Multi-tier agent architecture for open service ecosystems

Lea Kutvonen

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

Abstract. The present trend of enterprise computing is towards networked business; thus there is a high demand on a new layer of global infrastructure that facilitates inter-enterprise collaboration management and provides utilities for interoperability. We addresses the larger problem of collaboration management in cases where the collaboration is dynamically composed of business services from independent organisations. The Pilarcos architecture shows how generic collaboration management utilities can support, with a significant level of automation, the whole lifecycle of a collaboration from service selection with interoperability checking, negotiation with trust decisions and commitment to the operational phase, monitoring, and breach management. We analyse the multi-tier agent architecture nature of the open service ecosystems, and collaborative management of inter-enterprise collaborations within them.

1 Introduction

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 development has largely been enabled by the evolution of service-oriented computing (SOC) [1], business process management (BPM) and adaptation of Web services as a common family of technology. Interoperability issues have received attention as well from ontology researchers as from semantic web development. Even organisational differences have been addressed to detect and alleviate heterogeneity at process and conceptual levels. It is also marked with its application to service engineering (e.g. [2–5]). Electronic business collaborations have also received interest in the domains of multi-agent technologies (e.g., [6, 7]) and enterprise architectures (such as Zachman [8], TOGAF [9]), not to forget of enterprise interoperability domain (e.g., [10]) or virtual organisations environments (e.g., ECOLEAD [11–13] and CrossWork [14]). All of these aim to align business needs with the computing solutions available.

While common facilities for discovery of services and composition of them are emerging, a few essential challenges remain: i) preservation of partner autonomy of actors when establishing, controlling and dissolving collaborations, as well as

[*] AT2012, 15-16 October 2012, Dubrovnik, Croatia.

formulation of contracts between agents and balancing between human involved decision-making and automated support; ii) adherence to regulatory systems; and iii) evolution of the ecosystem and collaborations.

In our earlier work, we introduced *open service ecosystems* as such a collaboration governance environment [15–18]. Here we discuss ecosystem architecture from multi-agent system point of view: The basic actors include organisation-representing agents, and services that are organisation-governed and initiative-taking. Indeed, in the design of the ecosystem support and processes, agent technologies and speech act concepts, as well as deontic logic have played an essential role. Especially, focus is given on the responsibilities of agents, as there are some essential differences in comparison to related work that we find powerful. The key difference is the use of an explicit ecosystem entity that determines regulatory, methodological and ecosystem-specific disciplines that allow correctness-control in the dynamic inter-enterprise collaborations.

Section 2 provides an overview of the open service ecosystem: its stakeholders, management of inter-enterprise collaborations within it, and the management of services within each involved organisation. It also comments on the interrelated model-driven methods of developing collaboration types, i.e., templates for the eContracts (used for governing collaborations across organisational boundaries), and private and public ecosystem infrastructure services. Section 3 adds details on obligations the ecosystem and organisation-representing agents have, and the agent protocols at collaboration lifecycle, collaboration type design, ecosystem management, trust and privacy management, and decision-making at each organisation. Section 4 clarifies the contributions of the ecosystem design by providing some comparison to various elements of related frameworks.

2 Inter-enterprise collaborations in service ecosystems

In future, we vision, that individual users, enterprises or public organizations can easily compose new services from open service markets. Furthermore, these contract-governed collaborations can be managed by all 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 of governing trust and privacy-preservation issues.

The embedded interoperability support spans technical, semantic and pragmatic interoperability. We define interoperability, i.e. the capability to collaborate, as the effective capability to mutually communicate information in order to exchange proposals, requests, results, and commitments. 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 senders and receivers, both in terms of information representation and messaging sequences. Pragmatic interoperability captures the willingness of partners to perform the collaborative actions. 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.

Three key concepts in the Pilarcos open service ecosystems are those of inter-enterprise collaborations, eContract agents and ecosystem infrastructure. The Pilarcos architecture views *inter-enterprise collaboration* as a loosely-coupled, dynamic constellation of business services; it involves multiple partners through their software-based business services and their mutual interactions.

A *business service* is a software-supported service with a functionality suitable for a business need on the market and thus relevant for the networked business. In itself, each business service is an agent, in terms of being able to take initiative on some activity, being reactive to requests by other business services, and being governed by policies set by its owner. The relationship between business service and software supporting it resembles the relationship of an agent and web service [19]. Each of business services provide business protocol interfaces for each other, but also utilise locally provided agents for connecting to peer services through channels with appropriately configured properties (e.g., security, transactionality, nonrepudiation).

