International Journal of Enterprise Information Systems
Volume, number, pages

Interoperability middleware for federated business services in web-Pilarcos

Lea Kutvonen, Toni Ruokolainen, Janne Metso
Department of Computer Science, University of Helsinki, Finland
{Lea.Kutvonen|Toni.Ruokolainen|Janne.Metso}@cs.helsinki.fi

Participation into electronic business networks has become necessary for the success of enterprises. The web-Pilarcos B2B middleware is designed for lowering the cost of collaboration establishment and to facilitate management and maintenance of networks. The web-Pilarcos architecture and middleware addresses the interoperability of autonomous business services in inter-organisational context. The approach is a federated one: All business services are developed independently, and the B2B middleware services are used to ensure that technical, semantic, and pragmatic interoperability is maintained in the business network. In the design, attention has been given to the dynamic aspects and evolution of the network. This paper discusses the concepts provided for application and business network creators, and the supporting middleware-level knowledge repositories for interoperability support.

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 much integrates with RM-ODP standards work, and recently has found an interesting context in the INTEROP NoE collaboration.

Interoperability middleware for federated business services in web-Pilarcos

Participation into electronic business networks has become necessary for the success of enterprises. The web-Pilarcos B2B middleware is designed for lowering the cost of collaboration establishment and to facilitate management and maintenance of networks. The web-Pilarcos architecture and middleware addresses the interoperability of autonomous business services in inter-organisational context. The approach is a federated one: All business services are developed independently, and the B2B middleware services are used to ensure that technical, semantic, and pragmatic interoperability is maintained in the business network. In the design, attention has been given to the dynamic aspects and evolution of the network. This paper discusses the concepts provided for application and business network creators, and the supporting middleware-level knowledge repositories for interoperability support.

Introduction

The globalization of business and commerce makes enterprises increasingly dependent on their cooperation 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 timely to the requirements rising from the changing co-operation networks and their communications needs.

From the computing infrastructure side, the enterprise needs can be addressed by an architecture where business level services, B2B middleware, and abstract communication services are clearly separated from each other, and the relationships between collaboration life cycle, B2B middleware, and software engineering tools are changed from the traditional approach. By B2B middleware we mean general infrastructure services that provides concepts and operations for forming electronic business networks, eCommunities, and managing their life cycle.

The B2B middleware concepts and operations should be such, that strategical, process-related and technological needs of electronic business network management is filled. Such needs we believe to include the following.

- Formation of new business networks that provide added value services for clients.
- Joining to 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.

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. 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, the support environment consists of a breeding environment and operational environment. The breeding environment provides facilities for negotiating and modeling 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 web-Pilarcos project 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

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 much integrates with RM-ODP standards work, and recently has found an interesting context in the INTEROP NoE collaboration.

of BNMs is used as a semantic interoperability verification tool. Enactment of services and local business processes, either by applications or 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 synchronization and negotiation mechanism for partners to reach a safe point where new rules can be adopted.

The web-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 of partner service selection support, eContract management facilities, eCommunity life cycle management, breach detection by business-rule-aware monitors, and interoperability support facilities for technical, semantic, process-aware, and pragmatic aspects.

This paper concentrates on the breeding environment facilities for checking interoperability of business services for the purposes of forming an eCommunity, and the connection between business-level aspects and enterprises' computational services. Section 1 discusses interoperability challenges in the context of eCommunity management, and Section 2 briefly describes the web-Pilarcos B2B middleware services and repositories. Section 3 addresses the meta-information and corresponding public repositories presented by the web-Pilarcos middleware.

eCommunity management and interoperability

The web-Pilarcos architecture proposes a model of inter-enterprise collaborations as eCommunities comprising of independently developed business services. Business service denotes a set of functionalities provided by an enterprise to its clientele and cooperators, and that 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 realized by software, the service is also defined by the computing, information representation, and communication facilities used and required.

An enterprise that is able to run a computational service, i.e., a 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. In addition, we expect the service offer to be considered as a binding offer 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. The service offers and the structuring effect of service types is further discussed in Section . Essential for the offer structure is that the contract terms relevant for the business area become represented in a way that allows comparison.

The web-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.

