Enhancing the Maturity of Open Service Ecosystems and Inter-enterprise Collaborations

Lea Kutvonen

Enhancing the Maturity of Open Service Ecosystems and Inter-Enterprise Collaborations

Lea Kutvonen

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

Abstract. The present business era is labeled by collaborations across enterprise boundaries and by utilisation of service-based computing. Pervasive computing utilities are created to match the basic business activities, such as contracting and breach management, adaptation of innovative business models, and collaboration management. Categories of computer assisted breeding environments and automated service collaboration management ecosystems have been developed to address these needs. However, a maturity framework is required for comparing solutions and indicating gaps in systems development and standardisation, and for adoption of a sufficient set of multidisciplinary research and evaluation methodologies. This paper first introduces steps towards a maturity model, focusing on features that contribute to the correctness of collaborations and scalability of the ecosystem. Second, it introduces the choices made in Pilarcos ecosystem. Finally, it discusses the need for standards and maturity models on this domain, and raises issues on the research methodologies required.

1 Introduction

The present business era is labeled by collaborations across enterprise boundaries and by utilisation of service-based computing. Computing utilities are created to match the basic business activities, such as contracting and breach management, adaptation of innovative business models, and collaboration management.

These needs are addressed by trends of i) systems where breeding of collaborations across enterprise boundaries is facilitated by glocal applications (glocal= global + local aspects meet to make a pervasive environment) and ii) systems where business services are automatically composed using collateral business processes (choreographies) across organisational boundaries.

Roughly, we can consider the first one to be focused on enterprise interoperability. Enterprise interoperability solutions are likely to be run by human decision-makers, because the aim is to address unexpected, new business opportunities that require very close planning and implementation phases to become profitable. The latter focuses on service interoperability, expecting enterprise and business concerns to be used as governance policies, rules and decision-making input. Service interoperability solutions are likely to be allowed to run automatically, addressing new, but expected business cases for which a sufficient amount of software modules are available for runtime composition in a self-administrative manner. Essentially, the technical computing and engineering solutions are very similar, but the expected users of these technical facilities differ; this caricature indicates how the themes of enterprise interoperability, service interoperability and service ecosystems complement each other.

A third pattern to observe is the emergence of ecosystems, such as i) software ecosystem by Amazon, Nokia or Apple, ii) eBusiness networks as in supply chains, and iii) social networking platforms, like Facebook or LinkedIn. Each of these bring in elements of discovering new partners for collaboration with explicit or implicit behaviour patterns, business models of explicitly agreed nature and roles of involved partners, and capability of easy evolution. However, each of them addresses only one side of the expected mature ecosystem concerns.

For a mature business service ecosystem we expect i) overcoming innovation boundaries [1]; ii) explicit contracting on business and technology level while preserving partner autonomy in the ecosystem and in collaborations [2]; iii) trust management system to support private decision-making while allowing introduction of new partners into the ecosystem [3]; iv) breach detection and management in an automated, business situation sensitive way (for which the present business transaction techniques are not suitable [4]).

In the CINCO group, we have developed an open business-service ecosystem [2, 5, 6, 1, 3] architecture and supporting ecosystem infrastructure services [7], and furthermore, focused on the essential viewpoints and lifecycles [1] that generate correctness criteria for collaborations [6]. This is to address the key problems in inter-enterprise computing today: i) ad-hoc engineering and integration, either directly or through engineering tools that do not have sufficient scientific basis; ii) insecure and misplaced decision-making, e.g., engineers implementing fixed strategies affecting business model or user experience, and iii) missing control and governance of the composed collaboration.

This paper first introduces a comparison framework as a step towards a maturity model, focusing on features that contribute to the correctness of collaborations and to the scalability of the ecosystem. Second, we outline the choices made in the Pilarcos ecosystem infrastructure as an example. Thirdly, we discuss the need for standards and maturity models on this domain, and finally raise issues on the research methodologies required.

2 Ecosystem comparison framework

For the purposes of comparison we assume the concepts of (business) service, business process, collaboration, and interoperability to be present and that there is vocabulary for declaring their more detailed properties. In addition to these, as ecosystems have different focal areas, we split the comparison framework into three sections: i) innovation and engineering, ii) collaboration lifecycle and iii) ecosystem infrastructure concepts and service. Further, we must note how the ecosystem key concepts are connected across these viewpoints in each case.