The type of the service constellations is declared as *business network model* (BNM), expressed in terms of the roles and interactions within the collaboration, the involved member services, and policies governing the joint behaviour [15]. Intuitively, a BNM describes a business scenario.

The *eContract agent* governs the inter-enterprise collaboration and captures both business and technical level aspects of control, as well the large-granule state information to govern the dynamism of the collaboration. The eContract is structured according to a selected BNM.

We define the *service ecosystem* as an environment – open service market – where service providers and clients can meet, establish contract-governed collaborations and gain experience on the services and partners involved. Our goal is to create a consistent but evolving environment to support governance of inter-enterprise collaborations, and to provide for consistency control of the inter-enterprise collaborations themselves [20].

An essential part of the ecosystem is its *ecosystem infrastructure*, a set of CaaS agents (Collaboration-as-a-Service) that provide shared utilities for enterprises to discover and select services available in the ecosystem, negotiate and establish collaborations, govern those collaborations through eContract agents, and utilise reputation information and collaboration type information.

As the ecosystems are required to intertwine engineering, governance and operational needs of collaborations, the Pilarcos ecosystem architecture involves:

- enterprises providing and needing each others business services, with their published business service portfolios [15];
- business-domain governing consortia, with their published models of business networks and business models [15];
- infrastructure service providers of individual functions such as service discovery and selection, contract negotiation and commitment to new collab-

orations, monitoring of contracted behaviour of partners, breach detection and recovery [15] and reputation flows from past collaborations [21];
- consortia and agencies that define legislative rules for acceptable contracts [18] and joint ontology about vocabulary to be used for contract negotiation, commitment and control [22, 23, 20, 18]; and
- infrastructure knowledge-base providers that maintain the information underlying the ecosystem infrastructure functions; this role is essential in enforcing all conformance rules of all ecosystem activities [23].

The main ecosystem activities are illustrated in Figure 1, involving service engineering (left and bottom), ecosystem and collaboration governance (left and right) and operational-time collaboration support (right and bottom). The left side of Figure 1 depicts processes related to engineering steps at each involved enterprise or consortia. Here, metainformation is brought to the system by designers and analyzers. First, available services are published by service providers (enterprises including public and private sector providers). Second, the publicly known BNMs are created by teams of designers and published after acceptability analysis. Third, regulations for conducting collaboration at administrative domains are fed in by enterprise and ecosystem administrators knowledgeable about local and international laws and business domain practices. This body of knowledge accumulates into metainformation repositories within the globally accessible infrastructure layer. The repositories only accept models that fulfil the set consistency criteria, thus providing a point of control.

The arrows leading to the right at the right side of Figure 1 depict the life-cycle of each collaboration from negotiation to termination phase. The collaboration establishment is initiated by one of the partners suggesting the use of commonly known BNM that can be picked from the infrastructure repositories. Further, the infrastructure services help in discovery and selection of suitable partner services for the collaboration and running a negotiation protocol between the selected partners. Within the negotiation step, the local, private support agents of each partner consider especially the suitability of the collaboration for the enterprises strategies and sufficiency of trust to the other partners in the collaboration. In the enactment and control phase, the local support agents provide protective monitoring and the required contract-related communication.

The arrows leading to the left at the right side of Figure 1 depicts the experience information gathered from all the collaborations in the ecosystem and providing feedback information for re-engineering and future decision-making processes in the ecosystem. Especially important is the reputation information generation, as the reputation-based trust management concept facilitates the scalability of the ecosystem. Here we can rely on social ecosystem studies [24]: The number of potential partners in the ecosystem is very limited if there are no established behaviour 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 and the potential risk

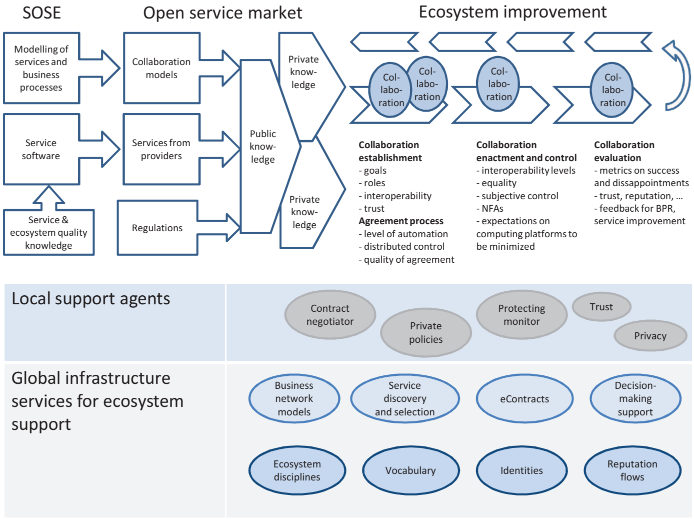

Fig. 1. An overview of the Pilarcos open service ecosystem.

predicted with reputation information, creates a cycle that has this necessary control function. It effectively emulates the social or legal system pressure of business domain. This functionality is much missing from other approaches.

