

Service-oriented middleware for managing inter-enterprise collaborations

Lea Kutvonen, Toni Ruokolainen, Sini Ruohomaa, Janne Metso

Department of Computer Science, University of Helsinki, Finland

{Lea.Kutvonen|Toni.Ruokolainen|Sini.Ruohomaa|Janne.Metso}@cs.helsinki.fi

Abstract

Participation in electronic business networks has become necessary for the success of enterprises. The strategic business needs for participating in multiple networks simultaneously and for managing changes in these networks are reflected as new requirements for the supporting computing facilities. The Pilarcos architecture addresses the needs of managed collaboration and interoperability of autonomous business services in an inter-organisational context. The Pilarcos B2B middleware is designed for lowering the cost and effort of collaboration establishment and to facilitate the management and maintenance of electronic business networks. The approach is a federated one: All business services are developed independently, and the provided B2B middleware services are used to ensure that technical, semantic, and pragmatic interoperability is maintained in the business network. In the architecture and middleware functionality design, attention has been given to the dynamic aspects and evolution of the network. This article discusses the concepts provided for application and business network creators, and the supporting middleware-level knowledge repositories for interoperability support.

1. Introduction

The globalisation of business and commerce makes enterprises increasingly dependent on their partners. Competition takes place between supply chains and networks of enterprises.

In this competition, the flexibility of enterprise information systems becomes critical. The IT systems and development teams should be able to respond in a timely manner to

This article is based on work performed in the Pilarcos and web-Pilarcos projects at the Department of Computer Science at the University of Helsinki. The Pilarcos project was funded by the National Technology Agency TEKES in Finland, Nokia, SysOpen and Tellabs. In web-Pilarcos, active partners have been VTT, Elisa and SysOpen. The work integrates to a large degree with RM-ODP standards work, and has found an interesting context in the INTEROP-NoE collaboration.

the requirements arising from the changing co-operation networks and their communications needs.

Traditionally, inter-enterprise collaboration has been supported by business process driven solutions that focus on the business functionality needs and the technology-homogenising needs of the collaboration. This leads to situations where a change in the business processes induces large re-development projects. Furthermore, technology changes may cause domino effects cascading on the computing systems of dependent collaborators.

The present goal, instead, is to narrow the gap between business management concepts and the computing solutions. This introduces a new category of dynamic management aspects to the computing facilities, which isolates business processes from the technology, and thus improves the agility of enterprises when it comes to participating in new business networks. In addition, the wave of service-oriented computing facilities creates new possibilities for enhancing the automation of services as building blocks of many different types of business networks simultaneously.

In this work, the major challenge is to develop a middleware that takes the burden of managing these loosely-coupled collaborations and maintains the correct interoperation between business services in a way that supports business management concepts more directly. In comparison to traditional integration solutions, the global solution must lean on federating B2B middleware services that support the management of contract-governed collaborations, as will be discussed below.

We present an overview of the middleware solutions suggested by the Pilarcos project for inter-enterprise collaboration management. Section 2 first outlines the model of global networked business and the B2B middleware role in supporting it, addressing the new computing challenges that are arising. The activities addressed include negotiating and describing new business networks, developing new business services for the open service markets, contracting with partners about a collaboration and forming a new business network, acting as a partner providing agreed-upon business services, and monitoring potential risks and breaches during the activities. As the partners are autonomous and only contractually bound to a common goal, there is no technical guarantee of correct behaviour in the business or technical sense. Therefore, a feedback loop for creating a “social pressure” effect is needed: the architecture includes a reputation-based trust management system.

Section 3 in turn outlines the Pilarcos middleware architecture. Since interoperability knowledge is a key issue to address in the architecture, Section 4 gives additional details on the essential knowledge types and roles of the repositories globally available. This interoperability knowledge has several important roles in the architecture: the pieces of information play a role a) in the service and collaboration creation processes, b) in the verification and observation of interoperability at operational time, and c) as elements in the contract structures needed for defining the business goals of the collaborations, which thus enables the validation or breach detection in the business operation across the inter-enterprise collaboration. Section 5 gives further insight into the realisation methods of some management activities.

Finally, the discussion is turned to the relationship of the Pilarcos architecture to other current research and development directions, and concludes with usability issues, impacts and future work.

2. Model of business networks and their management

The world of business is going through a revolution that emphasises open markets and agility, and extends the concept of autonomous actors from enterprises to governmental organisations, user communities and individuals.

In order to provide the right set of concepts and processes to support inter-enterprise collaborations, i.e. business networks, in this context, we need to choose the global enterprise architecture; the selection of business management activities is dependent on that.

Our model for the global business environment of the future is based on open service markets. The enterprises and other autonomous entities can provide services freely in the network, and those willing to use the services are provided with facilities on finding and selecting the services, and contracting on their use. The support should cover not only the establishment of the business network, but the operational and termination phases too. Furthermore, the environment should provide support for gathering information about the trustworthiness of potential partners, and enforce regulatory rules on the business networks.

It is commonly expected that the old traditional business domain silos (such as metal industry, forestry, medical services, media) will lose their standing and be forced to compete with the more flexible service industry, where software-based services provide a major growth potential. Traditional examples of business networks include supply chains and subcontractor networks. Collaboration rules have become defined by best practices in the field, traditions, standards, or through the enforcing power of the leading role operator on the domain. Services have been developed slowly and with high investments; inter-enterprise processes for collaborations have caused problems in terms of re-engineering, and re-investments. The computing systems expected to support information exchange between parties are monolithic and thus difficult to adapt to new circumstances; they have different assumptions on the basic business processes, and fail in sharing understanding of the exchanged information.

Another strong trend is the emphasis on the role of clients in regulating and reforming services, even innovatively co-creating services, in addition to just using them according to a pre-described interaction pattern. Both trends bring up the need for viewing business scenarios as multiparty constellations, where each partner has a distinct role and a related interaction pattern to follow. Moreover, both trends bring up the need to take into consideration the autonomous nature of each partner in the scenario. With autonomy we mean the ability to make independent decisions on participation in business scenarios, technological choices, policies that govern security and privacy needs, and decisions on providing information or services to other parties, just to mention a few aspects. This is in contrast to the traditional collaborations between autonomous enterprises, for example in the form of supply chains or subcontractor networks.

In this inter-enterprise architecture, the essential concepts are business service interoperability and contract-governed collaboration between business services. We understand *interoperability*, or the capability to collaborate, as the effective capability to mutually communicate information in order to exchange proposals, requests, results, and commitments (i.e., to exchange speech acts common in business). The term covers technical, semantic and pragmatic interoperability. Technical interoperability is concerned with connectivity between the computational services, allowing messages to be transported from one application to another. Semantic interoperability means that the message content becomes under-

stood in the same way by the senders and the receivers. This concerns both information representation and messaging sequences. Pragmatic interoperability captures the willingness of partners to perform the actions needed for the collaboration. This willingness to participate refers both to the capability of performing a requested action, and to policies dictating whether it is preferable for the enterprise.

We call all types of the inter-enterprise collaborations *business networks*. This is because the business management activities appearing in different types of business scenarios (e.g., supply chains, virtual enterprises, and subcontractor networks; even digital business ecosystems) largely repeat the same pattern. Looking at the common business scenarios from the supporting technology point of view, we can separate external business processes that express what interactions the players in the business network must take, and the service processing software at the location of each player. The nature of the supply chain or virtual enterprise becomes expressed and defined by the business processes, while the supporting technical environment can be identical for all types. From the technical point of view, we can consider that the primary goal of each independent organisation in these scenarios is to provide added-value services by combining existing services provided by different enterprises. However, because of different responsibility models, which are important for the business management perspective, we need to preserve the following separation:

- orchestration, where the coordinator of the composed service takes on the obligations of providing the service;
- collaboration, where a mutual contract is formed and the members of the collaboration are equal and have their contracted obligations; the coordination of the collaboration is maintained by the supporting infrastructure.

Therefore, each business network is viewed as a collaboration between autonomously administered business services. A business network is established dynamically to serve a certain business scenario or opportunity that is made commonly known by publishing a business network model (BNM). The business network model captures all external business processes that are relevant for the business scenario. The business network model also gives structure to the eContract, which is in the technical core of governing the collaboration at runtime; the eContract captures most of the social behaviour requirements in the collaboration. A business service is realised by a business application implementation, running under the administration of a single authority. The wide potential of activities of the business application is restricted and controlled by enterprise policies to the degree that the enterprise is prepared to make available for its clients. The business services can fill in roles in multiple business networks simultaneously, based on their ability to fulfil the behavioural and nonfunctional requirements of the role and the eContract.

The methodology for building the business networks is semi-automated: based on a selected business network model and service offers published by service providing enterprises, the B2B middleware is able to suggest eContracts that are ensured to represent interoperable collaborations. The expected way of business services to fit into the collaboration and into each other is defined in terms of interoperability and collaboration contract requirements and breaches. These aspects are to be continuously monitored during the collaboration lifecycle, at times triggering management actions.

The set of tools supporting this methodology addresses the repeating business management activities for the different types of business networks:

1. *Designing the collaboration strategy and goals* involves business process design and re-engineering activities. In (Ruokolainen & Kutvonen, 2007; Kutvonen, 1998), we have proposed a service-oriented software engineering (SOSE) tool chain for producing business network models and service types with related property frameworks, and publishing service offers. This information is made available in globally accessible interoperability knowledge repositories within the B2B middleware. In (Kutvonen, 2002; Kutvonen, Ruokolainen, & Metso, 2007), we have explained how these meta-information elements carry the functional aspects of a collaboration, and where placeholders for nonfunctional aspects are located. This management activity is supported by a global network of business network model repositories, as discussed in Sections 4 and 5.

2. *Development and provision of new services* by an enterprise involves definition of service interfaces, and the policies governing the behaviour of the service in collaborations. This activity is supported by a network of service type repositories that make existing service descriptions available, and mappable to similar services for interoperability support.

3. *Involvement in eContract negotiation and contract establishment* comprises finding potential new partners and selecting from them, determining the services agreed on, the expected level of service quality, payments and costs, definition of breaches and resolution methods for breaches. In (Kutvonen, Ruokolainen, & Metso, 2007) we have proposed a B2B middleware layer represented by an enterprise-based agent to provide protocols for refining contract negotiation, state management, breach notifications, and interfaces for requesting services for suggesting a business network and joining or leaving a community. Furthermore, the enterprise's negotiation support services facilitate control of enterprise policies and memberships in the collaborations, for example. This management activity is supported by a global network of service offer repositories; the semantic interoperability of service offer repositories is maintained by service type definitions (see Sections 4 and 5).

4. *Management of each business service by the service provider enterprise* is addressed by local service management and policy management facilities. This includes provision of the services and provisioning of the interactions with other services. Further, it specifically must take into account the contract-driven government of services within the collaboration. As the services are independently administered, monitoring must be used for detecting potential deviations from the contract. In connection to this, 5. *monitoring of the operation according to functional and nonfunctional criteria* is supported by the B2B middleware facilities embedded in communication channels. In (Kutvonen, Metso, & Ruokolainen, 2005), we have addressed how the functional meta-information and monitoring are related. Both the management of business services and communication between them and monitors has been addressed earlier by the group (Kutvonen, Ruokolainen, & Metso, 2007; Kutvonen & Metso, 2005; Nurmela & Kutvonen, 2007; Ruokolainen, Metso, & Kutvonen, 2007). These management activities are further discussed in Section 5, which also shows how service type and eContract information is utilised.