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.

This differs from the standard definitions deliberately by bringing in terms that are important in business terms (like contracts and negotiations), and enforcing concepts from speech act theories to be utilised, due to their suitability for expressing business needs and their technical support.

Due to parallel work, the definition also deviates from the term conceptual interoperability that is split into integrability (technical and syntactic), interoperability (semantic, pragmatic) and composability (dynamic, conceptual). Our definition captures the same levels but places composability as a goal of pragmatic interoperability.

The comparison framework will include the questions about the support for conceptual, dynamic, pragmatic and semantic interoperability.

Innovation and engineering: The traditional software engineering process produces monolithic artefacts that are built with the concepts supported by the engineering tools and the computing platform on which the artefacts are to be run. The process is based on knowledge on computer science and software engineering science, but omits key concepts from other scientific areas; there is little support for solving business issues, for addressing user experience alternatives, and crafting software module composability and management of compositions. Often, the hardest problems are on areas where the engineering phase is not the right time for solving the problem, but should allow operational time decision-making, because the decisions can depend on the presence of suitable partners, control of nonfunctional properties such as trust and privacy or transactionality, or regulations forced on the ecosystem to govern all its collaborations. Furthermore, the ecosystem evolution should not be considered only as a software versioning problem, just because the traditional engineering processes are not capable of handling other aspects.

Service-oriented software engineering (SOSE) [8] enhances the perspective by enriching the engineering process with lessons learned in service sciences in terms of requirements, and SOC platforms [9] and development tools in terms of development environment needs. The environment needs to be aware of the memberships, regulation systems and pervasive infrastructure services for runtime compositions. These facilities allow services supported by software artefacts to be composed together to a manageable entity that is aware of its business context and its users' situational preferences.

Special business-level challenges to address during the shared innovation and design phases include i) development of collaborative business models for independent partners; ii) partitioning of cost, risk and gained assets in the collaboration contract pattern; iii) trust between partners on being impartial at the design phase; and iv) management of collaborations being made possible by different collaboration and ecosystem members as their roles in the ecosystem requires.

The innovation phase creates declarations of business processes and collaboration models for the collaboration lifecycle support processes to utilise. Thus this is one of the collaboration correctness criteria sources, which furthermore is preferably to be considered impartial of ecosystem member incentives.

Collaboration lifecycle: The collaboration lifecycle includes traditional phases of i) establishment, ii) operation (or enactment) and control, and iii) dissolution, but also furthermore, iv) collection of experience information for the improvement of further ecosystem activities. Activities in these phases can be mapped to business terms like tenders, proposals, commitments, breaches, and opinions. The collaboration contract is an essential concept for making all the correctness criteria cumulated into the contract from the ecosystem, and collaboration partners. The classifying questions are captured in Table 1.

The essential differences in system architectures according to our surveys include splitting to i) enterprise interoperability or service interoperability sys-

All phases
Is the process a human process with computing support, or automated with human interventions supported? Is the contract involved a multiparty contract or client-server-based? Do the processes always allow partners to make subjective decisions, or is there a centralised decision-making point? Is the decision-making logic binary or deontic? Is the contract dynamic? Does it involve business or technology details or both?
Collaboration establishment
Nature of information involved: i) partners and their roles in the collaboration pattern; Nature of processes involved: i) Decision-making on trust for the suggested partners or services; ii) interoperability checking; iii) agreement process: level of automation, distribution of the control, quality of the resulting agreement.
Enactment and control of a collaboration
Levels of interoperability considered; Equality of partners in enactment or centralisation of orchestration control; support for subjective monitoring of processes and NFPs Whether expectations on the communication platform are implicit, explicitly stated, or requirements by which an open binding can be constructed at operational time.
Collaboration dissolution
Can be triggered by any partner at completion of the task or notification of a breach?
Experience collection
Metrics for successes and failures; Generation of experience information for reputation systems; Feedback generation for BPR and service improvement

Table 1. Classification questions for collaboration lifecycle.

tems; ii) dynamism of the collaboration contract and the availability of control interfaces at the enactment phase, iii) multiparty vs client-server constellations, and iv) methods for keeping the membership of the ecosystem in control.