We use the service publication process to start a contracting process between the business services. The business service basic properties 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 looses 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 organizational issue not fully addressable by computing solutions. Computationally, it is possible to control that exporters of service offers are authorized by their organizations for making external commitments, and that there is technical facilities to follow the thread of delegations and negotiations for determining the responsible party for each commitment.

The strategical requirements of an business network member towards the collaboration are expressed as a meta-level model that defines a set of external business processes. 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 fulfill it, and conformance rules determine limits for acceptable behaviour during the eCommunity operation. The explicit use of such model allows comparison and matching of strategical and pragmatical goals of members in the network.

Interoperability is a functionality provided by the middleware services, a transparent aspect for application-level services. Interoperability checking takes place when establishing an 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. Collaboration and eCommunity membership aspects together with pragmatic process-awareness, however, require application-level concepts and services.

Monitoring interoperability during eCommunity lifetime requires sensors and guards at each communication channel end. We assume an abstract communication infrastructure with selectable transparencies and support for non-functional aspects. From the service specifications it is known what traffic should be seen and in which order; in principle the rules can be extended to view the acceptability of contents structures and making trust related decisions.

The inter-enterprise collaboration management concepts supported by the web-Pilarcos architecture include those of

- an eCommunity that represents a specific collaboration, its operation, agreements and state; the eCommunities carry identities and are managed according to their eCommunity contract information;
- services that are provided by enterprises, used as members in eCommunities, and are made publicly available by exporting service offers;
- 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
- A set of repositories for storage of meta-models for communities, ontologies of service types, and services.

The eCommunity life cycle is mainly controlled in a eCommunity contract. The contract comprises of the BNM (to define the network structure), information about the member services at each role, some overview state information about the progress of the external business processes, and methods for changing the contract itself.

The eCommunity contract captures shared meta-information about the collaboration; reflective methods are used to keep the real system at each involved computing site correspondent with the 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 homogenized 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.

Figure 1 illustrates the phases at which additional contractual information is associated to the business service to reflect the computational service and its business aspects. In the offer publication, discovery, and later engagement activities, the service offers represent enterprise commitments for providing a business service with indicated properties, using the terms and conditions negotiated in the engagement phase. 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 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 that is capable of more varied behaviours.

The web-Pilarcos B2B middleware architecture

The B2B-middleware platform provides

- advanced service discovery based on improved services typing and constraint based selection,
- contract based management of collaboration between autonomous services, and
- proactive local monitoring of contract conformance.

Furthermore, repositories with relationship to collaboration modeling, software engineering, and deployment present the knowledge base required for B2B interoperability support.

The B2B middleware elements are illustrated in Figure 2, showing the populator as the active agent of the breeding environment, and network management agents (NMAs) and eContracts as the major player of the operational environment. For the support of the populator, the BNM design process involves the introduction and verification of new models to be stored into the repositories. Implementation of new services or 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. These processes feed in meta-level knowledge of potential participants in communities to be formed. The feeding processes are independent of each other, even withdrawing or deprecating information may take place. Each site or administrative domain, representing an autonomous ICT system, is expected to run a business process management agent. By autonomy we mean the potential for control over the private computing systems, and moreover on strategic business processes and policies. Breeding environment services like populators and type repositories are not required from all sites, but can be provided as infrastructure services as a business on its own right.

The *Populator* uses a given BNM for ensuring the pragmatic interoperability of partners to a eCommunity; it also uses a set of compulsory aspects in service offers to determine service types, communication channel requirements, and non-functional aspects to be agreed on 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 service offer repository. As there is 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 done in cooperation with *Network Management Agent (NMA)* and the *Contract* object. The agents are responsible for managing the inter-organizational coordination and management protocols. The contract object is responsible for making decisions regarding the eCommunity it represents. At each administrative domain, there is a NMA agent acting as a representative be-

