1 ADDRESSING AUTONOMY AND INTEROPERABILITY IN BREEDING ENVIRONMENTS

Toni Ruokolainen, Lea Kutvonen

Dept. of Computer Science
P.O. Box 68 (Gustaf Hllstrmin katu 2b)
FI-00014 UNIVERSITY OF HELSINKI
FINLAND

Toni.Ruokolainen@cs.Helsinki.FI,Lea.Kutvonen@cs.Helsinki.FI

Abstract:
Modern networked enterprises require flexibility and openness of computing systems to tolerate changes in ever-changing technology and business domains, and to gain a competitive edge. Dynamic establishment of virtual enterprises comprised of loosely-coupled and autonomic business services is currently considered as one of the best options to match these requirements. However, it is difficult to guarantee and maintain interoperability in such dynamic environments. This paper discusses the meta-information and maintenance facilities needed for verifying interoperability in federated collaborations. Special consideration is given for the aspect of autonomy during the virtual enterprise breeding process: what kind of autonomy can be supported and what kind of mechanisms are needed to support those aspects.

1.1 INTRODUCTION

Modern networked enterprises require flexibility and openness of computing systems to tolerate changes in ever-changing technology and business domains, and to gain a competitive edge. Dynamic establishment of virtual enterprises comprised of loosely-coupled and autonomic business services is currently considered as one of the best options to match these requirements. However, it is challenging to establish interoperability between the participating enterprise-information systems in such open and dynamic environments while respecting their autonomy.

Interoperability, or capability to collaborate, means effective capability of mutual communication of information. Interoperability covers technical, semantic and pragmatic interoperability. The technical interoperability concerns with connectivity between the computational services such that messages can be transported from one application to another. Semantic interoperability means that the message content becomes understood in the same way by the senders and the receivers. This may mean both information representation or messaging sequences. The pragmatic interoperability captures the willingness of partners for the actions necessary for the collaboration. The willingness to participate involves both capability of performing a requested action, and policies dictating whether the potential action is preferable for the enterprise to be involved in. While the technical interoperability can be established quite easily using modern middleware technologies, such as Web Services [W3C-WS, 2006], the semantic and pragmatic dimensions of interoperation are difficult to handle.

This paper analyses the metainformation and facilities needed for verifying interoperability during the breeding process for virtual enterprises. Special requirements of a federated collaboration environment which preserves autonomy while providing interoperability guarantees during community breeding and operation are discussed. First a framework for management of electronic business networks, or virtual enterprises, is introduced in Section 1.2. The web-Pilarcos architecture addresses the needs of modern inter-enterprise computing by utilising a federated collaboration model. An overview of the concepts and infrastructure services is given. Section 1.3 introduces the concepts and mechanisms for addressing autonomy-related aspects of inter-enterprise collaboration.

1.2 A BUSINESS NETWORK MANAGEMENT FRAMEWORK

The web-Pilarcos project concerns development of service-oriented middleware infrastructure for open inter-enterprise computing environments [Kutvonen et al., 2005a; Kutvonen et al., 2005b]. The web-Pilarcos framework proposes a federated model of inter-enterprise collaboration networks, or virtual enterprises, comprised of autonomic business services. Meta-information describing the properties of business services and networks, and infrastructure services providing a community breeding and management environment are utilised for establishing virtual enterprises. Web Services technology [W3C-WS, 2006] is used as a technical unification layer and communication middleware.

The inter-enterprise business collaboration networks are called eCommunities and they are established dynamically to serve a certain business scenario or opportunity. The operation of an eCommunity is governed by an electronic contract negotiated dynamically by the participants [Metso and Kutvonen, 2005]. Preliminary blanket agreements or initial trust relationships might be required before these electronic contracts can be formed. Given the necessary prerequisites, the collaboration is established by utilising the service trading and community population mechanisms provided by the middleware platform, and multilateral negotiations about the properties of the business network [Kutvonen et al., 2005a].

A business service denotes a set of functionalities provided by an enterprise to its clientele and cooperators, and is governed by the enterprise's own business rules and policies, as well as by business contracts and regulatory systems controlling the business area. 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 networks.

Functionality of business services is provided by the local computational services maintained by the enterprises. The computational service can be implemented using technologies suitable for the needs of each enterprise, such BPEL processes [Curbera et al., 2002], or CORBA [OMG, 2004] components. When the computational services are exported to the public service brokering mechanism as business services, they are provided with a Web Service [W3C-WS, 2006] compliant interface. This technological unification of business service interfaces provides the necessary means for achieving technical interoperability. Whereas the computational service provides the technical implementation needed to realise some business functionality, a business service extends this functionality with awareness of the business context comprising of policies, contracts and processes concerning its operation.