Collection of partner information
Process of collecting: i) How is the required information produced and published? Granularity of services? Notation suitable (conceptual coverage), extendable, efficient? ii) Does it cover processes, collaboration models, service behaviour /interfaces, NFPs? **Information collected**: i) How partners are identified? Trustworthy tracking of service offers for contractual needs? ii) Does service knowledge carry explicit requirements information about the runtime service bindings? **Information made available**: i) Suitability to predefined collaboration structures information available? Collaborations evolvable or fixed? ii) Kind of semantic interoperability support? Does the available information in the ecosystem level databases suffice for interoperability testing? iii) Matching of services to collaboration structure is supported by an ontology or type system? iv) Is there any reputation information associated? Are the services trustworthy, traceable, attributed on their quality? Is the partner/service repository impartial?
Partner discovery process
Directed for browsing or automated matching, discovery by demand? Client-server or multiparty search with aim to contracting? Quick temporal partner selection or forming strategic networks? Private agent or third party or distributed? Considers interoperability and NFPs?
Partner/service selection:
Level of automation? If automated, areas of metapolicies for decisions (in what kind of situations automated decisions are permitted)? Style of trust decisions taken? Business needs addressed? Provides for automated eContract negotiation? For contract enforcement? Considers performance and utility aspects?
Service selection
Do service offers carry interface syntax; behaviour description; service provider; location; type description availability; awareness of resources; awareness of trust; dynamic properties in offers? correctness of information, traceability of announcements? Security and trustworthiness of offers covered?
Enactment and monitoring:
Scope: external processes only or integrated internal processes? Enactment: active agents or WFMC engine (workflow management engine) or model interpreter or translated process description to implementation? Semantic data transformations explicit or implicit? Breach detection: immediate or delayed? NFP with business issues vs technical SLA? Who are the controllers?
Dissolution:
BPI metrics? Who provides reports and when? How is information utilised? Reputation information model and processes?

Table 2. Classification questions for infrastructure facilities.

Ecosystem infrastructure facilities: We take the ecosystem infrastructure as an unbiased, trusted party, and all ecosystem members have systemic trust into its services for each of the collaboration lifecycle phases. The questions investigating the variance within available services are shown in Table 2. In addition, the comparison of solutions should take into account how different threat scenarios have been addressed.

Conceptual connectivity between viewpoints: While creating ecosystem models, a small set of essential concepts appear in closely related forms in different viewpoints. For example, a collaboration model under design in the engineering viewpoint will reappear as a contract structure during the collaboration, and eventually will enforce structure for distributing gains and losses for the collaboration members at the dissolution.

For a mature ecosystem model, we require these related concepts in different viewpoints be bound together in the lifecycle models. Connectivity should be defined for main concepts, such as contract, business service, breach recovery processes, and NFP (nonfunctional property) frameworks, just to name a few. In a mature ecosystem, the connectivity is managed by metainformation governance, and can be evolved as needed at the ecosystem level.

These connections are mostly missed when projects focus on one viewpoint only, but the consequences are serious: Interoperability and correctness failures are often caused by ad hoc transitions from one phase to another.

3 Pilarcos open service ecosystem architecture

The Pilarcos open service ecosystem architecture intertwine engineering, governance and operational needs of collaborations and thus involves:

- enterprises providing and needing each others' business services, with their published business service portfolios [2, 1];
- business-domain governing consortia, with their published business scenarios and business models [1];
- infrastructure service providers of individual functions such as service discovery and selection, contract negotiation and commitment to new collaborations, monitoring of contracted behaviour of partners, breach detection and recovery [7, 2, 10] and reputation flows from past collaborations [3];
- consortia and agencies that define legislative rules for acceptable contracts [6] and joint ontology about vocabulary to be used for contract negotiation, commitment and control [11, 6]; 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 [7, 11, 1].

3.1 Key concepts and functionality

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 between an agent and web service [12]. Each business services provides 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 [2]. 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.

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.

From the business point of view, the Pilarcos ecosystem provides for the maturity of ecosystems by addressing at the same time four intertwining tiers [13], as illustrated in Figure 1. The main ecosystem activities involve service engineering (left and bottom), ecosystem and collaboration governance (left and right), operational-time collaboration support (right and bottom), and ecosystem governance (rules within infrastructure in the bottom), as discussed below.