In conclusion, the bottom part of Figure 1 represents the global, federated infrastructure services that participates the governance, engineering and collaboration management processes. The ecosystem infrastructure services provide generic protocols and knowledge-bases for enterprises to

- match the collaboration management needs; including service discovery and selection, eContracting, and breach management [15];
- evolve the ecosystem with processes to keep and enhance a coherent knowledge base; this includes the control of collaboration types and available services and interoperability management information [22, 23];
- regulate 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 [23];
- perform private decision-making on collaborations; this includes enterprises' expert system for making decisions related to contracts, breaches and trust [21];
- perform (globally) distributed service production; this involves production methods for service software and coordination models, and definition of open service ecosystem quality requirements for software; and
- collect feedback about the reputation of the (successfully or prematurely) terminating collaborations and their component services [21, 25].

From the business point of view, the ecosystem provides a) a global infrastructure for collaborations that utilize services provided by other ecosystem members; b) a natural environment for innovating new services and new collaboration types; and c) helps enterprises in adjusting to rapidly changing business situations and participation in natural competition between collaborations and ecosystem members.

3 Multi-tier agent system

The ecosystem-involved agents form four agent tiers to be discussed separately: i) the inter-enterprise collaboration that is governed by an eContract agent and in which enterprise-representing agents are involved in; ii) the collaboration-as-a-service (CaaS) community, formed by the ecosystem infrastructure services, including the populator agent, service type repository, BNM repository, and reputation-providing agents; there are simultaneously several CaaS communities from which enterprises can choose from; iii) the BNM defining community; and iv) the open service ecosystem that is grounded to a CaaS community but in addition has regulatory and policy-based restrictions, so it can direct the acceptability rules of BNMs and service offers in its domain, for example.

These communities are intertwined: CaaS agents and enterprise-representing agents appear in more than one of the above mentioned communities. Below, each of these communities will be discussed in further detail.

3.1 CaaS tier

The ecosystem infrastructure provides generic service agents for ecosystem and collaboration management support. These agents are used by enterprises' private agents during the collaboration lifecycle from establishment, through operation to dissolution. Also present as agents are those knowledge-bases that are essential for the consistency of the ecosystem behaviour and correctness of collaborations.

In the collaboration establishment phase, service discovery and selection is supported by a *populator agent*. The collaboration initiator selects a model from the *public BNM repository (agent)* and invokes the populator to find matching service offers from the trading service for each of the roles [15, 22]. The populator performs a static interoperability check to ensure that the service offers fit the collaboration model, and are compatible with other offers proposed into the same collaboration. New proposed service offer sets can be picked within given resource limits. The populator returns a contract proposal that ensures that the set of services it proposes do 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.

In comparison with other service offer repositories (UDDI [26], ODP/OMG trader [27]) the fundamental difference is the populator service providing a multi-partner matching instead of a client-server setup, and also checks not only technical and semantic interoperability but also takes into account pragmatic interoperability aspects. The pragmatic aspects include views to BNMs, acceptable role combinations and environment contract information (i.e., requirements of the communication channel properties). The information base utilized by the populator agent is based on *ODP trading service*.

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 public agent [15]. Automated negotiation supports the agreement phase of the collaboration [15] (see Collaboration tier) and leads to the formation of the specialised *eContract agent* for the collaboration instance. 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. The commitment concept in place at the eContract establishment time follow the ontology for commitments in multi-agent systems [28]: discharging, assigning, delegating, and cancelling. It also supports a new model of business transactions [25].