A business service is comprised of the corresponding computational service, a monitor, and a Business Network Agent as illustrated in the Figure 1.1. The business context awareness is provided during the operation of a business service in the corresponding Business Network Agent (NMA) which utilises the monitor to observe and control the operation of the computational service. Local contract and policy repositories are used to store information concerning the contract information, business rules and policies effective in the corresponding virtual enterprise.

The breeding environment consists of the public meta-information repositories and populator services. The service offer and type repositories provide a service trading mechanism, similarily to the ODP trading and type repository functions [ODP, 1997; ODP, 1999]. The type repositories are persistent storages of service type information which are used as the primary means for achieving interoperation between business services [Ruokolainen and Kutvonen, 2006]. The business network model repositories are used to publish and discover descriptions of virtual enterprise constellations, or Business Network Models (BNM) [Kutvonen et al., 2005a].

Given a business network model, the task of the breeding environment is to provide a set of potential eCommunities to be negotiated further by the participants. The community breeding environment utilises the meta-information services to accomplish this task. There are two phases in the breeding process: population phase performed by a populator service and subsequential negotiations performed by the NMAs of the participants. Breeding environment services, such as populators and type repositories, are not required from all sites, but can be provided as infrastructure services as a business on its own right.

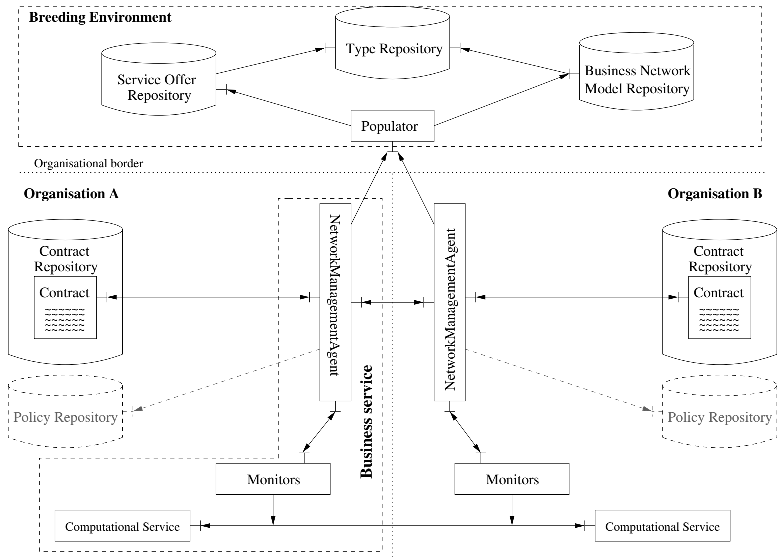

Figure 1.1 An overview of the web-Pilarcos architecture. Arrows represent communication relationships, boxes are active agents and cylinders are information repositories.

The populator represents a breeding process phase where appropriate business service providers are selected for eCommunity roles. The populator function takes a business network model and utilises service type and service offer repositories for fetching compatible business services providers for each eCommunity role. The populator selects the business services to an eCommunity on a basis of a contraint satisfaction process which considers the compability of the business service attributes. When a set of compatible service offers fulfilling the requirements of the business network model roles have been found, the populator returns the description of the populated eCommunity to the initiator of the population process. Population processes are initiated by enterprises willing to establish business collaborations.

After a successful population process, the corresponding eCommunity description is distributed to the participants. Negotiations are held between the participants to decide about the final properties of the collaboration. For the negotiation phase, the web-Pilarcos framework provides generic negotiation interfaces and meta-level protocols. The negotiations and eCommunity management during the operation of the community are handled by the NMAs [Kutvonen et al., 2005b; Metso and Kutvonen, 2005]. The NMAs provide uniform interfaces and they act as the representatives for the autonomous business services during the breeding process and operation of collaboration networks. The collaboration management interfaces of NMAs provide functionality for example for renegotiating part of the collaboration contract, to query the status of the contract, and to control transitions between eCommunity epochs, that is, distinct phases of the collaboration.

1.3 ADDRESSING AUTONOMY IN VE ENVIRONMENTS

Business services in an inter-enterprise computing environment are self governing entities. Their design, implementation and execution is legislated by the local administration policies. Usability of a business service is determined by not just technological issues such as availability and timeliness of service invocations, but also by willingness of the corresponding enterprise to take the requested actions and business strategies affecting the use of services.