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: i) available services are published by service providers (enterprises including public and private sector providers), ii) the publicly known BNMs are created by teams of designers and published after acceptability analysis, and iii) 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. All created collaborations inherit suitable correctness criteria to be monitored at the operation time. This *modeling tier* is where service and collaboration in-

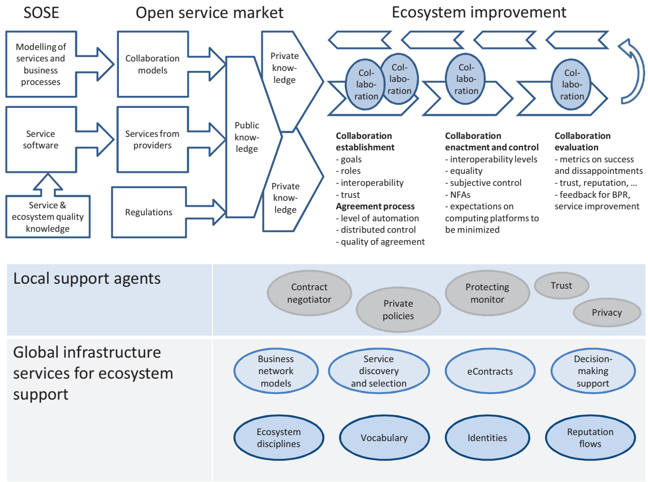

Fig. 1. An overview of the Pilarcos open service ecosystem.

novation take place, utilising the skills of designers and the feedback gathered from collaborations already operational. The methodologies to be used here apply service-oriented software engineering (SOSE) and model driven engineering (MDE) methods and tools.

The modeling tier and ecosystem repositories together give a basis for evolution of the ecosystem with service and collaboration models [11, 1].

The right side of Figure 1 depicts the *collaboration tier* supporting the lifecycles of collaborations from the establishment to the termination phase. The collaboration lifecycle management is automated in all routine cases, and triggers human intervention in new or undefined situations. The automated management decisions can be commitments to collaborations or refusals to participate. In practice, 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 in partners. In the enactment and control phase, the local support agents provide protective monitoring and the required contract-related communication.

In this way, the CaaS tier services become usable by enterprises or other organisations for making tenders, proposals, commitments, and to react to breaches, as well as initiating, negotiating, committing, and dissolving collaborations, and even, helping the subjective control of new kind of business transactions. The individual eContract is the key element for each collaboration as it governs that multiparty, dynamic agreement with details from business level to communication technology. The eContract also provides interfaces for each partner to notify their observations of the collaboration behaviour, deviations of the expected behaviour, their refusals to accept the recent progression of the collaboration, or their approvals on completing business transactions. The Pilarcos architecture emphasises on subjective and private decision-making support for partners on all collaboration phases.

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.

The bottom part of Figure 1 represents the global, federated infrastructure services that participates the governance, engineering and collaboration management processes, i.e. the *CaaS tier (Collaboration-as-a-service).* The CaaS tier includes ecosystem-widely available infrastructure services, such as service discovery and selection [2], eContracting [2], breach management [2], and reputation-based trust management system [3] that allows the scaling of the ecosystem membership. The scaling can be achieved only by creating incentives for parters to behave according to their contracts, and especially according to expectations at the ecosystem maintenance processes like reputation exchange and helps enterprises in adjusting to rapidly changing business situations and participation in natural competition between collaborations and ecosystem members.

The reputation-based trust management concept facilitates the scalability of the ecosystem. Here we can rely on social ecosystem studies [14]: 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 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.

Further, the *ecosystem tier* is the source of ecosystem level regulations, thus forming an explicit ecosystem engineering discipline. For each ecosystem this discipline has to be specialised individually.

3.2 Comparative details

Within the above frame, we take a more detailed look at Pilarcos using the maturity framework aspects. Comparisons to other systems in e.g. [13].

Innovation and engineering are addressed by the SOSE processes [1] producing BNM and service types into the infrastructure repositories (service offer repository, service type repository, BNM repository). Service type definitions form a basic vocabulary for declaring BNMs and publishing service offers, and can be reused. The BNMs can be designed collaboratively between multiple impartial organisations, and be then verified and validated for their suitability for the market domain. The vocabularies created by service types and BNMs eventually support the checking of pragmatic interoperability at the operational phase, as the business services in a collaboration do not have a joint inheritance hierarchy that would enforce interoperability.

