Interoperability middleware for federated enterprise applications in web-Pilarcos

Lea Kutvonen, Toni Ruokolainen, Janne Metso, Juha-Pekka Haataja

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

Summary. 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 enterprise applications in inter-organisational context. The approach is a federated one: All business applications 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 on the dynamic aspects and evolution of the network. This paper discusses the concepts provided for application and business network creators, and the middleware knowledge for interoperability support.

1 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

- form new business networks that provide added value services for clients;
- join to multiple networks at the same time without unnecessary restrictions on technologies or operational policies;
- take up new business processes and services cheaply;
- move existing business networks to new phases of life cycle so that new collaboration forms can be used;
- monitor the progress and correctness of the collaborative processes;
- automate some collaboration establishment and correction events; and
- protect 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, 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, like PRODNET [1], MASSYVE [2], FETISH-ETF [3] and WISE [4], 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 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 contents of this paper is a follows: Section 2 discusses interoperability challenges in the context of eCommunity management, and Section 3 briefly describes the web-Pilarcos B2B middleware services and repositories. Section 4 addresses the information repositories presented by the web-Pilarcos middleware. Section 5 discusses methods of finding interoperability problems and potential reactions on them.

2 eCommunity management and interoperability

The web-Pilarcos architecture proposes a model of inter-enterprise collaborations as eCommunities comprising of independently developed business applications. The applications represent local business services and processes, and are able to collaborate with other enterprises within those limits.

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 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. 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 application 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. 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.

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.

3 The web-Pilarcos B2B middleware architecture

The B2B-middleware platform provides a) advanced service discovery based on improved services typing and constraint based selection, b) contract based management of collaboration between autonomous services, and c) 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 service elements of the web-Pilarcos architecture address the need of joining four important processes: a) introduction of BNMs to the model repository, and introduction of supporting service types to the type repository; b) software engineering processes to provide implementations that correspond to the known service types and thus are applicable for the known BNMs; c) deployment of services and export of corresponding service offers to traders, effectively making a commitment to keep the service consistent with the service offer; d) eCommunity establishment process using the provided information.

These processes are only loosely interleaved. Business network models and the actual application 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.

The B2B middleware elements are illustrated in Fig. 1. The BNM design process involves 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. 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 from each other, even withdrawing or deprecating information may take place.

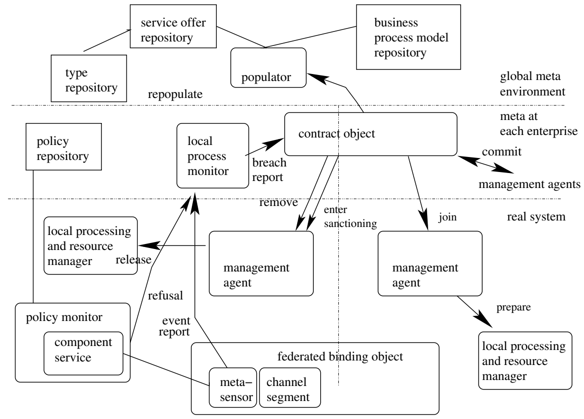

Fig. 1. The web-Pilarcos B2B middleware architecture, and flow of messages when a participant refuses a service due to policy conflict. Component service is the actual implementation for the functionality of a certain eCommunity role

The functional elements presented in Fig. 1 address the eCommunity life cycle management operations. 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 *Business Network Management Agent (BNMA)* 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 BNMA agent acting as a representative between 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).

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 BNMA 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 [5]) 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.

4 Interoperability knowledge in the global middleware

The three meta-information repositories in the B2B middleware 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 [6].

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 [6].

4.1 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 [7, 8]. 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 applications 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 [9]). 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 [10, 11]). 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 [12, 13] or more specialized XML-based ontologies such as RosettaNet, can be used [14]. The rules of the type system are based on behavioural session types, structural matching of syntactic information and semantic relations based on description logic [10, 15, 16]. Subtyping-like relationships that support service evolution are also important [17, 11, 16].

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 [6]. 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 *service offer repository* refers to services (like UDDI [18] and ODP trading service [19]) 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.

4.2 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 approach combines ideas from ODP enterprise viewpoint language [20] and those of separating functional units and their interconnection into distinct concepts of components and connectors [21].

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.

5 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 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 [22]. 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 [23]. 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 [24, 15]. 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, similarily to [25, 26].

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 [27]. 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 BNMAs 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 Conclusion

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 [28, 29].

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 [30].

Current work extends the monitoring system and the repositories.