6. *Termination of the collaboration either as the collaboration goals have been reached or due to breaches* is supported by the monitoring facilities and process descriptions embedded into eContracts for these purposes. In collaboration termination, it is possible to collect valuable information about the successfulness or failures of the collaboration, thus providing

further guidance for business-strategical decisions later, for example in the form of reputation information. Because the enterprises are autonomous, there are no technical means to enforce policies and behaviour on them; the social peer pressure supported by reputation and future potential for collaborations has a fundamental controlling role in the architecture. The mechanisms and effects on the collaboration establishment phase (Kutvonen, Metso, & Ruohomaa, 2006) and operational time (Kutvonen, Metso, & Ruohomaa, 2007) has been discussed in earlier work.

3. The Pilarcos middleware services and elements

The Pilarcos architecture proposes a model of inter-enterprise collaborations as eCommunities, consisting of independently developed business services. A business service denotes a set of functionalities provided by an enterprise to its clientele and partners. It is governed by the enterprise's own business rules and policies, as well as by business contracts and regulatory systems controlling the business area. Furthermore, as the business service is realised by software, the service is also defined by the computing, information representation, and communication facilities used and required.

The functionalities supported by the B2B middleware include

- a set of B2B middleware services for establishing, modifying, monitoring, and terminating eCommunities, or looking from the business service point of view, operations for joining and leaving an eCommunity either voluntarily or by community decision and leaving a trace in the global business world about the success of the collaboration; and
- a set of repositories for storage of meta-models for communities, ontologies of service types, and services, to support interoperability validation.

The business service providers are responsible for providing supporting meta-information to the B2B middleware repositories, but are otherwise freed from implementing any of the eCommunity life-cycle management. Instead, they are expected to use local middleware services for it.

The eCommunity life-cycle is mainly controlled in an eCommunity contract, or eContract. The eContract comprises the business network model, BNM (to define the network structure), information about the member services at each role, information about the expected nonfunctional properties of the services and their mutual communication (such as nonrepudiation, security, privacy-preservation), some overview state information about the progress of the external business processes, and methods for changing the contract itself.

In the eCommunity establishment process (and operational time management), the eContract is used to gather together all relevant information about both the business and the technical level details for the eCommunities. Figure 1 illustrates how contractual information derived from different sources becomes part of the eCommunity contract, and is used to govern a computational service in order to bring it up to represent the intended business service. In the following, these steps are discussed in more detail.

The strategical requirements of a business network are expressed as a meta-level model that defines a set of external business processes (upper right corner of Figure 1). We call this the business network model. The structure is defined in terms of roles and interactions between the roles. For each role, assignment rules define additional requirements for the service offer that can be accepted to fulfil it, and conformance rules determine limits for

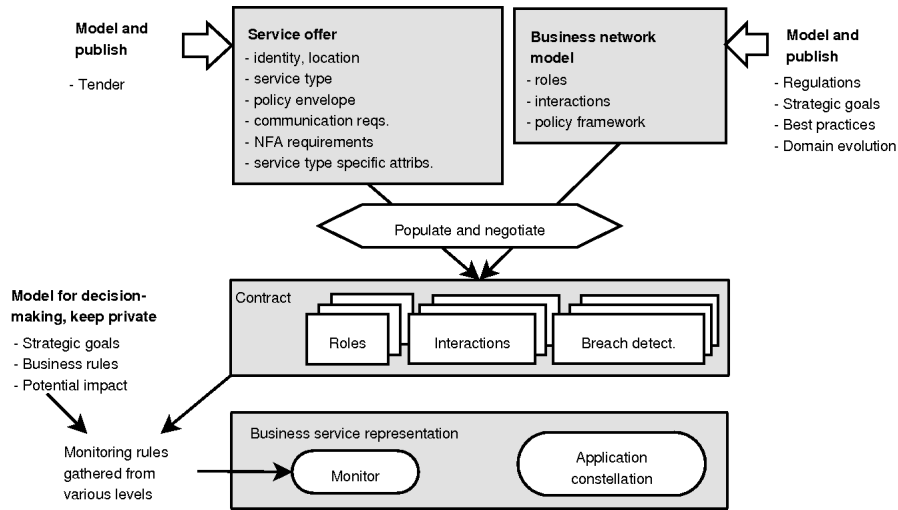


Figure 1. Source of business-level and technical-level information to control software behaviour to fulfil the business service commitments and restrictions.

acceptable behaviour during the eCommunity operation. The explicit use of such a model allows comparison and matching of strategical and pragmatic goals of members in the network by giving a working structure for comparable and negotiable service offers. The business network models should take into consideration all relevant legal and regulatory systems in the application area.

An enterprise that is able to run a computational service, i.e., an application constellation of software components providing interfaces for functionalities of a business service, can make it available for other parties by publishing its interface and property information (upper left corner of Figure 1). In addition, we expect the service offer to be considered as a commitment to provide the service with the identified properties, terms, and conditions. The information elements required in the service offer are determined by two aspects: first, by the requirements of the B2B middleware concepts, and second, by the mandatory properties defined for the service type in question, which will finally match the business network role requirements. Essential for the offer structure is that the contract terms relevant for the business area become represented in a way that allows them to be compared.

The contracting process between the business services is governed by the selected business network model. The basic properties of the business service become defined by the service offer, although the mechanism does not technically enforce the offer to be truthful or the service implementation to conform to the offer. However, in business terms, the enterprise loses credibility by false offers, and increased certainty levels can be acquired by external conformance testing and certifications. The process of enforcing enterprises to provide accurate service offers is mainly an organisational issue, not fully addressable by technical solutions. Computationally, it is possible to control that exporters of service offers are authorised by their organisations for making external commitments, and that there are technical facilities to follow the thread of delegations and negotiations for determining the party responsible for each commitment.

The resulting contract object contains both business network regulations and the agreed constraints for joint behaviour. This context information is configured, in suitable phases, to the monitor object governing all service requests passing in or out through the computational services interface. Thus the business rules and terms are passed to the monitor for controlling that the actual business service behaviour is not violated by the computational service, which is capable of more varied behaviours.

Because the business services are provided by autonomous service providers, there is no inherent guarantee that they would form an interoperable eCommunity. Therefore, special functionality for interoperability checking takes place when establishing a community, or entering a new service into an existing community. The applications themselves need only to concentrate on the local business logic, implemented on their local computing platform. For the eCommunity management functionality, it is necessary that the underlying B2B middleware is able to evaluate the interoperability of business services based on their service offers and to monitor the interoperability during the operational time.

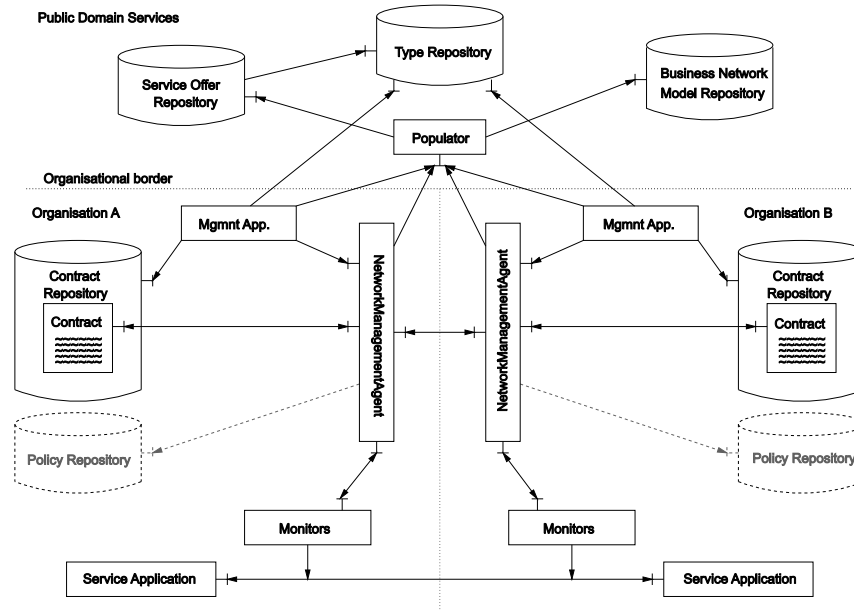


Figure 2. Service agents of the operational environment. Arrows represent communication relationships, boxes are active agents, and cylinders data stores.

The main functional elements needed for establishing and controlling eCommunities can be seen in Figure 2. The following will explain the main functionality of each module.

The upper part represents the breeding environment services, including the populator, service offer repository, business network model repository and type repository. These services can be placed in the public domain to be used by any enterprise. Breeding environment services like populators and type repositories are not required from all sites, but can be provided as infrastructure services as a business in its own right.

For the support of the populator, the business network model (BNM) design process involves the introduction and verification of new models to be stored into the repositories.

The implementation of new services or the introduction of legacy applications involves interaction with the type repository. New business services are published for use by exporting service offers to the corresponding repositories. Deployment processes are naturally augmented with service offer exports. These processes feed in meta-level knowledge of potential participants in communities to be formed. The feeding processes are independent of each other, and even withdrawing or deprecating information may take place.

The two lower parts of Figure 2 represent two autonomous enterprises. By autonomy we mean the potential for control over the private computing systems, and moreover on strategic business processes and policies. Each site or administrative domain, representing an autonomous ICT system, is expected to run a business process management agent. The lower part of the figure also shows the network management agents (NMAs) and eContracts as the major players of the operational environment. As discussed in the previous section, the eCommunity contract captures shared meta-information about the collaboration. At operational time, reflective methods are used to keep the real system at each involved computing site in correspondence with this meta-information. At each administrative computing domain, there is a local agent for management of knowledge about locally deployed services. The local management interfaces are homogenised by a protocol for requesting the system to prepare for running a service (resourcing), querying about communication points, releasing the service, etc. Likewise, all relevant changes in the real system are notified and thus change the meta-information accordingly. The eCommunity contract is an active object itself, and includes logic that may react to changes in the meta-information and request local sites for further negotiations or changes in the system state.

The *Populator* uses a given BNM for ensuring the pragmatic interoperability of partners to an eCommunity; it also uses a set of compulsory aspects in service offers to determine service types, communication channel requirements, and nonfunctional aspects to be agreed upon for the eCommunity. The populator represents a breeding process where services are selected for eCommunity roles. The population process is a constraint satisfaction challenge between candidates' attribute value spaces and constraints given for roles in the business network model. The service type definitions dictate the attributes and attribute value sets necessary to describe the service, and the actual values for each published service is found in the service offer repository. As there are dependencies between selected offers in interacting roles (on channels and NFA), the process is complex. The populator provides its clients with a set of interoperable communities from which to choose during negotiations. Replacement of partners in an existing community, or one partner changing to a significantly different service implementation are also situations where interoperability preconditions need to be checked.