The decisions on the partner selection and trust are postponed to collaboration establishment time. We have separated the model design phases from the collaboration establishment phase, to enable automation at the commitment phase and to separate the innovation phase as the actor sets involved in these phases differ. The traditional virtual organisation breeding environment way (e.g., in ECOLEAD and CrossWork [15]) of first choosing the partners and base the business processes on their capabilities actually forces the design phase for each individual collaboration, and the actors be shared in these two phases.

For acquiring correctness of collaborations [6], the infrastructure repositories must control the publication of offers or models following the rules provided by the ecosystem management. The control must consider traceability of the declaration makers, acceptability of the models in terms of best practices on a business domain, regulatory rules, and securing the coherence of the repository contents, especially the asserted relationships between stored concepts.

For the **collaboration lifecycle management** the key agents are the the private agents representing the involved enterprises and the eContract agent that governs the collaboration itself. The local support agents subjectively represent the enterprise, and provide a local interface to the ecosystem infrastructure services for the local business services. The enterprise agents are needed for tasks of 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 provides interfaces for application software or administrative interfaces to initiate collaboration establishment, or for responding to suggestions from other enterprises. The contract negotiator first utilises the populator for helping in the the initial service selection that is based on public information. As the populator provides suggestions for sets of interoperable partner services for the collaboration, the contract negotiator initiates the negotiation phase that involves private decision-making by all suggested partners. In the negotiation phase each suggested collaborating party can agree to join the collaboration, or refrain. The decisions are split to automatic rejections and approvals, and grey area cases that are forwarded for human decision-making with a kind of expert tool support [10, 3]. The decision-making is governed by enterprise policies [10, 3] related to i) strategic policies indicating what type of collaborations or which partners are of interest and worth investing the resources to collaborate with; ii) reputation-based trust that weights the anticipated risk and tolerated risk level [3]; and iii) 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 [3]. 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 [16]. Here, the subject can be a person, social group organisation or organisational group.

As the result of negotiation phase, eContract agent is created. It comprises of the collaboration metamodel thus providing a shared-language view on the collaboration structure, behaviour, policies and abstracted state. 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 logical eContract 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.

The eContracts include policies as rules of expected behaviour patterns. For policy expressions we use deontic logic [17]. 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.

In the enactment and control phase monitoring agents check the acceptability of the behaviour (messaging) [7]. The monitors receive rules from eContract and from their local enterprise policy repositories. The deontic-logic policy approach allows us to make clear distinction between violations of the contracts and acceptable behaviour according to that contract [18]. 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 some violation detected in the sequence of actions they get exposed to.

The private enterprise rules and eContract based 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. The enterprises may change their local policies during the collaboration and the arising contradictions can cause breaching business obligations, or failing quality of service agreements, 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. In case of an essential breach, the eContract is notified for triggering recovery steps. The breach recovery process is defined as part of the eContract, as there are different categories of theoretical recoverability capabilities.

At collaboration termination, successful or unsuccessful, experience reporting is required [3]. The local agent feeds reports to reputation flow agents that aggregate reputation information, arranged into several asset aspects including monetary, reputation, and control assets. The reputation information becomes available for future trust-decisions throughout the ecosystem. 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 [3, 16].

The **ecosystem infrastructure** provided by Pilarcos differs from related approaches. Instead of a simple service offer repository, in Pilarcos the service discovery and selection is supported by a *populator agent*. The collaboration initiator selects a model from the public BNM repository and invokes the populator to find matching service offers for remaining roles [2]. 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. Furthermore, the populator checks that the additional requirements indicated in all the involved service offers do not inhibit collaboration. New contract proposals can be picked within selected resource limits.

In comparison with other service offer repositories (UDDI [19], ODP/OMG trader [20]) 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.

Further, the negotiation phase is only refining the populator suggestion in terms of policy agreements and choices at the communication channel structures needed for dynamically configuring open bindings between business services. An open binding provides a constellation where distribution transparencies, transactionality support elements and security levels are selectable, and where a management interfaces stays available for the users of the binding [21].

For the enactment phase, Pilarcos does not include a business process execution engine, but business services are active agents able to independently trigger business process actions on each other. As the capabilities of the technical software supporting the service may be wider, policies and monitoring is needed to restrict that behaviour to the contracted or enterprise-widely accepted limits. Naturally, contracted behaviour limits are monitored only in the scope of external processes. Monitoring is enhanced towards the business-level NFPs [5]. Breach detection is designed to allow immediate resolution to take place, although each collaboration contract may have differently designed breach recovery processes captured in the eContract.

