how the concepts used in Pilarcos match with standards like IS15944 and IS10746.

To start with these approaches have very different goals and scopes:

- Pilarcos [14] develops facilities for open service ecosystems that support flexible business networks. The support has started from a breeding environment type of solution, supporting the establishment of business networks, but has continued to operational and dissolution time needs. Special focus has been on the evolution needs of the environment.
- RM-ODP [9], [10] is a reference model for distributed computing together with enterprise architecture framework for specification of ODP systems.
- ISO15944 [20] focuses on the alignment of business and operational views of open-edi to allow free selection of modeling techniques when designing business transactions. Requirements of importance include commercial and legal frameworks, and cultural adaptability.
- SOA roadmap [21] expects a reactive and adaptive computing environment for business-oriented use. It introduces the enterprise service bus (ESB), but leaves dynamically configurable runtime architecture as a grand challenge; it introduces service orchestration and service management, but leaves automated composition and self-reflection as future grand challenges.
- Web services [22] provide a standard means of inter-operating between different software applications, running on a variety of platforms and frameworks. Thus, the web services architecture is an interoperability architecture for generic use.
- ECOLEAD [3] has aimed to provide horizontal ICT

infrastructure, and virtual enterprise breeding environment for dynamic virtual enterprises in response to fast changing market conditions. This work has been based on theoretical foundations of collaborative networks.

Although Pilarcos has grown out from its original close correspondence with RM-ODP, it has still preserved many of the main concepts:

- The ODP communities and objects have grown to business networks and services; the ODP system interacting with its environment has been formalised by the ecosystem.
- Further, the ODP enterprise language specification with roles, interactions, policies and assignment rules is now denoted by the term business network models and have become more associated with business processes and choreography languages that are relevant concepts in the SOA, WS and ECOLEAD frames. In IS15944 the closest counterpart appears to be business object, i.e., reusable open-edi scenario.
- The eContract concept of Pilarcos has its counterparts in the ODP concepts of liaison and contractual context. While IS15944 has terms of commitment to apparently match with eContract, and business transactions that probably matches best with Pilarcos business networks the SOA, WS and ECOLEAD approaches do not focus on the contract as a separate agent.

All of the approaches provide additional infrastructure functionality. While IS15944 distributes scenarios and governs their reuse with regulatory systems, the other approaches focus on helping the client to find the services and bind to them. In web services that repository of service offers is called UDDI. The RM-ODP defines trader with rather similar functionality, but requires a type repository to be used to define structure and semantics for the offers. This solution allows protection of the consistency and trustworthiness of the offers. The same trend has been continued in Pilarcos, with the exception that the populator takes a whole business network model and populates all peer roles simultaneously, making multiway interoperability matching.

The life-cycle of a business network is in Pilarcos governed by a separate eContract agent and by local, subjective monitoring at each enterprise. The eContract life-cycle reflects the RM-ODP concepts of liaison, contractual context, establishing behaviour, and failure. In IS15944 a domain based regulator is used for monitoring the correct behaviour of business transactions. In most SOA or WS solutions monitoring is not needed separately as the business processes are enacted by distributed process engines.

Although reputation flows are only present in Pilarcos, some trust management related activities can be found in IS15944 and ECOLEAD. In IS15944 registry entries are first assessed by stewards before they are admitted; similarly in ECOLEAD each breeding environment member goes

though an elaborate and systematised trust assessment before accepted and profile registered.

In summary, Pilarcos approach is more focused on the self-reflective mechanisms that support business network life-cycle and the evolvability of the ecosystem. The ontology metamodels defined for structuring the infrastructure repositories and the relationships between items in different repositories are in key role in keeping the ecosystem safe and consistent. Pilarcos encapsulates enterprise-level choices on technology, policies, services and service engineering technologies, thus allowing heterogeneity on those aspects. Instead, new simple protocols have been introduced between enterprise-representing agents, eContract agents, and infrastructure services. These protocols create a management framework that is able to make commitments across enterprise boundaries; the information exchanged in this framework is sufficient for each participant to interpret requests, suggestions, commitments, failure complaints and policy statements.