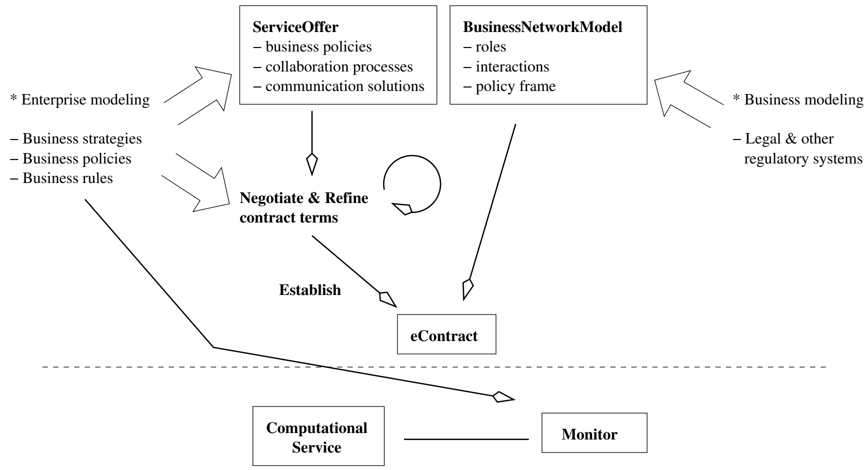

Figure 1. Overview of the flow of business-level information to the computational level.

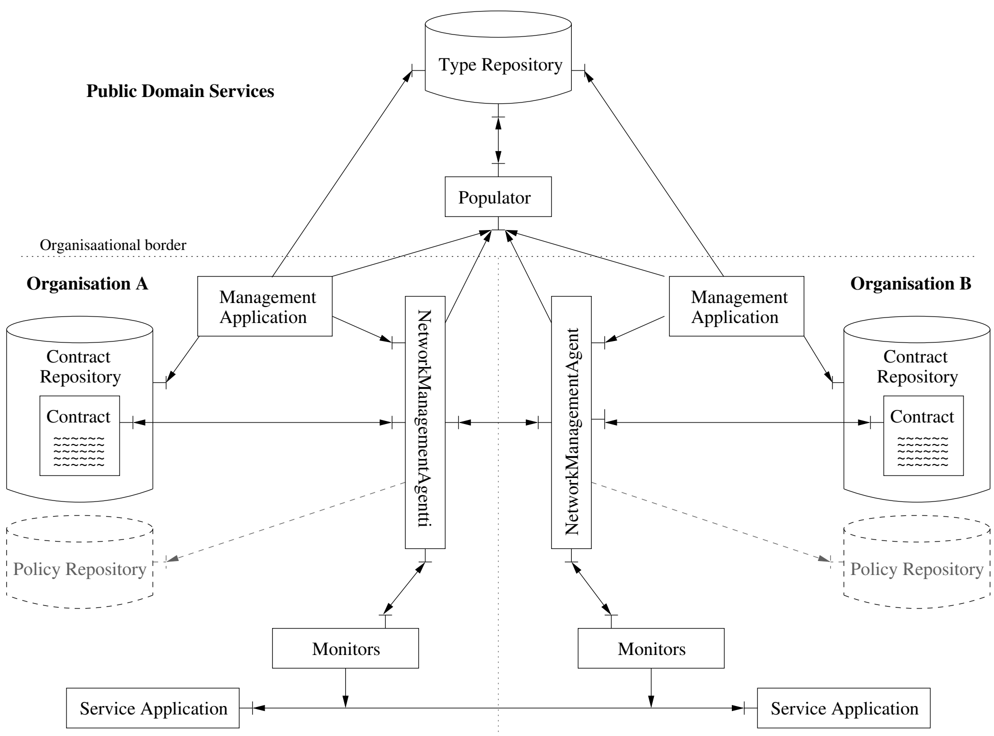

Figure 2. Service agents of the operational environment.

tween the eCommunity and the local service-providing system. For local administrators the agents provide an management interface for communities. Between themselves the agents have a protocol for notifications of task completions and contract breaches, and negotiation and commitment protocols for joint contract changes. Each local agent receives notifications of contract breaches and task completions from local monitors and propagates this information forward to other agents as needed. Local agent also feeds monitoring instructions to the monitor.

The *eCommunity contract* itself is a key element in the architecture, because it makes available at operational time aspects from different levels/viewpoints of the business network. Community contract describes technical, semantical, (external business) process-related, and pragmatical 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 BNM used for the eCommunity: each role is supplemented with information from participants service offer, each binding with connector parametrization information. Semantical aspects cover information representation formats in messages exchanged. The pragmatic aspects covered include functional description of business processes, policies constraining roles, and non-functional aspects. The non-functional aspects govern features like trust, security, QoS that are traditionally considered as additional platform level service solutions required. In addition, non-functional 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 captured in Figure 3.