In the policies embedded in the eContracts we consider deontic logic [29] rules to be appropriate; this is in line with the usage of ODP RM (open distributed processing reference model) [30] concepts and viewpoints as part of the formalisation of our architecture. Deontic logic is not binary (denied/compulsory), but uses rules of prohibition, obligation and permission instead. This is necessary in an environment where there is no single policy maker or enforcer of the policies but the actors are independent of each other. Thus it is not possible for force a partner to refrain from an action, or to force that partner to take another action. However, it is possible to agree that it is a violation of a prohibition to take certain actions, and in addition, to agree on the consequences of violations. The detailed behaviour on functional or nonfunctional aspects of the partners cannot either be (practically) agreed on, but some optional behaviour patterns can be allowed without causing violation management. This area is where permissions clarify the behaviour: something is optional to take place, and there is a specification in existence about the followup behaviour.

This policy approach allows us to make clear distinction between violations of the contracts and acceptable behaviour according to that contract [31]. However, each partner in the collaboration uses subjective rules for decision-making on whether to join the collaboration, or on whether to report to the eContract agent some violation detected in the sequence of actions they get exposed to.

The eContract provides interfaces for the collaboration partners for renegotiation, epoch changes (where membership or responsibilities can be changed), progressing to defined milestones in the business processes, and declaring detected breaches. The eContract is the key agent also at the collaboration termination phase when feedback on the success of the collaboration is collected for business process improvement, service improvement, ecosystem improvement and for partners' service reputation information feeds. Part of the information is relevant to be produced from the shared state information, while most of the data is best produced by the local monitors at each enterprise. The eContract forwards this information to appropriate repository agents.

Essential agents in the CaaS tier are also the reputation flow agents. For each successful or unsuccessful collaboration termination, reports will be fed for them. These agents aggregate reputation information on several asset aspects including monetary, reputation, and control assets. These in turn are available for private decision-making agents at the eContract negotiations in future. Therefore, a dynamic incentive mechanism is effectively created for ecosystem members to keep to their service offers and eContract commitments (including privacy rules), and especially to the reporting protocols [21, 32].

3.2 Inter-enterprise collaboration tier

The inter-enterprise collaboration tier is participated by the private agents of each involved enterprise, the eContract agent, and the reputation flow agent(s) of the ecosystem. The *local support agents* subjectively represent the enterprise, and provide a local interface to the ecosystem infrastructure services for the local business services. The essential tasks at which enterprises need their agents to control the collaboration contract or collaboration behaviour include i) contract negotiation, ii) monitoring during collaboration operation, and iii) experience reporting when the collaboration terminates either having reached its purpose or terminating prematurely due to breaches.

A *contract negotiator* agent represents an enterprise and is responsible of running collaboration management protocols on behalf of the enterprise delegating rights to it. The agent provides interfaces for application software or administrative interfaces to initiate collaboration establishment, or for responding to suggestions from other enterprises. While the initial service selection is based on public information, the needs for privacy of decision-making on the enterprises' commitments is incorporated to a negotiation phase. In the negotiation phase each suggested collaborating party can agree to join the collaboration, or refrain. In routine cases, it is possible for the enterprises' agent provide an automated response to the collaboration proposals: an explicit meta-policy guides the agent to pick routine rejections or commitments. Other situations can be recognised, for example, by uncertainty of the trustworthiness of the peers, uncertainty of the strategical benefit of the collaboration, or uncertainty of the acceptability of negative reputation effects caused by a refusal.

The contract negotiator is aware of enterprise policies that govern all negotiations and commitments, and all services. Where the contract negotiator is not able to make a decision (metapolicies deny the right from it) on whether to accept or reject a proposal, it passes the proposal on for human intervention - and the decision-making support system information is made an expert support system style [33, 21]. The decision-making is governed by enterprise policies [33, 21, 20] related to a) strategic policies indicating what type of collaborations or which partners are of interest and worth investing the resources to collaborate with; b) reputation-based trust that weights the anticipated risk and tolerated risk level [21]; and c) privacy-preservation that may overrule otherwise acceptable collaborations due too high privacy costs involved.

Although trust and privacy are closely related, the decision-making processes on the issues are separate and parallel. Trust decisions weight expected benefits against anticipated losses in a specific business case; privacy decisions guard access to private information, metainformation and behaviour patterns.

We define trust as the extent to which one party is willing to participate in a given action with a given partner in a given situation, considering the risks and incentives involved. Trust decisions are subjective evaluations made by the trustor, targeting a given trustee and a given action in terms of standard assets shared between organizations: monetary, reputation, control and satisfaction [21]. A trust decision is based on a comparison of the uncontrollable risks that allowing the action would cause, and the willingness to accept them, i.e., risk tolerance. The risk evaluation is expressed as probabilities of different outcomes, estimating how the partner will behave in the future. This estimate is based on earlier experience with the trustee. First-hand experiences and experiences shared by other ecosystem members form the trustee's reputation, which is the trustor's subjective perception of how trustworthy the trustee is. Risk tolerance builds on the business importance of allowing the action: different kinds of benefits may be realized by a positive decision alone, such as building a partnership, helping the inter-enterprise collaboration towards realizing its goals, and not triggering compensation clauses in the contract.