REFERENCES

- [1] J. A. Zachman, "A framework for information system architecture," *IBM Systems Journal*, vol. 26, no. 3, pp. 276–292, 1987.
- [2] The Open Group, "Togaf 8.1.1 online," Tech. Rep., <http://www.opengroup.org/architecture/togaf8-doc/arch/>.
- [3] L. M. Camarinha-Matos and H. Afsarmanesh, "Virtual enterprise modeling and support infrastructures: Applying multi-agent system approaches," in *Multi-Agent Systems and Applications. LNCS2086*. Springer, 2001.
- [4] A. J. Berre, B. Elvester, N. Figay, C. Guglielmina, S. G. Johnsen, D. Karlsen, T. Knothe, and S. Lippe, "The athena interoperability framework," in *Enterprise Interoperability II*. Springer, 2007.
- [5] N. Mehendjiev and P. Grefen, *Dynamic Business Process Formation for Instant Virtual Enterprises*. Springer, 2010.
- [6] L. Kutvonen, J. Metso, and S. Ruohomaa, "From trading to eCommunity management: Responding to social and contractual challenges," *Information Systems Frontiers (ISF) - Special Issue on Enterprise Services Computing: Evolution and Challenges*, vol. 9, no. 2–3, pp. 181–194, Jul. 2007. [Online]. Available: <http://dx.doi.org/10.1007/s10796-007-9031-x>
- [7] P. Maes, "Computational reflection," Ph.D. dissertation, 1987.
- [8] G. Coulson, "What is reflective middleware?" *IEEE Distributed Systems Online*, 2001, http://www.imamu.edu.sa/dcontent/IT_Topics/java/rmarticle1.pdf.
- [9] *Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing. Part 2: Foundations*, ISO/IEC JTC1, 1996, iS10746-2.
- [10] *Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing. Part 3: Architecture*, ISO/IEC JTC1, 1996, iS10746-3.
- [11] L. Kutvonen and J. Metso, "Services, contracts, policies and communities – relationship to ODP framework," in *WODPEC 2005*.
- [12] L. Kutvonen, "What applying of the ODP viewpoints teaches us about tool-chains," in *10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*. Hong Kong: IEEE Computer Society, Oct. 2006, p. 35. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/EDOCW.2006.82>
- [13] E. Fehr and U. Fischbacher, "The nature of human altruism," *Nature*, October 2003.
- [14] 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*, ser. Advances in Enterprise Information Systems (AEIS). IGI Global, Dec. 2008, pp. 209–241. [Online]. Available: <http://www.igi-global.com/reference/details.asp?id=9648>
- [15] T. Ruokolainen and L. Kutvonen, "Service Typing in Collaborative Systems," in *Enterprise Interoperability: New Challenges and Approaches*, G. Doumeingts, J. M?ller, G. Morel, and B. Vallespir, Eds. Springer, Apr. 2007, pp. 343–354.
- [16] —, "Managing interoperability knowledge in open service ecosystems," in *Proceedings of the 13th Enterprise Distributed Object Computing Conference Workshops, EDOCW*. Auckland, New Zealand: IEEE, Sep. 2009, pp. 203–211. [Online]. Available: <http://dx.doi.org/10.1109/EDOCW.2009.5331993>
- [17] T. Ruokolainen, "Modelling framework for interoperability management in collaborative computing environments," Department of Computer Science, University of Helsinki, Tech. Rep. C-2009-9, Jun. 2009, licentiate's thesis. [Online]. Available: lic-thesis.pdf
- [18] S. Ruohomaa and L. Kutvonen, "Making multi-dimensional trust decisions on inter-enterprise collaborations," in *Proceedings of the Third International Conference on Availability, Security and Reliability (ARES 2008)*. Barcelona, Spain: IEEE Computer Society, Mar. 2008, pp. 873–880. [Online]. Available: <http://dx.doi.org/10.1109/ARES.2008.32>
- [19] *Complex adaptive system*, Wikipedia. Wikimedia commons, 2005.
- [20] *Information Technology – Business agreement semantic descriptive techniques*, ISO/IEC, 2002–2010, iS15944.
- [21] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: a research roadmap," *International Journal of Cooperative Information Systems*, vol. 17, no. 2, pp. 223–255, 2008.
- [22] *Web services architecture*, W3C, 2004.