- *reference to the business network model;*
- *information about the epoch in which the network is;*
- *process for changing epoch;*
- *for each role*
- *assignment rules that specify the requirements on*
- *service type;*
- *nonfunctional aspects;*
- *restrictions on identity, participation on other business networks, etc;*
- *conformance rules that are used for determining conformance to the role which the assigned component is in the role; similar as above;*
- *for each interaction relationship between roles*
- *channel requirements*
- *locations of the channel endpoints*
- *QoS agreement; security agreement*
- *information presentation formats*
- *for each policy that governs the choices between alternative behaviour patterns in the business network model*
- *acceptable values or value ranges;*
- *references to alternative breach recovery processes;*
- *objective of the business network as business rules*

Figure 3. Information contents of the eContract.

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 between themselves.

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 a behaviour automaton to follow, based on the service choreographies described for the corresponding role. Monitoring reports can be acted on in various ways, scaling from post-operational auditing to proactive prevention of unwanted events. In web-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 web-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 application domain, those rules and process definitions are compulsory parts of BNMs.

The most important task of the monitors is to 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.

Interoperability knowledge in the global middleware

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 type of load, the good distribution styles differ (Kutvonen, 1998).

Service types and BNMs have separate life cycles as this provides isolation layers that keep local changes from involving the whole eCommunity and minimizes the effects of BNM enhancements to local services. Furthermore, each model requires only a reasonably narrow expertise to create. In addition to direct relationships between models, the repositories store transformation rules and components for improved transformer/interceptor re-usability (Kutvonen, 1998).

The service elements of the web-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;
- 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 form a quite separate profession. 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. The web-Pilarcos type repository design was initially born during the evolution of ODP type repository and OMG MOF specifications (IS14746, 1999; MOF, 2002). 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 interoperation 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 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 web-Pilarcos is described using a WSDL description without technical binding information (see (Christensen, Curbera, Meredith, & Weerawarana, 2001)). Each service supports only one kind of behaviour; different behaviour implies different service type. 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 web-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 specialized 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 web-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. Standardization of a new service type is however not necessary because the applicability and adoptance 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 judgment is not fully automated and cannot be made (due 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 interceptor creation is external to the type repository. The service types can thus be matched with each other in a more relaxed way, only limited with 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 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 realizations of abstract types,
- checking whether two type realizations are conformant and interchangeable,
- retrieving subtypes or supertypes of a type realization,
- retrieving templates for a given abstract type,
- translating one type realization to another,
- retrieving names for abstract types and type realizations in other type domains.

The *service offer repository* refers to services (like UDDI (Belwood & et al., 2004) and ODP trading service (IS13235, n.d.)) for locating services that are published using structured meta-information description of the service. We consider these descriptions as binding offers for 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. If the validation is successful, 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 web-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.

Business network model repository

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

The structure (topology) and properties of a business network are defined by its BNM that 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 non-functional properties. The ap-

```
service offer          := ((interface syntax) (interface protocol)
                            (information element format)
                           (nonfunctional aspects)*)*
                           (policies)
                            (platform requirements) (channel
requirements)
interface syntax       := <IDL specification> | <WSDL specification>
interfaceprotocol     := <partial ordering rules of operations in
syntax>
nonfunctional aspects  := <QoS offer> | <trust requirement> | <security
                          mechanism name>
information element format
                       := <schema>
policies               := <policy framework name> <policy name> <policy
                          value offer>
platform requirement   := <platform name>
channel requirements   := <channel type name> <binding type name>
```

Figure 4. Structure of service offers in web-Pilarcos.

proach combines ideas from 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 non-functional properties required. Role functionality is described as a composition of service types and role specific synchronization patterns. Synchronization patterns express causal relationships between actions in distinct services of a role (by setting preconditions for interactions using terms before, after etc).

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

Non-functional 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 captured into Figure 5.

Life-cycles of the contents in B2B repositories

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, pragmatical goals of members in the network. As the matching of network models is too hard a problem to solve by an automated process in general cases for a heterogeneous modeling 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 relationships between models and the transformation information associated. 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 becomes 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).