We define privacy as the right of subjects to determine themselves for whom, for what purpose, to what extent, and how information about them, or information held by them, is communicated to others [32]. Here, the subject can be a person, social group organisation or organisational group. Privacy control is the set of actions by which a subject makes decisions on refraining or involving in information exchange or sharing, and taking actions on detected privacy violations. A privacy violation is circumstances where information is held or used in a way that breaches the privacy declaration by the information owning subject. Privacy declaration is an expression created by the protected information owner that gathers together rules on to whom, for what purpose, to what extent and how information can be made available. The negotiation phase allows each partner to reject or agree the collaboration without exposing their private policies about the decision. Reasons for rejecting can involve type of collaboration, partner identity, partner reputation, or strategy on committing to the collaboration load. The privacy enforcement during collaborations is mainly performed by trusted infrastructure, because in many scenarios, the actual enforcement of the policies is performed in systems that are not under the direct control of the owners of the processed private information. The local monitor agents (see below) can intercept (and stop) both i) incoming requests that violate the privacy declarations at the contract level or cause a discrepancy against the receivers local policies. and ii) outgoing information exchangers that violate or are at risk of indirectly compromising its local privacy declarations or those of the collaboration.

Monitoring agents support the enactment and control phase of the collaboration [34] by checking the acceptability of the behaviour (messaging) it can directly assess. The monitors receive rules from eContract and from their local policy repositories. These rules 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. The contradictions can mean failing to fulfil an obligation, or failing to provide the agreed quality level of the service, such as availability, timeliness, and privacy-preservation, or as non-repudiation and immutability. At detected breach situations, the partner needs to decide (automatically or through human intervention) whether the breach is serious enough for terminating or leaving the collaboration. When an essential breach is detected, the eContract agent is notified. The eContract agent then triggers recovery steps, for example, terminating the collaboration, or changing the faulty member to a new one. The recovery capabilities are dependent on the BNM, and therefore, the breach recovery process is defined as part of the BNM.

When terminating the collaboration, *experience reporting* is required [21]. The experience reporting forms the core of social control in the open service ecosystem. As contract violations are detected by monitors, they become known to other actors as well. This creates a direct reputation impact that limits 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 [21]. False reports do not only affect the targeted service, but also inhibit the other actors' ability to assess its behaviour, reducing the social control impact of reputation. The experience reporting from all parties provides a reference point for ecosystem level membership that relies on a democratic measure.

The *eContract agent* comprises of the collaboration metamodel and operations changing it; thus, the eContract provides a shared-language view on the collaboration structure, behaviour, policies and abstracted state. The logical eContract agent that is physically replicated to the computing systems of each collaboration member. The private contract agents are responsible of keeping the local services in their governance in synchrony with the committed eContract status. Between themselves these local contract agents need to use protocols familiar from multi-agent systems or speech act theories: definitions and declarations, requests, suggestions, commitments, and opinions.

3.3 Modeling tier

The purpose of the modeling tier is to create BNM models and service types into the ecosystem. We also group distributed, service-oriented, model-driven software engineering processes into this domain as they produce metainformation to the custody of the CaaS repository agents. The Pilarcos architecture includes four essential metainformation repositories: service offer repository, service type repository, BNM repository, and reputation information flow. The repositories must control the publication of offers or models strictly, following the rules provided by the ecosystem management tier.

We have separated the BNM design phase from the collaboration establishment phase, to further automate the commitment phase. The traditional virtual organisation breeding environment way (e.g., in ECOLEAD and CrossWork [35]) of first choosing the partners and base the business processes on their capabilities actually forces the design phase for each individual collaboration. The business network models can be separately but collaboratively designed, verified and validated for their suitability for the market domain. These models also provide a common vocabulary for enterprises to use at the business network establishment negotiations: When a collaboration is being established, the pragmatic interoperability (processes and policies) is tested between partners. Thus, the business services forming a collaboration do not necessarily have a joint history in the breeding environment that would enforce interoperability but are just introduced to each other in a refining negotiation of the eContract.