Acknowledgment

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

References

1. Afsamanesh, H., Garita, C., Hertzberger, B., Santos Silva, V.: Management of distributed information in virtual enterprises - the PRODNET approach. In: ICE'97 - International Conference on Concurrent Enterprising. (1997)
2. Rabelo, R., Camarinha-Matos, L.M., Vallejos, R.V.: Agent-based brokerage for virtual enterprise creation in the moulds industry. In: E-business and Virtual Enterprises. (2000)
3. Camarinha-Matos, L.M., Afsarmanesh, H.: Service federation in virtual organisations. In: PROLAMAT'01, Budabest, Hungary (2001)
4. Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: The WISE approach to Electronic Commerce. Computer Systems Science and Engineering **15** (2000) 345–357
5. Neal, S., Cole, J.B., Linington, P.F., Milosevic, Z., Gibson, S., Kulkarni, S.: Identifying requirements for business contract language: a monitoring perspective. In: EDOC2003. (2003)
6. Kutvonen, L.: Trading services in open distributed environments. PhD thesis, Department of Computer Science, University of Helsinki (1998)
7. ISO/IEC JTC1: IS14769 ODP Type Repository Function. (2001)

8. Object Management Group: Common Facilities RFP-5: Meta-Object Facility. (1998) OMG TC Document cf/96-05-02.
9. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1. W3C. 1.1 edn. (2001)
10. Takeuchi, K., Honda, K., Kubo, M.: An interaction-based language and its typing system. In: 6th European Conference on Parallel Architectures and Languages. (1994) 398–413
11. Gay, S., Hole, M.: Types and subtypes for client-server interactions. LNCS **1576** (1999) 74–90
12. W3C: RDF Vocabulary Description Language 1.0: RDF Schema. (2004) W3C Recommendation 10 February 2004.
13. W3C: OWL Web Ontology Language Guide. (2004) W3C Recommendation 10 February 2004.
14. RosettaNet Consortium: Rosettanet implementation framework: Core specification v02.00.00 (2004) http://www.rosettanet.org/.
15. Jha, S., Palsberg, J., Zhao, T.: Efficient Type Matching. LNCS **2303** (2002) 187–206
16. Nardi, D., Brachman, R.: An Introduction to Description Logics. In: Description Logic Handbook. Cambridge University Press (2002) 5–44
17. Di Cosmo, R., Pottier, F., Rémy, D.: Subtyping recursive types modulo associative commutative products. Unpublished draft manuscript (2003)
18. OASIS consortium: UDDI version 3.0 published specification. (2002) http://www.uddi.orgspecification.html.
19. ISO/IEC JTC1: IS13235 ODP Trading function. (1997)
20. ISO/IEC JTC1: IS15414 ODP Enterprise Language. (2003)
21. Allen, R., Garlan, D.: Formalizing architectural connection. In: ICSE 1994, Sorrento, Italy (1994) 71–80
22. Garlan, D., Allen, R., Ockerbloom, J.: Architectural mismatch or why it's hard to build systems out of existing parts. In: ICSE 1995, ACM Press (1995) 179–185
23. Ruokolainen, T.: Component interoperability. Master's thesis, Department of Computer Science, University of Helsinki (2004) In Finnish.
24. Palsberg, J., Zhao, T.: Efficient and flexible matching of recursive types. Inf. Comput. **171** (2001) 364–387
25. Peer, J.: Bringing together semantic web and web services. LNCS **2342** (2002) 279–291
26. Sriharee, N., Senivongse, T.: Discovering Web Services Using Behavioural Constraints and Ontology. In Stefani, J.B., Demeure, I., Hagimont, D., eds.: DAIS 2003. Number 2893 in LNCS, Springer-Verlag (2003)
27. Kutvonen, L., Metso, J., Ruokolainen, T., Haataja, J.: Collaboration management in dynamic business networks. (2005) Manuscript.
28. Kutvonen, L.: Relating MDA and inter-enterprise collaboration management. In Akehurst, D., ed.: Second European Workshop on Model Driven Architecture (MDA). (2004) 84–88
29. Kutvonen, L.: Challenges for ODP-based infrastructure for managing dynamic B2B networks. In Vallecillo, A., Linington, P., Wood, B., eds.: Workshop on ODP for Enterprise Computing (WODPEC 2004). (2004) 57–64
30. Vähäaho, M., Haataja, J.P., Metso, J., Suoranta, T., Kutvonen, L.: Pilarcos prototype II. Technical report, Department of Computer Science, University of Helsinki (2002)