The eCommunity management is performed in cooperation with *Network Management Agent (NMA)* and the *Contract* object. The NMAs are responsible for managing the inter-organisational coordination and management protocols, while the contract object is responsible for making decisions regarding the eCommunity it represents. The NMAs are located so that there is an agent acting as a representative between the eCommunity and the local service-providing system at each administrative domain. The contract object is made available across all involved domains; redundancy is required for availability to be ensured also in most common failure situations. The NMAs provide three interfaces. For the local administrator, there is an eCommunity management interface for triggering renegotiations

on conditions and memberships. Between each other, the agents have a protocol for notifications of task completions and contract breaches, and another interface for negotiation and commitment protocols for joint contract changes. For communication with the local monitors, the NMAs provide an interface for receiving notifications of contract breaches and task completion and another for feeding monitoring instructions to the monitor. The protocols are described in detail in (Metso & Kutvonen, 2005). It is essential for the NMA role that it acts on behalf of the administrative domain it represents, and is able to access relevant policy information, which is private for the enterprise. Based on local policies and guidelines, the NMA is able to enforce decisions, for example, on the significance of the breach notifications received from the monitors, and subsequently decide on which protocol to trigger between the NMAs.

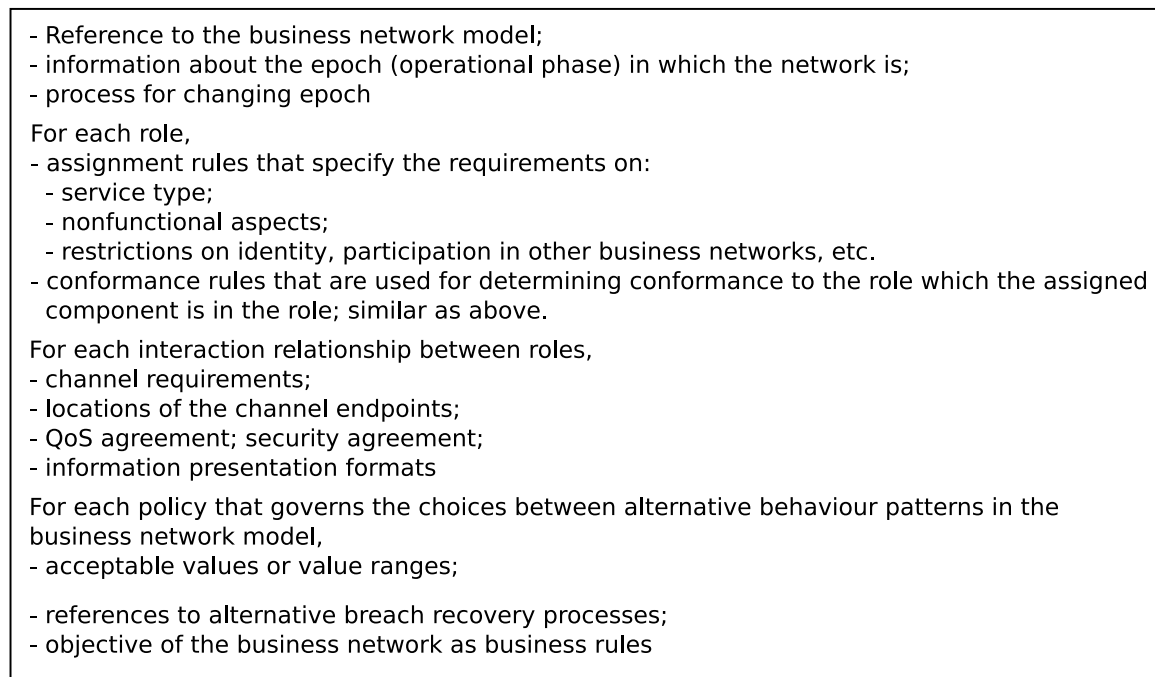


Figure 3. Information contents of the eContract.

The *eCommunity contract* is itself a key element in the architecture, because it makes aspects from different levels/viewpoints of the business network available at operational time. The community contract describes technical, semantic, (external business) process-related, and pragmatic aspects. Technical information includes service types and related behaviour descriptions, binding types between services, implementation-specific messages or function parameters, and policies used in the eCommunity. The structuring element of the contract is the business network model (BNM) used for the eCommunity: each role is supplemented with information from the participant's service offer, each binding with connector parametrisation information. Semantic aspects cover information representation formats in messages exchanged. The pragmatic aspects covered include functional description of business processes, policies constraining roles, and nonfunctional aspects. The nonfunctional

aspects govern features like trust, security and QoS, which are traditionally considered to require additional platform-level service solutions. In addition, nonfunctional aspects related to business process models capture more business-oriented features, like business rules (captured as policies and monitoring rules here). These main elements of the eContract are presented in Figure 3.

The structure of the contract is directly determined by the business network model in use. Most of the contract structure is a copy of the associated business network model. The business network model lacks identification of the business services as members, and only gives acceptable ranges for some negotiable policies. The business network model is also independent of the bindings (interaction support) the business services need to deploy among themselves. The model can, in addition, set requirements for the services beyond those set by the service type, such as support for transactions.

Monitors are part of the communication channel between participating services. A monitor has a generic sensor element that can be configured to filter traffic by classifying it to expected and unexpected event sequences (task started / completed, unacceptable traffic or lack of expected traffic). The network management agents provide each monitor with a behaviour automaton to follow, based on the service choreographies described for the corresponding role. The monitors can be used both to monitor the behaviour of the roles provided by their own organisation and the roles of the other organisations. Monitoring reports can be acted on in various ways, scaling from post-operational auditing to proactive prevention of unwanted events. In Pilarcos, the intent is to allow major breaches on agreed behaviour or policies to be acted on during the eCommunity operation, and allowing automatic recovery processes to be started. In this respect, the Pilarcos approach differs from related projects (like (Neal et al., 2003)) that otherwise use similar techniques. Because the definition of “severe breach” and the appropriate methods of potentially replacing misbehaving partners are specific to each application domain, those rules and process definitions are compulsory parts of BNMs.

The monitors have two very important tasks in the architecture, both involved with business aspects and (mis)trust in the global markets.

First, the monitors provide a method for enforcing the business-level policies and enterprise-wide operational policies onto the computational service constellations. In the operational environment, the monitor and the computational service constellation together form a representation of the business service.

Second, the monitors report suspicious events to NMAs, thus triggering announcements on misbehaviour or contract breaches, and providing a data source for reputation information.

4. Interoperability knowledge and its management

The three meta-information repositories in the B2B middleware – business network model repository, service type repository, and service offer repository – have a central role in establishing a knowledge base that allows interoperability tests on to be made. Essential target concepts are service types, service offers, and business network models. Each repository is distributed for scalability and improved accessibility. Due to different types of load, the good distribution styles differ (Kutvonen, 1998).

Service types and business network models (BNM) have separate life-cycles. This provides isolation layers that keep local changes from involving the whole eCommunity, and minimises the effects of BNM enhancements to local services. Furthermore, the creation of each model requires only a reasonably narrow expertise. In addition to direct relationships between models, the repositories store transformation rules and components for improved transformer/interceptor re-usability (Kutvonen, 1998). We will return to transformers and interceptors later in this section.

In addition to these three repositories, reputation information networks feed information about the experience gathered in eCommunities about the behaviour of business services.

The service elements of the Pilarcos breeding environment address the need of joining four important processes:

- introduction of BNMs to the model repository, and introduction of supporting service types to the type repository;
- software engineering processes to provide implementations that correspond to the known service types and thus are applicable for the known BNMs;
- deployment of services and export of corresponding service offers to traders, effectively making a commitment to keep the service consistent with the service offer;
- an eCommunity establishment process using the provided information.

These processes are only loosely interleaved. Business network models and the actual business services can be developed independently from each other; indeed, their development forms quite separate professions. In the platform, these concepts have to meet at the service description level.

Type and service offer repositories

The *type repository* provides a structured storage for type information related to services and their access interfaces. Operations are provided for publishing new types, comparing types, and creating relationships between types.

Service types are abstract descriptions of business service functionality. Service descriptions are used to ensure technical connectivity, semantic interoperability and behavioural compatibility in possibly heterogeneous environments. Service descriptions do not expose internal properties of business services, as this decreases the possibilities of reuse and evolution of services. Implementation-specific information, such as binding of a service into a specific communication protocol or address, is not covered by service type. Service type is like a contract, which an actual service must implement.

Service types are XML-based descriptions which define interface signatures, service attributes and an interface protocol. Interface signature in Pilarcos is described using a WSDL description without technical binding information (see (WSDL, 2001)). Each service supports only one kind of behaviour; different behaviour implies different service types. We refer to the definition of service behaviour as *interface protocol*, which is a behavioural description defining externally visible behaviour at one endpoint of a bilateral communication. Interface protocols in Pilarcos are based on session types (see (Takeuchi, Honda, & Kubo, 1994; Gay & Hole, 1999)). For behavioural descriptions, we have a simple XML-based process description language. Semantic interoperability of services is supported by binding ontological concepts to the exchanged documents. XML-based ontology description languages,

such as general-purpose description languages RDF(S) and OWL (RDF-S, 2004; OWL, 2004) or more specialised XML-based ontologies such as RosettaNet, can be used (RosettaNet Consortium, 2004). The rules of the type system are based on behavioural session types, structural matching of syntactic information and semantic relations based on description logic (Takeuchi et al., 1994; Jha, Palsberg, & Zhao, 2002; Nardi & Brachman, 2002). Subtyping-like relationships that support service evolution are also important (Di Cosmo, Pottier, & Rémy, 2005; Gay & Hole, 1999; Nardi & Brachman, 2002).

The type discipline in the Pilarcos platform is strictly managed. Every type definition must be contained by a type repository. Each type name, i.e. URI, must also identify the type repository responsible for managing the corresponding namespace and its type definitions. Without strict management of typing information, it would be impossible to ensure that types are unambiguously named, persistently stored, verified to be correct, and relationships between types verified and intact (Kutvonen, 1998). Type repositories can also be organised into a hierarchy for partitioning of namespaces.

Service types are published by institutions responsible for a business domain or by enterprises willing to promote use of new kinds of services. Standardisation of a new service type is not necessary, however, because the applicability and adoption of the service type is determined by peer acceptance.

The relationships of interest for the type repository users are: no match, similar types (equality of text or reference, subtyping), and interoperable with interception. The comparison and judgement is not fully automated and cannot be made (due to performance issues) at the time of query. Instead, the service type publication process involves verification of the type, comparison to other named types, and verification of the type relationships. The process of creating interceptors (i.e., transformers that change the inputs and outputs of one interface type so that they become suitable for another) is external to the type repository. The service types can thus be matched with each other in a more relaxed way, only limited with an interoperability requirement. As an enhancement, the cost of connection can be added to direct users to choose “native” types instead of transformed connections.

The initial Pilarcos type repository was designed and developed during the work on the ODP type repository function standard (IS14746, 1999), and OMG MOF specification (MOF, 2002). Although there are certain differences, most interfaces are similar. Thus the type repository offers operations (Kutvonen, 1998) for

- publishing realisations of abstract types,
- checking whether two type realisations are conformant and interchangeable,
- retrieving subtypes or supertypes of a type realisation,
- retrieving templates for a given abstract type,
- translating one type realisation to another,
- retrieving names for abstract types and type realisations in other type domains.