Autonomy manifests itself during the virtual enterprise life-cycle as degrees of freedom given to its participants. The following degrees of freedom can be considered as the most important aspects of autonomy that should be supported by a modern VE support environment: 1) design and implementation of services, 2) willingness to collaborate, and 3) decisions concerning the operation of services. Nonetheless, interoperability and autonomy are inter-dependent aspects of collaboration: autonomy allowed for VE participants must be compensated with corresponding mechanisms to guarantee interoperation.

The freedom of design and implementation of computational services leads to technological and semantic heteronegeity that must be dealt with. Fortunately most of the implementation technology-related barriers for inter-enterprise collaboration between the computational services can be solved with use of appropriate middleware, such as Web Services [W3C-WS, 2006] or CORBA [OMG, 2004]. The web-Pilarcos middleware uses Web Services as the common grounding for establishing the necessary technological unification. Provided with the technical interoperability between computational services, the more challenging aspects of interoperability and autonomy between business services can be addressed.

Freedom of design inflicts technical interoperability issues between business services. These issues are related to the selection of business service and communication properties, such as security, trust, and quality of service attributes. Most of these attributes can be published in the service offers, thus affecting directly to the service discovery performed during the breeding process. The public attribute values can be either static or dynamic: static properties contain their values directly in the service offers (for example the service provider name) while dynamic property values must be queried from the corresponding service provider (for example the current price for a service invocation).

Some of the business service attributes, however, are confidential and are not published before necessary commitments or relationships between the partners have been initiated. These attributes manifest the pragmatic aspects of interoperation where enterprises may express their willingness to join a collaboration and expose their internal decision processes in controllable manner. For example trust-relationships, preferences for partners in cooperation, or pricing-information are typically private information, since publication of such information might hurt the enterprise's business.

We make a distinction between two kinds of private attributes: those that are available after a negotiation, and those that are available after an initial commitment. The "must negotiate"-kind of private attributes are used in situations where the service provider does not want to expose the attribute value before the client has exposed something of himself. For example the service

Visibility	Value type	Availability	Example
Public	Static	Service offer	Service provider name
Public	Dynamic	Service provider	Simple pricing information
Private	Dynamic	Must negotiate	Quality of service
Private	Dynamic	Must commit	Discount on the price of services

Figure 1.2 Characterisation of business service attribute kinds.

provider might want to decide on the quality of service on the basis of the client in question. This is something that can be achieved through negotiations without exposing the preferences affecting the decision. The client may refuse to join the collaboration after the corresponding attribute value has been exposed. The "must commit"-kind of private attributes are more demanding for the client, since the service provider does not reveal the attribute value but after a commitment to join the collaboration has been made by the client. The commitment would typically be attached with responsibilities, sanctions, or compensations for the client if he decides to decline the offer afterwards.

Figure 1.2 provides a characterisation of the different business service attribute kinds. The first two kinds contribute to the technical aspect of interoperability while the two latter kinds of business service attributes are used to express the pragmatic aspects of interoperability. Each attribute kind is attached with a type of the value (static or dynamic), availability (how the actual attribute value can be attained), and an example of a typical usage of such attribute kind.

The freedom of design induces semantic interoperability problems which must be solved during the VE breeding process. In the web-Pilarcos framework which follows the service-oriented computing approach, semantics are considered with respect to the meaning of individual messages exchanged between business services as well as the behavioural aspects of the business services. Service types which provide abstract descriptions of expected business service structure, behaviour, and document semantics are used for this purpose [Ruokolainen and Kutvonen, 2006; Kutvonen et al., 2005a]. Service offer and service type repositories provide the required functionality for validating semantic interoperability during community breeding; for performance reasons actual validation is not performed during service discovery but as part of the service offer publication process [Ruokolainen and Kutvonen, 2006].

A participant must have an opportunity to decide about joining a collaboration during the virtual enterprise breeding process. The willingness to collaborate is addressed during the negotiation phase of the breeding process. Negotiations are used primarily to come into conclusion about shared properties among the autonomic agents, however, negotiation is also a mechanism to introduce autonomic decision procedures into the breeding process. An enterprise can keep its decision procedures and preferences private, since a negotiation about joining the collaboration leaves an option to decline the invitation to join the community.

An enterprise must have the possibility to constraint the use of their services due to the local organisational policies. Organisational policies declare autonomic intentions of organisations and they are specified through the concepts of obligation, permission and prohibitions [Steen and Derrick, 1999]. Policies modify behaviour of business services by requiring certain actions to be taken instead of the others, or by prohibiting certain actions.

When organisational policies of collaborating participants are known beforehand, policy conflicts can be identified during the breeding process. Organisational policies are inherently dynamic entities and not even necessarily published outside the organisations. However, to establish interoperability some guarantees on stability of expected business service behaviour must be provided. For this purpose, different commitment types with respect to business service provision are introduced.