Conceptual connectivity is one of the cornerstones of Pilarcos development. For consistency enforcement, the ecosystem repositories are governed by several ontologies or heaps of metamodel hierarchies [11, 1]. The purpose is to connect innovation time and enactment time concepts together, and thus ensure that there is no conceptual misunderstandings caused by the change of modeling team tools to enactment time monitors. The conceptual connectivity facilities are based on a conceptual analysis that captures the key concepts and processes required by all ecosystems, and a methodology for creating tailored, evolvable ecosystems for a certain business domain [1].

4 Discussion

Our experiments on developing the Pilarcos ecosystems have created some opinions on the direction of future work, standards, and expectations on the requirements on the scientific base on the field.

The CINCO group mission is to develop a mature, dynamic ecosystem architecture for protecting organisations from interoperability mistakes, future needs of major collaboration platform change, and for supporting easy innovation in multiple, governed inter-enterprise environments. The Pilarcos contributions address the key problems in inter-enterprise computing: i) ad-hoc engineering and integration (can be within tools); ii) insecure and misplaced decision-making; and iii) missing control and governance of the systems composed.

We expect **mature open service ecosystems** to provide

- ecosystem infrastructure with management functionality involving embedded model verification and validation;
- private and public decision-making points that address the needs of business stakeholders and can be policy-driven but allow intervening;
- scalability through automation for breach detection and limiting of misbehaviour of the ecosystem members (incl. trust, privacy);
- systematic support and automation on collaboration lifecycle management involving autonomous parties;
- enhanced safety/correctness of collaboration lifecycles based on eContracts and underlying metamodel hierarchy; and
- a subjective, relaxed view to business transactions.

Open service ecosystems provide an environment in which enterprises (or organisations, even individuals) can easily pick a collaboration model, find potential partners beyond their normal strategic networks, and manage the lifecycle of the dynamic collaboration.

In order to adopt and trust ecosystem services, these enterprises need to preserve their autonomy and to gain understanding on how, and to which extent, the ecosystem services can ensure the correctness of collaborations. Correctness of collaborations can intuitively be connected to freedom of deadlocks and livelocks, fairness, consistency of the partners view on the state of affairs at each milestone of the collaboration, and conformance of the collaboration behaviour to the subjectively set policy requirements for the collaboration. Only some of these properties the enterprises themselves can enforce while the others must be produced by collaborative engineering and control functionalities at the ecosystem or at the collaboration level.

Furthermore, for the sustainability of the ecosystems it is essential that the evolution of the ecosystem is supported and the facilities provided by the ecosystem are scalable in terms of the ecosystem membership and service numbers and capacity of learning from experience. A future challenge is to appropriately interface organisational processes with the ecosystem agents.

Existing **standards** on the field are limited to singular business domains where tailored dictionaries and processes, or technological solutions are defined. In addition, trials of description languages for services and business processes have been made, leading much to the same expressive power. Inclusion of semantics notes is still not resolving the needs of composability, but a more systemic, and unfortunately, rather complex solution is required. This requires the courage to view all elements of the ecosystems at the same time. Some of the major cornerstones lie within the conceptual metamodeling hierarchy that is not prone to standardisation as such, but more likely to appear in forms of ecosystem engineering methodologies [1].

In order to choose appropriate candidate areas for standardisation, we must understand the evolution path of large systems. At present, this field has seen individual environments built, focused on one or two of the essential viewpoints. In addition, interoperability solutions based on tools and modeling have been tried; much of the present EU-level research is working with problems of the model and tool interoperability. This provides for collaboration management from a design perspective. The next major step is to push the capabilities generated by these tools into a common infrastructure in a generic form. This provides for ecosystem engineering and intertwining above-discussed perspectives. The new standards should not hinder this step, but the standards should be chosen from the areas supporting both the ecosystem engineering and the collaboration management tiers. It is likely that those standards may indicate different maturity levels for systems in different phases.

Good standard candidates include contract structuring, domain-specific collaboration modeling methods, innovation support, and an open binding infrastructure to mature ESBs.