The BNMs must be verified and validated carefully before being accepted to the repository. The engineering methodology used must declare the authority for submitting the model to a business network model repository on behalf of the designing team; we assume a team here, because the BNM design requires expertise from multiple domains, such as business best practices, associated regulation systems, enterprise and process modelling, market situation and room for new process innovations, and access to the feedback from past collaborations and their experience. From the technical point of view, BNMs are compositions of business processes. Thus many of the existing business process or protocol verification tools can check for well-formedness, aliveness, deadlock freeness and other process properties. In addition to this control flow point of view, also the information flows must be considered carefully to detect privacy-threatening exchange of information, excess exchange of information causing bad performance and privacy-threatening cumulation of information to roles.

Service type definitions form a basic vocabulary for declaring business network models and publishing service offers. The type definitions can be reused in multiple collaborations, too, thus creating opportunities for business services to be used in cross-domain business networks.

The metainformation repositories form an ontology forest where the BNMs form roots. The service types created can be inserted to the name space of the first BNM they are used, to differentiate them from other similarly named types in other application domains.

3.4 Open service ecosystem tier

The open service ecosystem management tier is responsible of capturing the consistency, acceptability and regulatory aspects into the ecosystem knowledge-bases. These models and rules regulate the behaviour of the generic CaaS agents. Furthermore, the knowledge-base is not static but evolvable, thus making the evolution of the ecosystem and the individual collaborations within it evolvable.

For consistency enforcement, the ecosystem repositories are governed by four ontology metamodels [23, 20]: i) domain ontology metamodel that defines basic ecosystem concepts like collaboration, service, and contract; ii) methodology metamodel that defines phases of service engineering processes and especially the produced artefacts; iii)domain reference metamodel that captures the infrastructure elements by defining the operational-time support functions and artefacts manipulated; iv) 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 an operational-time infrastructure function, and has a technical storage representation. Thus, the design and production time artefacts become also artefacts at the operational time. The metamodels are also extendable at each abstraction layer: it is possible to add new top level concepts and relate them to the existing ontology.

As an example of the metamodel effects on the collaborations, we can consider the route by which regulations are embedded to the monitors of a collaboration: At the design time of a BNM regulations are used to validate the BNM before acceptance, and suitable policies can be embedded into the BNM itself to allow optional behaviours or adaptability to changes in the regulations restricting the use of the optional behaviours. At the collaboration negotiation time, this BNM is used as the basis of the eContract and thus the regulatory system becomes inherited. The partners of the collaboration each use both the eContract policies and their local enterprise policies to govern their local behaviour. Eventually, the regulatory rules get monitored by each of the partners locally and thus can trigger breach management and trust consequences, for example.

4 Discussion

The contributions of the open service ecosystem architecture include i) the CaaS tier; ii) evolution and regulatory system support to collaborations through the ecosystem tier; iii) the collaboration governance through the eContract agent; iv) private agents to provide subjective, adjustable monitoring of breaches; and b) incentives and breach management processes trough the ecosystem tier.

Our *research methods* include i) prototyping of the CaaS tier services [15, 34, 21, 23]; ii) creating and analysing Coloured Petri Net model of the collaboration lifecycles [25]; iii) providing a hierarchy of metamodels to ensure the design-time and run-time artefacts meet the needs of both phases properly and that non-functional property framework with trust, privacy and business-enhanced service level agreement needs can be systematically be satisfied [17, 23, 16]; iv) specified key parts of the system using ODP RM [18] and v) simulations and performance measures of the trust management facilities and CaaS service prototypes.

In related work, concepts of business ecosystems and digital ecosystems appear more frequently. However, there are some goal-level difference often present. The term service ecosystem is often used to refer to an environment where a platform exists for applications to be available in a shared manner from multiple organisations or individuals for maintaining their information flow needs. The difference of open service ecosystems to these is the introduction of CaaS and ecosystem tiers that allow changing the platform and the model and rule bases for governing the acceptability and correctness criteria in the ecosystem.

For digital ecosystems (e.g. Digital Business Ecosystems project 507953, [36]) the purpose of ecosystem introduction is similar; agents represent software-based services, and information exchange patterns join them together. Habitats in digital ecosystems seem to represent similar phenomenon than business networks in our work. In digital ecosystems, an essential goal is to help agents migrate to different habitats and my evolution also increase their fitness. Fitness is measured in terms of suitability to sufficient amount of requests made in the ecosystem. In our work, services belong to a service portfolio of an enterprise that is member in an ecosystem - the services become available for other ecosystem members as SaaS services. In terms of fitness measures, populators do match service offers to collaboration requests based on the suitability of service type and in the negotiations, each potential partner decide on the trustworthiness of that composed business network. However, individual services are not evolved within the collaboration lifecycle but behind the modeling and production tier, and thus appear as new, improved services while non-requested services just stay passive and can eventually be administratively removed. Also fitness in mind, monitors at each enterprise collect constantly information about the usefulness of the services utilised, thus enabling re-engineering feedback.