Figure 6 illustrates the various processes involved: publication of a) service types and b) business network models, c) service offers, and d) eCommunity population and negotiation processes. These processes are inter-related but not tightly dependent; for example new service types can be published without a business network model using them. Figure 6 illustrates these processes. The service publication functionality is similar to the UDDI (Belwood & et al., 2004) or the ODP trading mechanisms (IS13235, n.d.); while the type management system resembles the ODP type repository function (IS14746, 1999) and enforces a typing discipline to follow over service offer repositories. The BNM repository is a shared storage of business collaboration information that enable enterprises to share business transaction models, such as the ebXML-repository (Kotok & Webber, 2001), although with more automated and repeatable breeding process. The used notations are not discussed here, but they resemble ODP enterprise language and use XML-style notations (see (Kutvonen, 2004a) and (Kutvonen, Ruokolainen, Metso, & Haataja, 2005)). In the service publication process (step 1), 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 non-functional 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 con-

```
business network model  := (business process model)*
                           (role association requirement)*
                           (process policy rule)*
business process model  := (process model name) ((role) (role assignment rule)*
                           (role conformance rule)*)*
                           ((interaction) [channel requirement])*
                           (process policy rule)*
                           (breach recovery process)
role association requirement
                        := associate (role name) with (role name)
process policy rule     := <monitoring function call>
                          %% Invariant expressed in terms of roles, process models
                          %% and related policies. Includes "objective" of the community.
role name               := <role> in <process model name>
process model name      := <repository reference>:<identifier>
role                    := <repository reference>:<identifier>
role assignment rule    := (service type) (trading criteria)
role conformance rule   := <monitoring function call>
interaction             := ((epoch) ((policy guard) (information exchange))*
                           (epoch termination process)(epoch monitor rules))*
epoch                   := <epoch name>
epoch termination process:=   <reorganization process for the community,  adding and removing roles etc.>
policy guard            := <Each policy value denotes one of the
                            alternative processes within an epoch. The guard expresses
                            which alternative is in use within the community (instance).>
information exchange      := (from <role> to <role>: (information element) by
                            ("invocation" |" announcement" | "peer-to-peer protocol")
                            <service type>)*
information element      := <element name>
  %% The element name denotes a specification (XML/ontology based) that determines
  %% the information structure, and semantical relationships. A mapping to
  %% transfer syntax (information grounding) is separately used when service offers are
  %% matched agains the process requirements and when channels are established.
epoch monitor rules      :=  <Rules to be monitored by an infrastructure provided
                              function. These rules cover functional
                              conformance rules, nonfunctional conformance rules
                              and information conformance rules.>
service type             :=  <service interface syntax> <service interface protocol>
trading criteria         := <semantic requirements>
  %% Selection criteria for service offer repository (trader) using service semantics and
  %% business related attributes in service offers. The available attiributes are
  %% dependent on the service type and ontology within which the repository is formed.
channel requirement      :=  <declared values>
  %% Declared values within a framework that defines channel as a service with
  %% control interface, service level, and distribution transparency support.
breach recovery process  := (process model name) (breach type)
breach type              := "dissolving" | "restarting" | "continuable with sanctions"
```

Figure 5. BNF-style structuring of business network models. Parentheses are used for grouping words of a term together, angle brackets for terminals not fully specified here, and asterisks for repetition. *Editorial note: We are considering replacing this with an updated, more compact version.*

formance validation process. For this purpose, a service type corresponding the claimed service type is retrieved from the type repository (step 2). The service type defines syntactical structures for service interface signature and messages, externally visible service behaviour and semantics for exchanged messages (Kutvonen, Ruokolainen, et al., 2005). Conformance validation is executed by the service type repository holding the corresponding service type (step 3). Only after a successful validation, the service offer is published (step 4a), otherwise a service typing mismatch is reported between the service offer and its claimed type (step 4b).

In the figure, also the eCommunity establishment process is detailed: it is initiated by a willing partner, the corresponding business network model is first fetched from the BNM repository (step A). The population process (step D) provides a set of interoperable eCommunity proposals where 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 (step B), after which the service offer repository can be used to provide the corresponding service providers (step C). After population, and the subsequent negotiation, the eCommunity contract is received (step E) 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 (Lea Viljanen & Kutvonen, 2004; Kutvonen, 1998).