The *service offer repository* refers to services (like UDDI (Belwood & et al., 2004) and the ODP trading service (IS13235, 1995)) for locating services that are published using a structured meta-information description of the service. We consider these descriptions commitments to provide the service. When a new service offer is published, type repository functionality is used to validate the conformance between the offer and the corresponding service type.

Given a service type and a service offer, the conformance validation algorithm verifies

if the service offer corresponds to the behavioural and structural properties of the corresponding service type. Concerning behaviour, each communication action described in the service offer has to be matched with the communication behaviour defined in the service type. Both substitutability and compatibility relationships can be used as a behavioural matching criteria. For structural properties, similarity between the document structures used in the service offer and document types declared in the service type must be matched.

Functional properties of the type checking algorithm are characterised by the service-typing rules which are based on the session typing discipline (Takeuchi et al., 1994; Gay & Hole, 1999). The session typing discipline provides formal characterisations for service substitutability and compatibility which are based on the notions of session subtyping and duality (Vallecillo, Vasconcelos, & Ravara, 2003). The algorithm for validating session subtyping is syntax-driven and is expected to be efficient in practice (Vallecillo et al., 2003); however, no formal validation of this claim has yet been given.

If the conformance validation is successful, the service offer is published into a service offer repository with the claimed service type. The service offer publishing process requires predefined service types. In the Pilarcos architecture, we expect service types to mandate properties for expressing issues affecting technical, semantic, and pragmatic interoperability. This is reflected by the structure required by service offers, as illustrated in Figure 4.

- Interface syntax and protocol; either IDL or WSDL specification for the service interface, and partial ordering rules of operations;
 - schema for information elements;
 - list of nonfunctional aspects, such as QoS offers (acceptable range for QoS), trust requirements, or name for security mechanism to be used;
 - requirements for platform;
 - requirements for communication channel service level

Figure 4. Structure of service offers in Pilarcos.

Business network model repository

The business network model (BNM) repository provides interfaces for publishing models, verifying their properties, and comparing and querying models for population or software engineering processes.

The structure (topology) and properties of a business network are defined by its BNM, which explicates the roles of partners and the interactions between roles that are needed for reaching the objective of the eCommunity. A BNM comprises a collection of roles, a set of connectors and a set of architecture-specific nonfunctional properties. The approach combines ideas from the ODP enterprise viewpoint language (IS15414, 2003) and those of separating functional units and their interconnection into distinct concepts of components and connectors (Allen & Garlan, 1994).

A role represents a logical business service or entity in an administrative domain. The role definition expresses the functional and nonfunctional properties required. Role functionality is described as a composition of service types and role-specific synchronisation patterns. Synchronisation patterns express causal relationships between actions in distinct services of a role (by setting preconditions for interactions using the terms before, after etc).

Interaction relationships between roles are described by bilateral connectors between service interfaces. Connectors may define other communication-related properties, such as control or data adaption, eCommunity coordination and nonfunctional properties of communication.

Nonfunctional properties are managed as named values that are used for selecting the right technical configurations from the underlying platform. Some properties are used for dynamic branching of behaviour at operational time. These decisions stem from the business level, but the negotiation and commitment protocols needed are preferably transparent to the business services.

The elements of the business network model descriptions required are presented in Figure 5.

- A set of named business process models, each presented in terms of named roles with
- role assignment rules for guiding the populator to retrieve suitable service offers from the offer repository;
- role conformance rules for guiding the monitors to notify its BNA about all deviations from these rules;
- interaction pattern expected at each communication channel; may be simple invocation, announcement or full session specification;
- name for the breach recovery process to be used;
- role association requirements expressing how a business service is required to simultaneously act in named roles of named business process models; this allows functional slicing of the model for reuse and design purposes;
- policy rules overarching the business network model; these policies provide invariants that need to be fulfilled;
- breach recovery process name and type (type for example "dissolving", "restarting", or "continuable with sanctions")

Figure 5. The contents of a business network model.

Reputation information

The overall architecture assumes that terminating eCommunities provide information to the global network of their satisfaction to the past collaboration. This information gives a basis for predicting the likely behaviour of a partner in an eCommunity, and thus facilities for deciding on whether an action is worth the related risk of partner failure.

The required information must be associated to the business service in question, not only to the enterprise providing it, or the technical environment supporting it. Naturally, changing any of these aspects will change the reputed target as well. On the other hand, persistent enough identification of the responsible service providers is essential.

The reputation information is accumulated on two forums; it is the current, private view of a trustor's trustworthiness formed from local experience and shared third party experience. The reputation views building on these two very different sources are stored separately up until the moment of a trust decision.

Both external and local reputation views follow the same format. They count the outcomes of each type (experiences) for each asset, and tag the information with credibility information.

The assets selected so far are named monetary, reputation, control and fulfilment. The monetary asset can denote only concrete costs and gains, while the reputation asset encompasses both the enterprises reputation rating in any reputation systems, as well as the more abstract notion of its public relations, appearance in the media, and the attitudes of its associates, partners and customers towards it. The control asset represents the general need for an enterprise to protect itself from outside influences: to maintain control over its security, privacy and other aspects of its autonomy. The fulfilment asset is the one most tightly connected with a trustee. It encompasses whether the trustee does its part of what was agreed, leaves something relevant undone or does something it was not strictly expected to. Where the base for the monetary asset is the wealth of the organisation, the base for fulfilment is the general trend of respected agreements, which reflects on the success of the organisation.

The outcomes of actions are represented through their effect on assets: major negative effect, minor negative effect, no effect, minor positive effect, major positive effect, and unknown effect, if the outcome could not be determined for a particular asset (Ruohomaa & Kutvonen, 2008).

Relationships of metadata

The meta-information repositories' contents are interdependent. A verified business network model acts as a template for the eCommunity. The model to be used as a contract template is first negotiated between the potential partners, involving comparison and matching of strategical, pragmatic goals of members in the network. As the matching of multiple network models is too hard a problem to solve by an automated process in general cases for a heterogeneous modelling environment, we require a single shared model to be agreed on at the eCommunity establishment. Checking that all parties expect the same business network model is one of the pragmatic interoperability aspects.

Within the business network models, service types are used as means to define requirements for role players. Again, the matching problem is too hard in a general theoretical sense, and therefore we have focused on practical goals: grouping of similar models, and identifying suitable transformers or adapters available when similar models need to be mapped together.

The service type repository is used for holding such a relationship between models and the associated transformation information. The actual adapters are produced in a separate process starting from the service type descriptions (Kutvonen, 2004b) for configuring a communication channel between peers so that the information exchange is understood correctly and there is no known deadlock in the sequence of message exchanges. The adapters can address modifications at multiple levels of interoperability, such as data representation modifications, and changing the communication pattern (for example, splitting a request of a task to a set of requests for subtasks from the peer).

The service publication functionality is similar to the UDDI (Belwood & et al., 2004) or the ODP trading mechanisms (IS13235, 1995); the type management system resembles the ODP type repository function (IS14746, 1999) and enforces a typing discipline to follow over service offer repositories.

In the service publication process, service providers send service offers to the service offer repository, to state claims about the type and properties of the services. A service offer

describes functional and nonfunctional properties of the service to be published: the actual service interface signature, service behaviour, requirements for technical bindings (e.g. transport protocol), and attributes such as service quality and trust-related commitments. The service offer repository then initiates a conformance validation process. For this purpose, a service type corresponding to the claimed service type is retrieved from the type repository. The service type defines syntactical structures for service interface signature and messages, externally visible service behaviour and semantics for exchanged messages (Kutvonen, Ruokolainen, & Metso, 2007). Conformance validation is executed by the service type repository holding the corresponding service type. Only after a successful validation, the service offer is published, otherwise a service typing mismatch is reported between the service offer and its claimed type.

The BNM repository is a shared storage of business collaboration information that enables enterprises to share business transaction models, similar to the ebXML-repository (Kotok & Webber, 2001), although with a more automated and repeatable breeding process. The notations used are not discussed here, but they resemble the ODP enterprise language and use XML-style notations (see (Kutvonen, 2004a) and (Kutvonen, Ruokolainen, & Metso, 2007)).

For each eCommunity establishment process initiated by a willing partner, the corresponding business network model is first fetched from the BNM repository. The population process provides a set of interoperable eCommunity proposals where the roles of the BNM are filled with potential partners. For this purpose, the type repository is consulted for providing service types matching the requirements of the business network model, after which the service offer repository can be used to provide the corresponding service providers. After population, and the subsequent negotiation, the eCommunity contract is received and distributed to every participant.

The service interoperability and correct operation of the community assumes that the meta-level information on BNMs, service types, and service offers is correct. Therefore, we find it necessary to collect the meta-information into repositories, where the trustworthiness of the information source can be controlled, and the quality of the information can be validated by the repository management actions. These aspects must be weaved into the tasks involved with eCommunity establishment, such as service publication or discovery (Ruohomaa, Viljanen, & Kutvonen, 2006; Kutvonen, 1998).

5. eCommunity management aspects

This section presents some key management activities or aspects. First, development of business network models for the breeding environment is discussed. Then two aspects of the eCommunity lifecycle are detailed: semi-automated decision-making support for enterprise systems for participating business network activities, and verification and observation of interoperability.

Designing the collaboration strategy and goals

Traditionally, the establishment of business networks starts by negotiation of the joint objectives and goals, and collaborative definition of the joint processes, and definition of the methods of connecting individual computing elements to a coherent whole. This phase

is supported by breeding environments where selection of partners, learning about their capabilities, and designing the joint business network model takes place. In this process the set of functionality is determined, as well as a set of business policies that must be adopted.

Although all this is necessary for the business network establishment, it is not necessary to perform the whole process independently for each business network. Neither is it necessary to repeat the whole process when partners have wishes to make changes to the collaboration goals, processes, or supporting applications or computing platforms.

We have separated the business network design phase from the network establishment phase. On one hand, the business network models can be collaboratively designed, verified and validated for their suitability. On the other hand, these models provide a common vocabulary for enterprises to use at the business network establishment negotiations: When a collaboration is being established, the pragmatic interoperability (including processes and policies) is tested between partners. This means that the business services forming a collaboration do not necessarily have a joint history in the breeding environment that would enforce interoperability, but can simply be introduced to each other in a refining negotiation of the eContract.

As a consequence, the architecture

- uses policies for refining the process models;
- must include operational-time facilities for ensuring interoperability and breach detection between the partners;
- uses policies for constraining the acceptability of a business service in a role of the business network; and
- is able to support dynamic changes on the policies during the operational time.

In the business network model designs, it is beneficial to create rather abstract behaviour groups within which actual business networks can vary, in order to support a wide potential for collaboration and evolution. Within each model, more precise alternative behaviour can be chosen by setting the guiding policy value at the eContract.

The ability to perform dynamic policy management is a strong tool: selecting policy languages and targets suitably, most business management needs related to strategies and business rules can be modelled and transformed into rules that can be monitored at runtime. Effectively, the introduction of different types of policies allows mapping business domain guidelines directly to B2B middleware facilities.