Commitments are classified on the basis of their time-frame and negotiability. The participants of an eCommunity may commit to their service offers over the whole virtual enterprise life-time, epochs, and business services sessions. ﬁIndividual actions or business transactions are considered too fine-grained to be used as objects of commitments. An epoch is a block of collaboration where the set of roles and services is stable. An epoch change captures a major reorganisation of the collaboration structure, membership, and commitments [Kutvonen et al., 2005a]. Service sessions are the smallest behavioural structures that can be committed to in the web-Pilarcos framework. The session boundaries are prescribed by the service types and they are used to define "natural" modular units of business service behaviour. Each of these commitment types can be in turn pre-negotiable or re-negotiable. A pre-negotiated commitment can not be changed between the commitment points, whereas re-negotiable commitments can.

1.4 CONCLUSION

The B2B-middleware developed in web-Pilarcos project provides support for autonomously administered business services that collaborate in a loosely coupled eCommunities. The eCommunity establishment and maintenance does not include facilities for distributed enactment of business processes, but instead focuses on ensuring semantic and pragmatic interoperability. Especially the aspects related to autonomy of the virtual enterprise participants are taken into considereation.

The web-Pilarcos middleware provides a virtual enterprise breeding environment and an operational environment (e.g, survey in [Camarnha-Matos, 2003]). In many projects, like PRODNET [Afsamanesh et al., 1997], MASSYVE [Rabelo et al., 2000] and FETISH-ETF [Camarinha-Matos and Afsarmanesh, 2001], the support environment provides facilities for negotiating and modelling the collaboration processes; the operational environment controls the enactment of the processes. Many of the virtual enterprise support environments use a unified architecture approach: there is a shared abstract model to which all enterprises have to adapt their local services. In contrast to this, the approach in the web-Pilarcos project is federated: enterprises seek out partners that have services with which they are able to interoperate. The federated model provides the necessary flexibility and mechanisms to address pragmatic interoperability issues and autonomy.

Acknowledgments

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 INTEROP NoE collaboration.

References

Afsamanesh, H., Garita, C., Hertzberger, B., and Santos Silva, V. (1997). Management of distributed information in virtual enterprises - the PRODNET approach. In *ICE'97 - International Conference on Concurrent Enterprising.*

Camarinha-Matos, L. M. and Afsarmanesh, H. (2001). Service Federation in Virtual Organisations. In *PROLAMAT'01*, Budabest, Hungary.

Camarnha-Matos, L. M. (2003). Infrastructure for virtual organizations – where we are. In *Proceedings of ETFA'03 - 9th international conference on Emerging Technologies and Factory Automation*, Lisboa, Portual.

Curbera, F., Goland, Y., Klein, J., Leyman, F., Roller, D., Thatte, S., and Weerawarana, S. (2002). *Business Process Execution Language for Web Services (BPEL4WS) 1.0.* http://www.ibm.com/developerworks/library/ws-bpel.

Kutvonen, L., Metso, J., and Ruokolainen, T. (2005a). Inter-enterprise collaboration management in dynamic business networks. In *OTM Confederated International Conferences*, volume 3760 of *LNCS*. Springer-Verlag.

Kutvonen, L., Ruokolainen, T., Metso, J., and Haataja, J. (2005b). Interoperability middleware for federated enterprise applications in web-Pilarcos. In *Interoperability of Enterprise Software and Applications.* Springer-Verlag.

Metso, J. and Kutvonen, L. (2005). Managing Virtual Organizations with Contracts. In *Workshop on Contract Architectures and Languages (CoALa2005)*, Enschede, The Netherlands.

ODP (1997). *ISO/IEC 13235: Information Technology – Open Distributed Processing. ODP Trading function.* ISO/IEC JTC1.

ODP (1999). *ISO/IEC 14769: Information technology - Open distributed processing - Type repository function.* ISO/IEC JTC1/SC7.

OMG (2004). *Common Object Request Broker Architecture (CORBA) v3.0.3.* Object Management Group. OMG Document formal/04-03-01.

Rabelo, R., Camarinha-Matos, L. M., and Vallejos, R. V. (2000). Agent-based brokerage for virtual enterprise creation in the moulds industry. In *E-business and Virtual Enterprises.* http://gsigma-grucon.ufsc.br/massyve.

Ruokolainen, T. and Kutvonen, L. (2006). Service Typing in Collaborative Systems. Accepted to the IESA'06 conference.

Steen, M. and Derrick, J. (1999). Formalising ODP enterprise policies. In *EDOC'99*, pages 84–93. IEEE.

W3C-WS (2006). *Web Services Activity.* W3C. `http://www.w3.org/2002/ws/`.