Verification and observation of interoperability

The web-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 web-Pilarcos

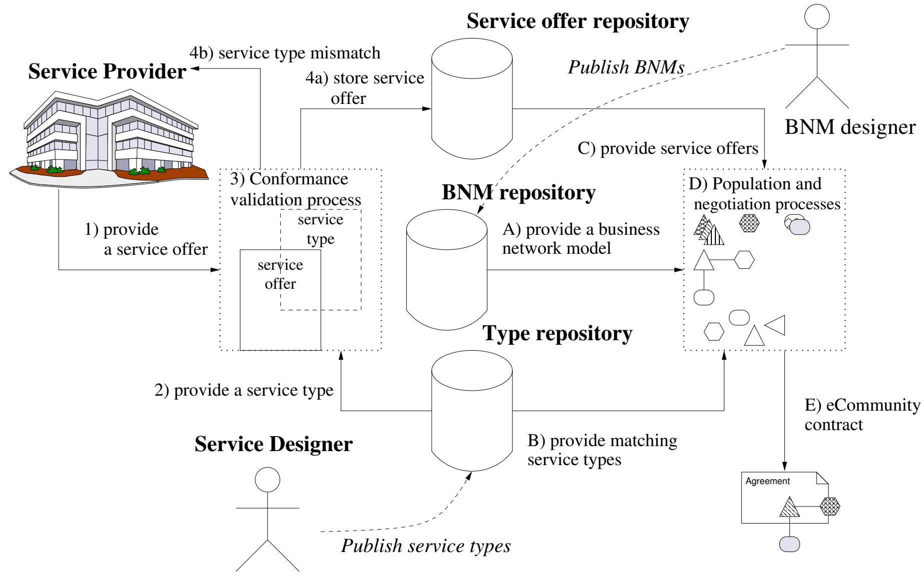

Figure 6. Repository usage during eCommunity life-cycle.

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 formal basis.

When an eCommunity is established, we ensure *sufficient* conditions for interoperability of services during service discovery and population. 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.

Conditions for an interoperable eCommunity are fulfilled by three solutions. First, 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, the use of constraint matching for accepting service offers to fulfill roles in the BNM. And third, the augmentation of the constraint matching process by the interference of further constraints arising from the selected offers for neighbour roles.

Relevant issues in role related constraints cover interface syntax with behaviour descriptions, syntax of documents to be exchanged, semantical aspects of control and information flows, and nonfunctional aspects like trust and business policies that further restrict the behaviour.

To promote evolution of syntactic structures of 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 web-Pilarcos platform but it will be implemented in future versions. Matching of semantic concepts shall be implemented using standard theories and tools, similarity to (Peer, 2002; Sriharee & Senivongse, 2003).

Behavioural interoperability is considered in the extent of verifying that service offers and role requirements for service behaviour match. We even do not seek to completely prove that a eCommunity behaves correctly, as this would need verification of behaviours between every possible participant in a eCommunity during its establishment process. Even in theory, a complete pre-operational verification of a eCommunity behaviour would be impossible, because of 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.

The monitoring system can be given a fairly free set of rules to monitor passing message traffic, different informational and behavioural aspects are fairly straightforward to monitor (Kutvonen, Metso, & Ruokolainen, 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 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 is not allowed to be exchanged (e.g., private documents, unknown structure); d) 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 NMAs 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.

Related work

The B2B middleware developed in web-Pilarcos 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 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, Alonso, Schuldt, & Schuler, 2000), rely on models for distributed business process enactment. However, the web-Pilarcos approach leaves enactment as a local business processing task, concentrating on interoperability monitoring.

The web-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 its composition using more basic computing services. The web-Pilarcos contract structure captures these aspects in its BNMs, binding requirements, and behavioural and non-functional 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). The web-Pilarcos aims for more real-time intervention.

In the web-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 abstract, external business process describing the collaboration requirements; the lower layer comprised of 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 web-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, web-Pilarcos uses negotiated policy-values to choose between predefined behaviour alternatives. The web-Pilarcos solution even requires that well-formed contracts include suitable recovery processes that involve whole communities. In contrast to METEOR-S, the web-Pilarcos platform has no central tool for making the whole of 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, Camarinha-Matos, & Vallejos, 2000). The main difference seems to be the separation of business services and B2B middleware services from each other. The web-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 of semantic verification and failure resolution, and use separate monitors to help and report.