The design of business network models is by nature a distributed activity: the resulting model should be acceptable for the business domain, represent the best practices, and even follow the regulations on that business domain or domains addressed. Therefore a common vocabulary is needed on-line for the designers to use, and strong guidance towards reuse of existing business process models is necessary.

Considering the present business process definition languages (e.g., BPMN, WS-BPEL, XPD, WS-CDL; survey (W. van der Aalst, ter Hofstede, & Weske, 2003)) the main concepts are roles and interaction relationships between them. Then, the assignment criteria for roles, and interaction patterns are determined. We do not deviate from this general direction, but emphasise some special features that are relevant for service orientation, nonfunctional aspect management, and evolution support on the service markets.

First, the created models are published in an abstract (black box (Norta, 2007)) form, only revealing the obligated interactions between roles. This view is then to be refined by

other design and configuration phases. The business network models are constructed by connecting together business processes that each have a single starting point, single termination point and one functional goal (which is essential for verification purposes as well (W. M. P. van der Aalst, Verbeek, & Kumar, 2001)). Connecting the processes together takes place by explicating which roles at each process model must collocate at a combined role. The new role inherits the service requirements from all these collocated roles. The business process models are annotated by criteria for assignment of business services and operational time criteria for not causing a breach. When the combined roles are created, annotations are added for restricting collocations, for example, to avoid legally invalid combinations where something expected to be externally supervised performs the supervision role itself.

Second, the business rules for the collaboration are explicitly defined, and thus changeable. The rules determining breaches are explicit, as is the agreement on what recovery process to use. For this purpose, a) multiple recovery business processes should be defined and consistently viewed as a set of best practices definitions, and b) all business network models should be analysed to determine their recoverability style. Some networks are not able to recover from breaches but need to be terminated, while others may recover from the loss of some members, and further some require a set of compensation actions to take place before either continuing operation or terminating.

Third, all business network models should be analysed and verified for properties like liveness, fairness, privacy-preservation (data flow sufficiency and minimality), termination of processes, and recoverability.

We have found it natural for the process designers to work process-wise; however, due to the autonomy of the domains providing the actual services, the models must be at least transformed to a role-based model. The basic structure of the business network model comes from the roles and functional interactions, while nonfunctional aspects are added on to collaboration, role and interaction levels. Each nonfunctional aspect may have its own domain-specific aspect language in use: the expressive requirements fall to the categories already present in current rule languages (e.g., RuleML (Grosz & Poon, 2003)), policy languages (e.g., Ponder (Damianou, Dulay, Lupu, & Sloman, 2001)), service level agreement languages (e.g., WSLA, WS-Policy; survey (Nurmela & Kutvonen, 2007)), or eContract languages (e.g., BCA (Milosevic, Jøssang, Patton, & Dimitrakos, 2002), (Goodchild, Herring, & Milosevic, 2000)).

Decision-making support

As the negotiation of the business network structure and goals have been factored to a separate step that results to an explicit, published model, the eContract negotiation between enterprises becomes more restricted in its scope. Effectively, the negotiation must result into a situation where it is ensured by static validation that interoperability at all levels exists between all parties, and that all parties are willing to participate in the collaboration. Furthermore, the refining negotiation must select the policy values to be used for this particular collaboration and stored into the eContract.

The supporting facilities to be used here are as follows (Kutvonen, Ruokolainen, & Metso, 2007; Kutvonen et al., 2006). First, the B2B middleware provides population of the business network followed by a generic negotiation protocol between the enterprise agents. The population process ensures that according to the claims in service offers the business

services becoming members of the business network can be interoperable at all levels. Then, the proposed eContract draft is set to each enterprise to gather commitments of participation, or further refinements on the policies suggested. In the service offers, ranges of policy values are announced, so there is sufficient potential of finding shared values from the shared domain of potentially acceptable values.

The negotiation cycle ensures privacy of decision-making for each participant. In routine cases, it is possible for the agent that represents an enterprise to provide an automated response to the collaboration proposals. However, it is important that there is an explicit metapolicy guiding the agent to decide what cases can be considered as routine rejections or commitments. The cases can be detected, 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.

Both for the automated decision-making and for the support of human intervention, we propose to use an expert system to gather the relevant knowledge and to feed governing policies to the enterprise system, i.e., the relevant nonfunctional aspects of the collaboration and its contributing services.

The decisions to join a collaboration balance between the risk of failure or loss of assets as a consequence of participation, and the potential benefits of participation. That is, the expert system should compute a three-value outcome (agree, disagree, call for human intervention) on whether a service or a collaboration is dependable and beneficial for the enterprise in a given context and situation (Metso, 2007; Ruohomaa, 2007). By dependability we mean that the service fulfils its business purpose and the use of the service does not involve intolerable risk of monetary loss or reputation loss, for example caused by delivery failures or unacceptable delivery delays. Semantically, the decision to join the collaboration means two things. From the service providers viewpoint, an outsourcing relationship to the rest of the collaboration community becomes effective. From the collaboration point of view, an insourcing relationship takes effect. We consider insourcing and outsourcing to have technically identical “clauses”: three levels of interoperability and commitment to behaviour according to the eContract.

Computationally, the system computes values for risk and risk tolerance, both of which are vectors over a set of assets, such as monetary assets, reputation, fulfilment of purpose, and control of autonomy (Ruohomaa et al., 2006). For the risk values, the essential input comes from reputation information, i.e., first-hand experiences and positive and negative recommendations by members of earlier collaborations. For the risk tolerance, the essential input is from the perceived importance of the tasks or business network. The starting values for the importance and loss scenarios should be created by an extensive risk analysis and strategical business analysis.

When the risk vector is compared to the tolerance vector, a simple and safe decision is made as follows:

- agree, when no tolerance thresholds for acceptability are violated;
- disagree, when no tolerance vector values for disagreeability are violated; and
- propagate to human decision-making, when any tolerance vector value gets classified differently from the other vector values, all vector values fall between acceptability and disagreeability thresholds, or the metapolicy classifies the case as non-automatable.

When the request is forwarded to human consideration with all the relevant informa-

tion; the formulations and scope are yet to be detailed. The information should support the understanding the proposed collaboration, its business values and risks, trust on potential partners, privacy-preservation and so on. For the automated cases, the similar decision is based on a set of interoperability levels and nonfunctional aspects.

Besides the business network establishment phase, the same kind of decision-making takes place when entering significant tasks or business transactions within the collaboration.

Furthermore, during the operation of the business network, the monitors governing the business services can proactively or passively scan the messaging, reporting to enterprise level agents if the eContract is breached. The breaches can mean failing to fulfil an obligation, or failing to provide the agreed quality level of the service; more formally, failing to provide the level of dependability expected. At such situations, decisions are needed on whether the event is serious enough for terminating or leaving the business network. The same type of knowledge about the operational environment can be used, and again the expert system can make automated decisions or redirect the request for human intervention.

The concept of dependability, in terms of fulfilling the contracted aspects, can be concretised on two fields. There are general properties that can be set as service level expectations for any service, such as availability, timeliness, and privacy-preservation, or interaction relationship, such as non-repudiation and immutability. In addition, there are properties that are relevant for individual service types, each requiring a definition of value domain and metrics for defining the service levels relevant for the property. For example, reputation information (recommendations) can have a credibility property associated to it, determining how completely a recommendation from that source is assumed truthful. Another example is the traditional QoS levels with different metrics for data bandwidth and jitter in transfer.

The service level management lifecycle from property and metrics definition, to agreement establishment and control (Nurmela & Kutvonen, 2007) is present in the tool cycle illustrated earlier in Figure 1. The type definitions must include property frameworks; the property framework definitions form a common vocabulary for the domain and should represent standard definitions. Likewise, the type and business network model definitions can utilise policy framework definitions when expressing which part of the behaviour can be changed at operational time. The property and policy frameworks form a common vocabulary and thus create a strong tool on expressing the market needs and also in directing the markets. In addition, the set of property frameworks and policy frameworks is evolvable, because the definitions are stored in common B2B middleware repositories and thus made available to all enterprises.

In relation to other work (e.g., survey (Medjahed, Benatallah, Bouguettaya, Ngu, & Elmagarming, 2003), eNegotiation (Chiu, Cheung, Hung, Chiu, & Chung, 2004), OMNI (Vaquez-Salceda, Dignum, & Dignum, 2005), (Neisse, Pereira, Granville, Almeida, & Tarouco, 2004)) and outsourcing management systems, we emphasise a) use of predefined contract templates that capture not only business level or technical level issues, but both; b) running a multi-partner negotiation instead of bilateral negotiations; c) support of contract template evolution through the facilities for creating new business network models and policy variations; d) agility of business networks gained by operational time negotiations and renegotiations that is based on ontologies and abstract enough behaviour models created at design time; e) privacy of decision-making and using interoperability knowledge

effectively for it; and f) potential to use multiple negotiation protocols for different types of collaborations (auctioning systems, simple commitment protocols).

Verification and observation of interoperability

The Pilarcos middleware aims for maintaining correct collaborative behaviour in eCommunities, involving several aspects of interoperability requirements. The requirements cover technical, semantic, and pragmatic aspects, i.e., awareness of collaborative behaviour and policies. Traditional verification and static analysis methods are complemented by dynamic observation of behaviour conformance against the contracted BNM and policies.

The research and prototype building in the Pilarcos project focuses on interoperability and eCommunity management problems at the business service level, i.e. at the level of eCommunity, its participants, behaviour and life-cycle. As we presume that services are implemented or wrapped using Web Services technology, technical interoperability at the lower protocol levels is well provided by a service-oriented technical middleware layer.

Interoperability problems in software systems stem mainly from components' implicit and incorrect assumptions about behaviour of their surrounding environment (Garlan, Allen, & Ockerbloom, 1995). Every aspect of service and eCommunity functionality must be made explicit using unambiguous notations. Concepts of compatibility and substitutability are key issues in integration of autonomous services into communities; descriptions of services and communities must be founded on a formal basis.

When an eCommunity is established, we ensure *sufficient* conditions for interoperability of services during service discovery and population. Conditions for an interoperable eCommunity are fulfilled by three solutions. First, by the use of a verified BNM as a basic structuring rule for the eCommunity. The various business process models intertwined into the network model can be verified to be for example deadlock-free and complete by traditional protocol-verification tools. Second, by the use of constraint matching for accepting service offers to fulfil roles in the BNM. Previously verified compatibility and substitutability relationships between service types, provided by the type repositories, and validated conformance claims between service offers and corresponding services types are utilised in this process. Third, by the augmentation of the constraint matching process by the interference of further constraints arising from the selected offers for neighbour roles.

Behavioural interoperability is considered in the extent of verifying that service offers and role requirements for service behaviour match. By and large, this is accomplished by utilising the already verified correspondences between service offers and services types. However, roles may also impose additional constraints for the behavioural patterns of the contained services, such as obligations to perform specific transactions. Other relevant issues in role-related constraints cover interface syntax with behaviour descriptions, syntax of documents to be exchanged, semantic aspects of control and information flows, and non-functional aspects like trust and business policies that further restrict the behaviour. The role specific constraints are validated during the design of the BNM (verifying that the constraints imposed by the role do not conflict with the properties of the corresponding service types), and during the operation of the community (verifying that the service implementations actually conform to these constraints). These validation procedures can be implemented using model checking techniques.