The ALIVE project [37, 38](FP7-215890) service governance is addressed through a model for dynamic SOA using agent-based technology. The model captures different levels of abstraction for the specification of governance, expectations, and behaviour of the services, not just their functionality, so that their run-time interactions are predictable, observable, and controllable. The contracts between partner services are dynamic and autonomously configurable, the participants will need the ability to negotiate at run-time to establish the SLAs, to monitor compliance with them, and to take actions to maintain them. While the work addresses trust issues and quality of service level agreements, the presented classification of ecosystems [38] does not include environments where the surrounding ecosystem would have control principles or functionalities for disciplining the member behaviours. However, the ALIVE solution proposes a set of tools for organisations to create choreographies between components of agentified services across multiple organisations, as well as tools for monitoring and controlling these compositions. Thus also this toolset helps humans to manage each independent collaboration through models.

Work from CONSOLIDER project on open system communication [39] provides an interesting counterpart for our work. We can relate collaborations with electronic institutions, BNMs with scene networks, and the details of our application-level business protocols between business services with utterance patterns. Interestingly, the Z language is utilised for expressing information state changes in transitions; the ODP RM information viewpoint semantics is defined in Z, and thus the formal foundations in joining utterances (computational viewpoint) and information state changes (information viewpoint) together seem similar in nature.

Our future work plans include projects on the processes of organisations to interface with the described tiers of eContracts, ecosystems and local policy-governing agents, especially in terms of trust and privacy decisions.

References

1. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: a research roadmap. IJCIS **17**(2) (2008) 223–255
2. Stojanovic, Z., Dahanayake, A., eds.: Service-Oriented Software System Engineering: Challenges and Practices. Idea Group Publishing (2005)
3. Tsai, W.T., Wei, X., Paul, R., Chung, J.Y., Huang, Q., Chen, Y.: Service-oriented system engineering (SOSE) and its applications to embedded system development. Service Oriented Computing and Applications **1**(1) (April 2007) 3–17
4. Margaria, T., Steffen, B.: Service engineering: Linking business and it. Computer **39**(10) (2006) 45–55
5. Papazoglou, M.P., van den Heuvel, W.J.: Business process development life cycle methodology. Commun. ACM **50**(10) (2007) 79–85
6. Wooldridge, M.: An Introduction to Multiagent Systems. Wiley (2002)
7. Winikoff, M., Desai, N., Liu, A.: Principles and practice of multi-agent systems. Multiagent and Grid Systems **8**(2) (2012) 125–126
8. Zachman, J.A.: A framework for information system architecture. IBM Systems Journal **26**(3) (1987) 276–292
9. The Open Group Architecture Forum: TOGAF 9: The Open Group Architecture Framework. The Open Group (October 2008)
10. : Future Internet Enterprise Systems FInES Research Roadmap 2025. (March 2012) Draft version 2.0.
11. Camarinha-Matos, L.M., Afsarmanesh, H.: Virtual enterprise modeling and support infrastructures: Applying multi-agent system approaches. In: Multi-Agent Systems and Applications. LNCS2086, Springer (2001)
12. Rabelo, R.J., Gusmeroli, S., Arana, C., Nagellen, T.: The ECOLEAD ICT Infrastructure for Collaborative Networked Organizations. In: Network-Centric Collaboration and Supporting Frameworks. Volume 224., Springer (2006) 451–460
13. Msanjila, S., Afsarmanesh, H.: HICI: An approach for identifying trust elements in the cse of technological trust perspective in VBEs. In: 2nd International Conference on Availability, Reliability and Security. (2007) 757–764
14. Mehandjiev, N., Grefen, P., eds.: Dynamic Business Process Formation for Instant Virtual Enterprises. Springer (2010)
15. Kutvonen, L., Metso, J., Ruohomaa, S.: From trading to eCommunity management: Responding to social and contractual challenges. Information Systems Frontiers **9**(2–3) (July 2007) 181–194
16. Ruokolainen, T., Kutvonen, L.: 21: Framework for managing features of open service ecosystems. In: Handbook of Research on Non-Functional Properties for Service-Oriented Systems: Future Directions. IGI Global (December 2011)
17. Ruokolainen, T., Ruohomaa, S., Kutvonen, L.: Solving service ecosystem governance. In: EDOC2011 Workshops, IEEE Computer Society (August 2011) 18–25
18. Kutvonen, L.: ODP RM reflections on open service ecosystems. Computer Standards & Interfaces (February 2012) In press.
19. Payne, T.: Web services from an agent perspective. Intelligent Systems, IEEE **23**(2) (march-april 2008) 12 –14