The field is multidisciplinary in nature, which causes debates on the **science base or research and evaluation methodologies**. This interdisciplinary nature forces us to solutions constructed from elements from more than one scientific field. Considering our work with Pilarcos, we have applied several underlying computer science disciplines (such as extended state machines and protocol verification, even coloured Petri nets lately; patterns of reflective systems to control software artefacts with models; multi-agent technologies and type disciplines), complex adaptive systems theories, speech act theories with multi-agent systems, and basics of business, economy and psychology.

An essential goal in our work is that the ecosystem innovation and collaboration management environment should protect the engineers and especially the business-oriented users of the ecosystem facilities from most of these scientific considerations. Complexity should be hidden within the tools and methodologies, and the metainformation hierarchies built so that the complex rules are taken into consideration automatically, in the guidance of expert systems.

Considering the evaluation methods, the situation is equally complex. With Pilarcos, we started from constructive research, building prototypes and measuring their performance and balancing the cost with the achieved automation. We also had ATAM-like discussions with collaborating companies for validating question setting, focus of interest, and thresholds for adoption. On the side, we made threat analyses to understand security weaknesses of the introduced services. For the architecture as a whole we have done ODP-based modeling for major parts, giving concepts and processes a formalism beyond the functional verification. Recently some group members have taken the direction of design science. In addition, we have been pondering if there should be a basic benchmark defined for trust management and ecosystem governance [22].

Researchers can only make sure to catch for each project people with solid research skills in different, supporting disciplines and evaluation processes, and furthermore, ensure efficient cooperation despite the seemingly different goals. More importantly, researcher educators and funders should recall that overly multidisciplinary groups do not provide sufficient support for most researchers. Unfortunately, many of the current research financing instruments fail by requiring too much simultaneous multidisciplinary work that leads to progress by additional detail where structural innovation is needed.

Acknowledgements This paper draws from work in the CINCO group (Collaborative and Interoperable Computing, http://cinco.cs.helsinki.fi/) at the University of Helsinki. Over the years, the group has been funded by various company-related projects through TEKES, and the Academy of Finland.

References

1. Ruokolainen, T.: A Model-Driven Approach to Service Ecosystem Engineering. PhD thesis, Department of Computer Science (2013) Accepted for defence.
2. 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
3. Ruohomaa, S.: The effect of reputation on trust decisions in inter-enterprise collaborations. PhD thesis, Univ of Helsinki, Dept of Computer Science (2012)

4. Kutvonen, L., Norta, A., Ruohomaa, S.: Inter-enterprise business transaction management in open service ecosystems. In: EDOC2012. (September 2012)
5. 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)
6. Kutvonen, L.: ODP RM reflections on open service ecosystems. Computer Standards & Interfaces (February 2012) In press.
7. 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
8. Stojanovic, Z., Dahanayake, A., eds.: Service-Oriented Software System Engineering: Challenges and Practices. Idea Group Publishing (2005)
9. Papazoglou, M.: Web Services and SOA: Principles and Technology (2nd ed.). Prentice Hall (2012)
10. 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
11. Ruokolainen, T.: Modelling framework for interoperability management in collaborative computing environments. Licentiate thesis, University of Helsinki, Department of Computer Science (2009)
12. Payne, T.: Web services from an agent perspective. Intelligent Systems, IEEE **23**(2) (march-april 2008) 12 –14
13. Kutvonen, L.: Multi-tier agent architecture for open service ecosystems. In: Proceedings of First International Conference on Agreement Technologies, Dubrovnik, Croatia (October 2012)
14. Fehr, E., Fischbacher, U.: The nature of human altruism. Nature (October 2003)
15. 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)
16. Shen, Y., Miettinen, M., Moen, P., Kutvonen, L.: Privacy preservation approach in service ecosystems. In: EDOC2011 Workshops. (2011) 283–292
17. von Wright, G.H.: Deontic logic. Mind **60**(237) (1951) pp. 1–15
18. 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
19. 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
20. ISO/IEC: "IS13235-1: Information technology - Open distributed processing – Trading function: Specification". (1998)
21. Fitzpatrick, T., Blair, G.S., Coulson, G., Davies, N., Robin, P.: Supporting adaptive multimedia applications through open bindings. In: Configurable Distributed Systems. (1998) 128–135
22. Ruohomaa, S., Kutvonen, L.: Behavioural evaluation of reputation-based trust systems. In: 5th IFIP conference on Enterprise Interoperability (IWEI13). (2013)