To promote the evolution and flexible use of syntactic structures utilised by the services, we will adopt principles of by-structure matching instead of by-name matching for service interface comparisons (Ruokolainen, 2004). Using structural typing constructors for WSDL and XML-Schema definitions we can decide if two WSDL interface descriptions are structurally equal. This interface matching is done using an approach similar to (Palsberg & Zhao, 2001; Jha et al., 2002). Service selection and matching based on semantic concepts is not addressed in the present version of the Pilarcos platform but it will be implemented in future versions. Matching of semantic concepts shall be implemented using standard theories and tools, similarly to (Peer, 2002; Sriharee & Senivongse, 2003).

We do not even seek to completely prove that an eCommunity behaves correctly, as this would need verification of behaviours between every possible participant in an eCommunity during its establishment process. Even in theory, a complete pre-operational verification of an eCommunity behaviour would be impossible due to dynamic changes in the system, such as evolving business policies. Instead, service types are considered as contracts, and the subtyping of session types as proof of conformance. Inevitable behaviour and policy conflicts are observed and acted on during operational time by the monitoring system.

During runtime, however, participants of an eCommunity may behave incorrectly due to outdated service descriptions, changed business policies or technical problems. To overcome, or at least identify, interoperability problems during operation of communities, we have adopted an approach based on runtime monitoring of eCommunity contracts.

The monitoring system can be given a fairly free set of rules to monitor passing message traffic, and different informational and behavioural aspects are fairly straightforward to monitor (Kutvonen et al., 2005). The monitoring system reports detected situations (task started, completed, unacceptable traffic or lack of expected traffic). In monitoring, the challenges lie in the performance of the communication system, the design of monitoring rules, and decision engine.

Some breaches that can be detected by monitoring include a) messages from parties that are not partners in the eCommunity; b) transactions that are not acceptable in the current state of the eCommunity life-cycle or not fulfilling precedence requirements; c) information contents are not allowed to be exchanged (e.g., private documents, unknown structure); d) the expected flow of information is broken; and e) obligatory transactions are not performed.

Each administrative domain can have their own decision method on how critical a breach is considered. The eCommunity contract provides methods for network management agents (NMA) to invoke in case of breaches, either for information only, or for the removal of the partner in fault. The eCommunity contract carries these rules for deciding which recovery or sanction processes to use.

6. Related work

The Pilarcos architecture and services address the infrastructure requirements and solutions for bridging the gap between enterprise-level business considerations and the corresponding service management at the computing platforms. The global infrastructure services, transparently used by B2B middleware services at each enterprise, comprise partner service selection support, eContract management facilities, eCommunity life-cycle manage-

ment, breach detection by business-rule-aware monitors, and interoperability support facilities for technical, semantic, process-aware, and pragmatic aspects.

Traditionally, inter-enterprise collaboration has required integration of enterprise computing systems or applications. The topical integration techniques vary from new generation ERP systems, and process-orientation to distributed workflow management systems. A significant amount of research is currently focusing on virtual-enterprise approaches. Virtual enterprises are joint ventures of independent enterprises joining a shared collaboration process. In many projects (like PRODNET (Afsamanesh, Garita, Hertzberger, & Santos Silva, 1997), MASSYVE (Rabelo, Camarinha-Matos, & Vallejos, 2000), FETISH-ETF (Camarinha-Matos & Afsarmanesh, 2001) and WISE (Lazcano, Alonso, Schuldt, & Schuler, 2000; Alonso, 1999)) the support environment consists of a breeding environment and operational environment. The breeding environment provides facilities for negotiating and modelling the collaboration processes; the operational environment controls the enactment of the processes. Many of the virtual enterprise support environments use a unified architecture approach: there is a shared abstract model to which all enterprises have to adapt their local services.

In contrast to this, the approach in the Pilarcos architecture is federated: enterprises seek out partners that have services with which they are able to interoperate (within the strategically acceptable limits). A collaboration model (business network model, BNM) is used for explicitly expressing what kind of collaboration is wanted and comparison of BNMs is used as a semantic interoperability verification tool. Enactment of services and local business processes, either by applications or a local workflow management system are required features of the service management facilities of each local computing system. This design choice has been made in order to make the evolution of BNMs and business networks themselves more flexible. Changes in the model to follow require that the model is explicitly available at the operational time, and that there is a synchronisation and negotiation mechanism for partners to reach a safe point where new rules can be adopted.

Related to other toolsets for inter-enterprise collaboration management, the goal is rather different. For example, ATHENA (Berre et al., 2007) appears to provide a knowledge base for finding out good solution examples for repeating problems. An other example is ECOLEAD (Camarinha-Matos & Afsarmanesh, 2006; Rabelo, Gusmeroli, Arana, & Nagellen, 2006) where the breeding environment of virtual enterprises is mainly aiming at providing a strategic network facilities for automating the distributed business scenario, business process design and eContracting. The Pilarcos approach is not the only federated approach, however: (Montagut & Molva, 2005) and (Davidsson et al., 2006) present a rather similar management approach of a separate abstract business network that is populated with independent servers.

The fundamental difference between approaches is due to the changing goals of interoperability and with the changing maturity of B2B collaboration support. Each evolution phase has its characteristic challenges and solution architectures, as illustrated in Figure 6. The issues of interest focus on the second and third wave, while the first wave completes the picture by showing the traditional integration of application silos; typical solutions included data integration, presentation of joint portals, application integration, distributed workflow management and use of middleware (Linthicum, 2001).

The second wave introduces generated solutions that are based on shared models. The

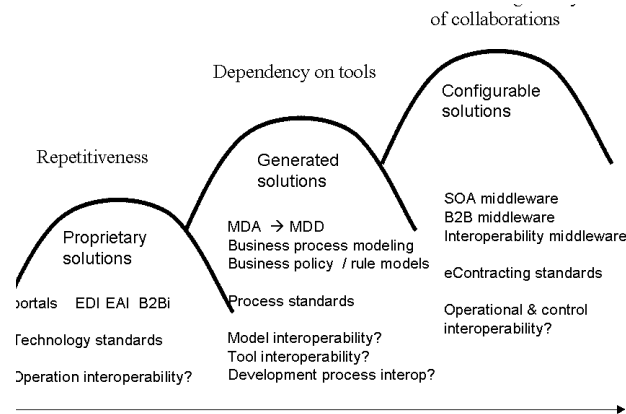


Figure 6. From manual integration to middleware supported interoperability.

emergence of service-oriented architectures (SOA) (Papazoglou & Georgakopoulos, 2003) to a wide audience has secured the use of the concepts of services, e-contracts, and metainformation for describing services. On this basis, the model-driven engineering approach (MDE) (Schmidt, 2006) provides tools for creating a unifying model for collaboration and generation of services and workflows that ensure interoperation between services provided by collaborating enterprises. Interoperability is ensured by joint design efforts and interoperation of design tools used at each collaborators system. The interoperability challenges focus on the production tools, their ability to exchange models and to generate logically similar implementation skeletons onto technically differing platforms.

The third wave illustrates the future enterprise computing systems that contain common, generic facilities for federated management of inter-enterprise collaborations. This type of solution well matches to the EU FP7 work program (European Commission, 2007), and challenges raised by the enterprise interoperability cluster (Li, Crave, Gilo, & van den Berg, 2007). Likewise, solutions suggested for digital business ecosystems have similar goals (Nachira, Dini, A.Nicolai, Louarn, & Léon, 2007). We call them B2B middleware (Kutvonen et al., 2006). The goal of this middleware layer is to provide a breeding environment for the establishment of new collaborations, and an operational time environment for controlling them. The automation of processes in these environments require, however, a well-formed set of knowledge about the interoperability features of services involved. Further, as the services and business processes of enterprises change, the interoperability knowledge must be dynamically increased. Thus, the interoperability knowledge is considered to be dynamically evolving, strictly typed and regulated by an evolving type discipline, heterogeneous in representation, and commonly available. Towards this goal, there is still lack of shared ontologies for eContracts, protocols for automated management of inter-enterprise collaborations, and control of the nonfunctional aspects of these collaborations.

Using the knowledge gathered into these repositories the B2B middleware agents can

collect suggestions for new collaborations, and furthermore, check and negotiate for a multi-party contracts so that all partners a) share the intent of using the same business network model; b) conform to the role requirements given to them; c) share NFA models and communication channel types with those partners they have direct communication with; and d) conform to the generic policies (business rules for example) defined for the collaboration.

The Pilarcos architecture has been developed in interaction with RM ODP (reference model of open distributed processing (IS10746, 1996)) and is founded on many of the principles also visible in current SOA (service oriented architecture (Papazoglou & Georgakopoulos, 2003)) trend. Shared foundations include the strong encapsulation of business services into autonomous units, introduction of meta-information services for service discovery and selection, and loose coupling of services for composites or collaborations.

The Pilarcos concept of eContracts ties together ICT-related viewpoints of ODP (Open Distributed Processing reference model (IS10746, 1996)), also ranging to some features of business aspects. The ODP-RM introduces information, computational, engineering and technical viewpoints. Each of these present interrelated but somewhat independent aspects of the collaboration features and their composition using more basic computing services. The Pilarcos contract structure captures these aspects in its BNMs, binding requirements, and behavioural and nonfunctional monitoring rules (Kutvonen, 2004a). The initial Pilarcos type repository was developed during the work on the ODP type repository function standard (IS14746, 1999), and OMG MOF specification (MOF, 2002). In other projects, like BCA (Milosevic, Linington, Gibson, Kulkarni, & Cole, 2004), contracts have legal and business level focus and detect contract breaches post-operatively (Quirchmayr, Milosevic, Tagg, Cole, & Kulkarni, 2002). Pilarcos aims for more real-time intervention.

The B2B middleware developed in the Pilarcos project series provides support for autonomously administered peer services that collaborate in a loosely coupled eCommunity. The eCommunity management by design excludes the need for distributed enactment services, but in contrast provides facilities for ensuring interoperability at the semantic and pragmatic level. In this respect the federated approach has a different focus from those in most other P2P community management systems, such as ADEPT (Reichert & Dadam, 1998) or METEOR (Aggarwal, Verma, Miller, & Milnor, 2004), and contract-driven integration approaches, such as ebXML (Kotok & Webber, 2001). Even most virtual enterprise support environments, such as CrossFlow (Grefen, Aberer, Hoffner, & Ludwig, 2000) and WISE (workflow-based Internet services) (Lazcano et al., 2000), rely on models for distributed business process enactment. However, the Pilarcos approach leaves enactment as a local business processing task, concentrating on interoperability monitoring.