Conclusion

The web-Pilarcos architecture provides a B2B middleware layer that supports management of virtual organizations. The management facilities are based on 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. Those monitors basically react to events that should not take place at that service or resource interface. Those self-protective reactions are then used as triggers for corrective actions for the benefit of the whole virtual organization.

The web-Pilarcos approach supports autonomous services to form federated communities. 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 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 for evolution needs. This is further discussed in (Kutvonen, 2004b, 2004a).

The provision of the web-Pilarcos architecture requires further development of business process modeling techniques. The collaboration of business processes or workflows should be modeled without unnecessary revealing of local processing steps. Instead, only the collaborative part (external view) should be agreed on and monitored. Work is already 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 semantical similarity of business process model as part of the interoperability tests during eCommunity establishment.

The federated approach has been criticized for the lack of advise 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 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 already operational communities. Existing models can be frozen so that new communities are not any 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.

An other 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 of 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, on the cost of the completeness of the results.

Specific features of the web-Pilarcos breeding environment include the level of automation expected, the relaxed matching of service types aimed, 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 web-Pilarcos facilities is in providing control over technological and evolution-involved problems, not in the aggressive enhancement to new business partners. The major development are on the architecture however is the trust management, on which we have started a separate development project (Lea Viljanen & Kutvonen, 2004).

The federated type repository service is an essential element of a B2B middleware that supports 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 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 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 on the trial of developing evolution support for dynamic, inter-enterprise networks.

References

Aggarwal, R., Verma, K., Miller, J., & Milnor, W. (2004, September). Contstraint 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.

Belwood, T., & et al. (2004). *UDDI version 3.0.* (Retrieved September 1st, 2005, from http://uddi.org/pubs/uddi_v3.htm)

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)

Daskalopulu, A., Dimitrakos, T., & Maibaum, T. (2002). Evidence-Based Electronic Contract Performance Monitoring. *The IN-*

FORMS Journal of Group Decision and Negotiation. (Special Issue on Formal Modelling in E-Commerce)

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.

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.

Grefen, P., Aberer, K., Hoffner, Y., & Ludwig, H. (2000). Cross-Flow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. *International Journal of Computer Systmes Sciences and Engineering*, *15*(5), 277-290.

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. (n.d.). (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*. Unpublished doctoral dissertation, Department of Computer Science, University of Helsinki.

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)* (p. 57-64). 29071 Málaga, Spain: University of Málaga.

Kutvonen, L. (2004b, September). Relating MDA and inter-enterprise collaboration management. In D. Akehurst (Ed.), *Second European Workshop on Model Driven Architecture (EWMDA)* (pp. 84–88). Canterbury, Kent CT2 7NF, UK: 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). 29071 Málaga, Spain: University of Málaga.

Kutvonen, L., Metso, J., & Ruokolainen, T. (2005, November). Inter-enterprise collaboration management in dynamic business networks. In R. Meersman & Z. Tari (Eds.), *OnTheMove 2005 DOA, ODBASE and CoopIS vol.1* (Vol. 3760). Berlin, Heidelberg: Springer-Verlag.

Kutvonen, L., Ruokolainen, T., Metso, J., & Haataja, J. (2005). Interoperability middleware for federated enterprise applications in web-Pilarcos. In *International Conference on Interoperability of Enteprise Software and Applications (INTEROP-ESA'05).* London, UK: Springer Verlag. (To appear.)

Lazcano, A., Alonso, G., Schuldt, H., & Schuler, C. (2000). The WISE approach to Electronic Commerce. *Computer Systems Science and Engineering*, *15*(5), 345–357.

Lea Viljanen, S. R., & Kutvonen, L. (2004, October). The TuBE approach to trust management. In *3rd iTrust internal workshop.* (To appear.)

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)

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.

Nardi, D., & Brachman, R. (2002). The description logic handbook. In F. Baader, D. Calvanese, D. L. McGuinnes, 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.

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). Introduction. *Communications of the ACM*, *46*(10), 24–28.

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 Busimess towards Smart Organizations* (pp. 281–290). Deventer, The Netherlands: Kluwer, B.V.

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/)

Ruokolainen, T. (2004). *Component interoperability*. Unpublished master's thesis, Department of Computer Science, University of Helsinki. (In Finnish)

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.