20. Ruokolainen, T., Kutvonen, L.: An architecture framework for facilitating sustainability of open service ecosystems. In: EDOC2012 workshops. (2012)
21. Ruohomaa, S.: The effect of reputation on trust decisions in inter-enterprise collaborations. PhD thesis, Univ of Helsinki, Dept of Computer Science (2012)
22. Ruokolainen, T., Kutvonen, L.: Service Typing in Collaborative Systems. In Doumeingts, G., Mueller, J., Morel, G., Vallespir, B., eds.: Enterprise Interoperability: New Challenges and Approaches, Springer (April 2007) 343–354
23. Ruokolainen, T.: Modelling framework for interoperability management in collaborative computing environments. Licentiate thesis, University of Helsinki, Department of Computer Science (2009)
24. Fehr, E., Fischbacher, U.: The nature of human altruism. Nature (October 2003)
25. Kutvonen, L., Norta, A., Ruohomaa, S.: Inter-enterprise business transaction management in open service ecosystems. In: EDOC2012. (September 2012)
26. Bellwood, T., Clment, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y.L., Januszewski, K., Lee, S., McKee, B., Munter, J., von Riegen, C.: UDDI Version 3.0. UDDI Spec Technical Committee Specification, 19 July 2002
27. ISO/IEC: "IS13235-1: Information technology - Open distributed processing – Trading function: Specification". (1998)
28. Singh, M., Huhns, M.: Service-Oriented Computing: Semantic, Processes, Agents. John Wiley & Sons,Ltd., West Sussex, England (2005)
29. von Wright, G.H.: Deontic logic. Mind **60**(237) (1951) pp. 1–15
30. ISO/IEC: IS 10746-3 Information Technology — Open Distributed Processing Reference Model: Architecture, also ITU-T Recommendation X.903. (2010)
31. Ruohomaa, S., Kaur, P., Kutvonen, L.: From subjective reputation to verifiable experiences - augmenting peer-control mechanisms for open service ecosystems. In: Trust Management VI. Volume 374 of IFIP Advances in Information and Communication Technology., Surat, India (May 2012) 142–157
32. Shen, Y., Miettinen, M., Moen, P., Kutvonen, L.: Privacy preservation approach in service ecosystems. In: EDOC2011 Workshops. (2011) 283–292
33. Kutvonen, L., Ruohomaa, S., Metso, J.: Automating decisions for inter-enterprise collaboration management. In: Pervasive Collaborative Networks. IFIP Working Conference on Virtual Enterprises. Number LNCS283 (2008) 127–134
34. Kutvonen, L., Ruokolainen, T., Metso, J.: Interoperability middleware for federated business services in web-Pilarcos. International Journal of Enterprise Information Systems, Special issue on Interoperability of Enterprise Systems and Applications **3**(1) (January 2007) 1–21
35. Mehandjiev, N., Afsarmanesh, H., Camarinha-Matos, L., Kutvonen, L., Norta, A.: Comparable Approaches to IVE. Advanced Information and Knowledge Processing. In: Dynamic Business Process Formation for Instant Virtual Enterprises. Springer (2010)
36. Briscoe, G., Sadedin, S., De Wilde, P.: Digital ecosystems: Ecosystem-oriented architectures. Natural Computing **10** (2011) 1143–1194
37. Dignum, F., Dignum, V., Padget, J., Vázquez-Salceda, J.: Organizing web services to develop dynamic, flexible, distributed systems. In: 11th Information Integration and Web-based Applications and Services. iiWAS '09 (2009) 225–234
38. Brazier, F.M.T., Dignum, V., Huhns, M.N., Derksen, C., Dignum, F., Lessner, T., Padget, J.A., Quillinan, T.B., Singh, M.P.: Agent-based organisational governance of services. Multiagent and Grid Systems **8**(1) (2012) 3–18
39. dInverno, M., Luck, M., Noriega, P., Rodriguez-Aguilar, J.A., Sierra, C.: Communicating open systems. Artificial Intelligence **186**(0) (2012) 38 – 94