In the Pilarcos middleware, the eCommunity life-cycle is built to be collaboration-process-aware. The architecture model acts on two abstraction layers, the upper layer involved with the abstract, external business process describing the collaboration requirements; the lower layer comprising actual services bound to the eCommunity dynamically. In this kind of environment, static verification of models and interoperability cannot be complete. In the B2B middleware provided by the Pilarcos project, we find it necessary to develop control environments for monitoring and reflectively restructuring the operational eCommunities, besides a breeding environment. The goals are similar to other projects, but the solution methods differ. While ADEPT supports direct modification of the workflow control structures, Pilarcos uses negotiated policy-values to choose between predefined behaviour

alternatives. The Pilarcos solution even requires that well-formed contracts include suitable recovery processes that involve whole communities. In contrast to METEOR-S, the Pilarcos platform has no central tool for making the whole of an interoperability analysis, but partial static verification is done at the meta-data repositories, and monitoring is used to detect further problems.

The B2B middleware is in some extent comparable to agent-based approaches, such as MASSYVE (Rabelo et al., 2000). The main difference seems to be the separation of business services and B2B middleware services from each other. The Pilarcos middleware agents do not provide workflow execution, but expect local application management to play that part. In contrast to (Daskalopulu, Dimitrakos, & Maibaum, 2002), the middleware agents are responsible for semantic verification and failure resolution, and use separate monitors to help and report.

7. Conclusion

The Pilarcos architecture provides a B2B middleware layer that supports management of business networks. The management facilities are based on a shared vision of meta-information captured into a eContract. Changes in the contract are locally reflected to the enterprise computing system; and correspondingly, relevant progress and breach reports are delivered to partners through the eContract.

The architecture follows a federated approach: participating services are independent and pre-existing, and the collaborative behaviour model is used only for watching conformance. Enforcement of the contract is reached through the independent monitoring facilities at each participant. The monitors basically react to events that should not take place at that service or resource interface. These self-protective reactions are then used as triggers for corrective actions for the benefit of the whole business network.

The Pilarcos approach supports autonomous services to form federated communities. A federated approach means that there is no overarching shared collaboration model from which the services would be derived. Instead, the services stand on their own and interoperability from the collaboration process, semantic and technical view must be maintained explicitly by B2B middleware. From the BNM, it would be possible to use the popular model-driven approach and generate applications, but those are not resistant to evolution needs. This is further discussed in (Kutvonen, 2004b, 2004a).

In this kind of environment, several strategical, process-related and technological needs can be attended by the business network management:

- Formation of new business networks that provide added value services for clients.
- Joining multiple networks at the same time without unnecessary restrictions on technologies or operational policies.
- Taking up new business processes and services with relatively low cost.
- Moving existing business networks to new phases of life-cycle so that new collaboration forms can be used.
- Monitoring the progress and correctness of the collaborative processes.
- Automating some collaboration establishment and correction events.
- Protecting local services and computing solutions from the changes and failures of the collaboration partner services and solutions.

The provision of the Pilarcos architecture requires further development of business process modelling techniques. The collaboration of business processes or workflows should be modelled without unnecessary revealing of local processing steps. Instead, only the collaborative part (external view) should be agreed on and monitored. Work has already been started by the component-driven approach on splitting workflows into Web Services. The structural needs of business process models are also widened by the requirements of incorporating reusable sanctioning, recovery, and compensation processes into eCommunity contracts. Furthermore, shared ontologies and repositories for business process models should be made available. Such facilities would improve the potential for reaching interoperability in an environment where service components are truly developed independently from each other. More fundamentally, ontologies and repositories would create a facility for checking semantic similarity of a business process model as part of the interoperability tests during eCommunity establishment.

The federated approach has been criticised for the lack of advice for service elements to be developed. However, making existing business network models globally available and thus exposing repeating patterns of roles — i.e., expected local business processes — gives the required guidance. Such publishing has already taken place with RosettaNet etc; our solution is to provide a repository for external process descriptions that can be augmented on demand, and that will provide an element of evolution support. These model definitions can be added to the repositories at will, without interfering with already operational communities. Existing models can be frozen so that new communities are no more formed using them, but are not actually removed automatically. The verification and matching hierarchies within the repositories may depend on them, and of course, operational communities may do references.

Another criticism frequently arising is the performance penalty of the eCommunity interoperability checking. From our earlier prototype on the populator process, we can judge that the cost of the process and its scalability are acceptable (Kutvonen & Metso, 2005). The scalability of the open-ended search for potential partners from service offer repositories indicates a large search space; the matching process is further complicated by the interdependencies between selected partners in terms of available communication solutions and policies. The populator algorithms address the potentially exponential growth of the search space by limiting the resources used for the search, at the cost of the completeness of the results.

Specific features of the Pilarcos breeding environment include the level of automation expected, the relaxed matching of service types aimed for, and the use of explicit business network model repositories.

The level of automation in eContracting has to be considered carefully. Enterprises are generally not ready to allow automated agents to take business-level decisions. Therefore, the automatically acceptable commitments have to be guarded by enterprise policies, and to be directed towards routine decisions. The main impact on the Pilarcos facilities is in providing control over technological and evolution-involved problems, not in the aggressive enhancement to new business partners. The major development on the architecture, however, is trust management, on which we have started a separate development project (Ruohomaa et al., 2006).

The federated type repository service is an essential element of a B2B middleware that supports the establishment of new business networks, or in a more simple case, connection

between independently administered clients and servers. The role of the type repository is to provide a trustworthy source of service type information, and furthermore, provide transformation services for communication between almost similar interfaces. The service types can thus be matched with each other in a more relaxed way, only limited with the interoperability requirement. As an enhancement, the cost of connection can be added to direct users to choose “native” types instead of transformed connections. The service type matching approach supports evolution of services in a heterogeneous environment, where independent actors create new items, and where market forces has an effect on the usability of items, in addition to the verifiable correctness properties. Furthermore, the approach gives a natural tool for managing one type of transformation components needed in the current component-based, model-driven networking environment.

The use of explicit business network model repository is an ontology-defining tool that allows dynamic development and quick publication of new collaboration models. This is one of the key elements in the trial of developing evolution support for dynamic, inter-enterprise networks.

References

- Afsamanesh, H., Garita, C., Hertzberger, B., & Santos Silva, V. (1997). Management of distributed information in virtual enterprises - the PRODNET approach. In *ICE'97 - International Conference on Concurrent Enterprising*.
- Aggarwal, R., Verma, K., Miller, J., & Milnor, W. (2004, September). Constraint Driven Web Service Composition in METEOR-S. In *IEEE International Conference on Services Computing (SCC'04)* (pp. 23–30). Los Alamitos, CA, USA: IEEE Society Press.
- Allen, R., & Garlan, D. (1994). Formalizing architectural connection. In *ICSE '94: Proceedings of the 16th international conference on Software engineering* (pp. 71–80). Los Alamitos, CA, USA: IEEE Computer Society Press.
- Alonso, G. (1999). WISE: Business to Business e-Commerce. In *9th workshop on research issues on data engineering (RIDE-VE'99)*. Sydney.
- Belwood, T., & et al. (2004). *UDDI version 3.0*. (Retrieved September 1st, 2005, from http://uddi.org/pubs/uddi_v3.htm)
- Berre, A.-J., Elvesaeter, B., Fgay, N., Guglielmina, C., Johnsen, S., Karlse, D., et al. (2007, March 28–30). The ATHENA Interoperability Framework. In *Enterprise Interoperability II*. Funchal, Portugal: Springer.
- Camarinha-Matos, L. M., & Afsarmanesh, H. (2001, November). Service federation in virtual organisations. In *Prolamat'01*. Budapest, Hungary.
- Camarinha-Matos, L. M., & Afsarmanesh, H. (2006, September). Modeling framework for collaborative networked organizations. In *Network-Centric Collaboration and Supporting Frameworks. Seventh IFIP Working Conference on Virtual Enterprises* (pp. 3–14). Springer.
- Chiu, D. K. W., Cheung, S. C., Hung, P. C. K., Chiu, S. Y. Y., & Chung, A. K. K. (2004). Developing e-Negotiation support with a meta-modeling approach in a web services environment. *Decision Support Systems*, 40, 51–69. (ISSN:0167-9236, Special issue: Web services and process management)
- Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001, March). *Web Services Description Language (WSDL) 1.1*. (Retrieved September 1st, 2005, from <http://www.w3c.org/TR/wsdl>)

- Damianou, N., Dulay, N., Lupu, E., & Sloman, M. (2001, January). The Ponder policy specification language. In *Workshop on policies for distributed systems and networks (Policy2001)* (Vol. 1995). HP Labs Bristol: Springer-Verlag.
- Daskalopulu, A., Dimitrakos, T., & Maibaum, T. (2002). Evidence-Based Electronic Contract Performance Monitoring. *The INFORMS Journal of Group Decision and Negotiation*. (Special Issue on Formal Modelling in E-Commerce.)
- Davidsson, P., Hederstierna, A., Jacobsson, A., Persson, J. A., Carlsson, B., Johansson, S. J., et al. (2006). The concept and technology of plug and play business. In *ICEIS 2006 - proceedings of the eighth international conference on enterprise information systems: Databases and information systems integration* (p. 213-217). Paphos, Cyprus.
- Di Cosmo, R., Pottier, F., & Rémy, D. (2005, April). Subtyping Recursive Types modulo Associative Commutative Products. In P. Urzyczyn (Ed.), *7th International Conference on Typed Lambda Calculi and Applications (TLCA 2005)* (Vol. 3461, pp. 179–193). Berlin, Heidelberg, Germany: Springer-Verlag.
- European Commission. (2007, June). *EC FP7 ICT Work Programme* (Tech. Rep.). EC.
- Garlan, D., Allen, R., & Ockerbloom, J. (1995). Architectural mismatch or why it's hard to build systems out of existing parts. In *ICSE '95: Proceedings of the 17th international conference on Software engineering* (pp. 179–185). New York, NY, USA: ACM Press.
- Gay, S. J., & Hole, M. (1999). Types and Subtypes for Client-Server Interactions. In *ESOP '99: Proceedings of the 8th European Symposium on Programming Languages and Systems* (Vol. 1576, pp. 74–90). London, UK: Springer-Verlag.
- Goodchild, A., Herring, C., & Milosevic, Z. (2000, June). Business contracts for B2B. In *Proceedings of the CAISE00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing ISDO'00*. Stockholm, Sweden: CEUR-WS.org.
- Grefen, P., Aberer, K., Hoffner, Y., & Ludwig, H. (2000). CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. *International Journal of Computer Systems Sciences and Engineering*, 15 (5), 277-290.
- Grosz, B. N., & Poon, T. (2003). SweetDeal: Representing agent contracts with exceptions using XML rules, ontologies and process descriptions. In *Proceedings of the international conference on the World Wide Web*. Budapest, Hungary: WWW2003.
- Information Technology – Open Distributed Processing – Reference Model – Enterprise Language*. (2003). (IS15414)
- Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. ODP Trading Function*. (1995). (IS13235)
- Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing*. (1996). (IS10746)
- Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing. ODP Type repository function*. (1999). (IS14746)
- Jha, S., Palsberg, J., & Zhao, T. (2002). Efficient type matching. In *FoSSaCS '02: Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures* (pp. 187–204). London, UK: Springer-Verlag.
- Kotok, A., & Webber, D. R. R. (2001). *ebXML: The New Global Standard for Doing Business Over the Internet*. Boston, USA: New Riders.

- Kutvonen, L. (1998). *Trading services in open distributed environments*. Department of Computer Science, University of Helsinki. (PhD thesis. A-1998-2)
- Kutvonen, L. (2002). Automated management of inter-organisational applications. In *Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC 2002)* (p. 27-38). Lausanne, Switzerland: IEEE.
- Kutvonen, L. (2004a). Challenges for ODP-based infrastructure for managing dynamic B2B networks. In A. Vallecillo, P. Linington, & B. Wood (Eds.), *Workshop on odp for enterprise computing (wodpec 2004)* (pp. 57–64). Monterey, California.
- Kutvonen, L. (2004b, September). Relating MDA and inter-enterprise collaboration management. In *Second European Workshop on Model Driven Architecture (MDA), EWMDA-2* (pp. 84–88). (Published as technical report No. 17-04 in University of Kent.)
- Kutvonen, L., & Metso, J. (2005, September). Services, contracts, policies and eCommunities – Relationship to ODP framework. In P. Linington, A. Tanaka, S. Tyndale-Biscoe, & A. Vallecillo (Eds.), *Workshop on ODP for Enterprise Computing (WODPEC 2005)* (pp. 62–69). Enschede, The Netherlands.
- Kutvonen, L., Metso, J., & Ruohomaa, S. (2006, October). From trading to eCommunity population: Responding to social and contractual challenges. In *Proceedings of the 10th IEEE International edoc conference (edoc 2006)* (pp. 199–210). Hong Kong: IEEE.
- Kutvonen, L., Metso, J., & Ruohomaa, S. (2007, July). From trading to eCommunity management: Responding to social and contractual challenges. *Information Systems Frontiers (ISF) - Special Issue on Enterprise Services Computing: Evolution and Challenges*, 9(2–3), 181–194.
- Kutvonen, L., Metso, J., & Ruokolainen, T. (2005, November). Inter-enterprise collaboration management in dynamic business networks. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE* (Vol. 3760). Agia Napa, Cyprus: Springer-Verlag.
- Kutvonen, L., Ruokolainen, T., & Metso, J. (2007, January). Interoperability middleware for federated business services in web-Pilarcos. *International Journal of Enterprise Information Systems, Special issue on Interoperability of Enterprise Systems and Applications*, 3(1), 1–21.
- Lazcano, A., Alonso, G., Schuldt, H., & Schuler, C. (2000). The WISE approach to Electronic Commerce. *Computer Systems Science and Engineering*, 15(5), 345–357.
- Li, M.-S., Crave, S., Gilo, A., & van den Berg, R. (2007). *Value proposition for enterprise interoperability*. EC.
- Linthicum, D. S. (2001). *B2B Application Integration - eBusiness-Enable Your Enterprise*. Addison-Wesley.
- Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A. H. H., & Elmagarming, A. K. (2003). Business-to-business interactions: issues and enabling technologies. *The VLDB Journal*(12), 59-85.
- Meta Object Facility (MOF) Specification — Version 1.4*. (2002, April). (Retrieved September 1st, 2005, from <http://www.omg.org/docs/formal/02-04-03.pdf>)
- Metso, J. (2007, April 26–27). Pragmatic aspects in computer-supported negotiations of virtual enterprise contracts. In *Web proceedings of the I-ESA '07 Doctoral symposium*. Funchal, Portugal. (To appear.)
- Metso, J., & Kutvonen, L. (2005, September). Managing Virtual Organizations with Contracts. In *Workshop on Contract Architectures and Languages (CoALa2005)*. Enschede, The Netherlands.

- Milosevic, Z., Jøsang, A., Patton, M., & Dimitrakos, T. (2002). Discretionary enforcement of electronic contracts. In *6th enterprise distributed object computing conference (EDOC 2002)*. Enschede, The Netherlands: IEEE Computer Society.
- Milosevic, Z., Linington, P. F., Gibson, S., Kulkarni, S., & Cole, J. B. (2004). Inter-Organisational Collaborations Supported by E-Contracts. In W. Lamersdorf, V. Tschammer, & S. Amarger (Eds.), *Building The E-Service Society: E-Commerce, E-Business, and E-Government - IFIP 18th World Computer Congress TC6/TC8/TC11 4th International Conference on E-Commerce, E-Business, E-Government (I3E 2004)* (p. 413-429). Kluwer.
- Montagut, F., & Molva, R. (2005). Enabling pervasive execution of workflows. In *International conference on collaborative computing: Networking, applications and worksharing*. San Jose, CA, USA: IEEE.
- Nachira, F., Dini, P., A.Nicolai, Louarn, M., & Léon, L. (2007). *Digital business ecosystems*. European Commission.
- Nardi, D., & Brachman, R. (2002). The description logic handbook. In F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, & P. Pate-Schneider (Eds.), (pp. 5–44). Cambridge CB2 2RU, UK: Cambridge University Press.
- Neal, S., Cole, J. B., Linington, P. F., Milosevic, Z., Gibson, S., & Kulkarni, S. (2003). Identifying requirements for Business Contract Language: a Monitoring Perspective. In *7th International Enterprise Distributed Object Computing Conference (EDOC 2003)* (pp. 50–61). Los Alamitos, CA, USA: IEEE.
- Neisse, R., Pereira, E. D. V., Granville, L. Z., Almeida, M. J. B., & Tarouco, L. M. R. (2004). A hierarchical policy-based architecture for integrated management of grids and networks. In *Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)* (p. 103-106). New York, USA.
- Norta, A. (2007). *Exploring dynamic inter-organisational business process collaboration*. Technicshe Universiteit Eindhoven, Department of Technology Management.
- Nurmela, T., & Kutvonen, L. (2007, June). Service level agreement management in federated virtual organizations. In *Distributed applications and interoperable systems (DAIS2007)*. Paphos, Cyprus: Springer-Verlag.
- OWL Web Ontology Language Guide*. (2004, February). (W3C Recommendation 10 February 2004, Retrieved September 1st, 2005, from <http://www.w3.org/TR/owl-guide/>)
- Palsberg, J., & Zhao, T. (2001). Efficient and flexible matching of recursive types. *Information and Computation*, 171(2), 364–387.
- Papazoglou, M. P., & Georgakopoulos, D. (2003, October). Service oriented computing. *Communications of the ACM*, 46(10).
- Peer, J. (2002). Bringing Together Semantic Web and Web Services. In *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web* (pp. 279–291). London, UK: Springer-Verlag.
- Quirchmayr, G., Milosevic, Z., Tagg, R., Cole, J., & Kulkarni, S. (2002). Establishment of Virtual Enterprise Contracts. In *Database and Expert Systems Applications : 13th International Conference* (Vol. 2453, pp. 236–248). London, UK: Springer-Verlag.
- Rabelo, R. J., Camarinha-Matos, L. M., & Vallejos, R. V. (2000). Agent-based brokerage for virtual enterprise creation in the moulds industry. In *Proceedings of the IFIP TC5/WG5.3 Second IFIP Working Conference on Infrastructures for Virtual Organizations: Managing Cooperation in Virtual Organizations and Electronic Business towards Smart Organizations* (pp. 281–290). Deventer, The Netherlands: Kluwer, B.V.

- Rabelo, R. J., Gusmeroli, S., Arana, C., & Nagellen, T. (2006). The ECOLEAD ICT infrastructure for collaborative networked organizations. In *Network-centric collaboration and supporting frameworks* (Vol. 224, p. 451-460). Springer-Verlag.
- RDF Vocabulary Description Language 1.0: RDF Schema*. (2004, February). (W3C Recommendation 10 February 2004. Retrieved September 1st, 2005, from <http://www.w3.org/TR/rdf-schema/>)
- Reichert, M., & Dadam, P. (1998). ADEPTflex – Supporting Dynamic Changes of Workflow Without Losing Control. *Journal of Intelligent Information Systems*, 10(2), 93-129. (Special Issue on Workflow Management)
- RosettaNet Consortium. (2004). *RosettaNet Implementation Framework: Core Specification V02.00.00*. (Retrieved September 1st, 2005, from <http://www.rosettanet.org/>)
- Ruohomaa, S. (2007, April 26–27). Trust management for inter-enterprise collaborations. In *Web proceedings of the I-ESA '07 Doctoral symposium*. Funchal, Portugal. (To appear.)
- Ruohomaa, S., & Kutvonen, L. (2008, March). Making multi-dimensional trust decisions on inter-enterprise collaborations. In *Proceedings of the Third international conference on availability, reliability and security (ARES 2008)*. Barcelona, Spain: IEEE.
- Ruohomaa, S., Viljanen, L., & Kutvonen, L. (2006, March). Guarding enterprise collaborations with trust decisions—the TuBE approach. In *Interoperability for Enterprise Software and Applications. Proceedings of the Workshops and the Doctoral Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006* (pp. 237–248). Bordeaux, France: ISTE Ltd.
- Ruokolainen, T. (2004). *Component interoperability*. University of Helsinki, Department of Computer Science. (MSc thesis C-2004-42. In Finnish.)
- Ruokolainen, T., & Kutvonen, L. (2007, April). Service Typing in Collaborative Systems. In G. Doumeingts, J. Müller, G. Morel, & B. Vallespir (Eds.), *Enterprise Interoperability: New Challenges and Approaches* (pp. 343–354). Springer.
- Ruokolainen, T., Metso, J., & Kutvonen, L. (2007, March). Ontology for federated management of business networks. In L. Kutvonen, P. Linington, J.-H. Morin, & S. Ruohomaa (Eds.), *Pre-proceedings of IS-TSPQ 2007 — The 2nd international workshop on Interoperability solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems* (pp. 41–54). University of Helsinki, Department of Computer Science Publications Series B, Report B-2007-3.
- Schmidt, D. (2006, February). Model-driven engineering. *IEEE Computer*, 39(2), 25–31.
- Sriharee, N., & Senivongse, T. (2003, November). Discovering Web Services Using Behavioural Constraints and Ontology. In J.-B. Stefani, I. Demeure, & D. Hagimont (Eds.), *Distributed Applications and Interoperable Systems: 4th IFIP WG6.1 International Conference (DAIS 2003)* (pp. 248–259). Springer-Verlag.
- Takeuchi, K., Honda, K., & Kubo, M. (1994). An Interaction-based Language and its Typing System. In *PARLE '94: Proceedings of the 6th International PARLE Conference on Parallel Architectures and Languages Europe* (pp. 398–413). London, UK: Springer-Verlag.
- Vallecillo, A., Vasconcelos, V. T., & Ravara, A. (2003). Typing the behavior of objects and components using session types. *Electronic Notes in Theoretical Computer Science*, 68(3). (Presented at FOCLASA'02)
- van der Aalst, W., ter Hofstede, A., & Weske, M. (2003). Business process management: A survey. In *Proceedings of the First International Conference on Business Process Management*. Eindhoven, The Netherlands: Springer-Verlag.

- van der Aalst, W. M. P., Verbeek, H. M. W., & Kumar, A. (2001). Xel/woflan: Verification of an xml/petri-net based language for inter-organisational workflows. In *Proceedings of the 6th Informis Conference on Information Systems and Technology (CIST-2001)* (p. 30-45).
- Vaquez-Salceda, J., Dignum, V., & Dignum, F. (2005). Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11, 307–360.