# Using business network models in web-Pilarcos

Lea Kutvonen

Department of Computer Science, University of Helsinki, Finland
Lea.Kutvonen@cs.Helsinki.FI

## 1 Introduction

Current developments on IT supported markets draw enterprises to join dynamic business networks. In these networks, enterprises can jointly provide added value services and thus gain and hold their share at a business domain. However, challenges for the collaborative computing environment are created by the autonomy of enterprises and the dynamicity of collaborations.

In the web-Pilarcos project, our interests focus on business networks (i.e. virtual enterprises, VEs; federations) that are loosely coupled and flexible. The enterprises of a business network are autonomous in many respects, such as the selection of service components they place externally available and the evolution life-cycle of those services. Even services used by some existing VE should be withdrawable. The enterprises are autonomous in selecting their computing platform, and schedule of technical changes in it. Naturally, the VE management should leave enterprises autonomous on decisions of the kind of collaborations they enter, and partners they may accept. The VEs should be flexible in terms of changing partners (because of their own initiative or because contract breaches), and changes in epochs (each of which may have different set of participants and responsibilities, and differing interaction patterns).

The web-Pilarcos project uses a federated approach to VE coordination. Federated systems are such that (at least for some collaboration features) collaboration is reached without pre-established shared computing platform and without implicitly shared metamodel on the joint activities. The web-Pilarcos architecture trusts on a breeding environment that supports negotiation on metainformation-based collaboration models and local mappings to dynamically selected platform facilities.

The collaborating service components are aware of the business process model used between them, but do not implement the control of it. The service components are loosely coupled; they react on invocation or notification messages received, and are isolated from each other by infrastructure connectors that capture required transformations. The application and business process view to connectors between applications is abstract; however applications run each on an independently selected platform and may have different view of syntactical structuring of exchanged information.

This paper outlines the web-Pilarcos environment and the use of business process models in it. Instead of discussing related work, space is given for future work.

## 2 The web-Pilarcos services for VE management

The web-Pilarcos architecture provides breeding facilities and operational environment for VE management. The breeding environment provides repositories for business process models, service types and service offers. The key agent is a federation populator that creates verified suggestions for federation contracts. A federation contract – explicitly but in platform-independent terms – declares the functional cooperation patterns between members, identifies the members of the VE, and captures the non-functional features of the collaboration. It also captures the rules and processes related to dynamic changes of the VE. The populator uses a business network model as a contract template and retrieves suitable service offers for suggested federation contracts in such a way that interoperability is ensured. By this process, the collaborators reach a shared understanding of the business network model they conform to, what their mutual responsibilities are within that model, and how information and abstractions of processing are expressed.

When a federation contract is formed, essentially the interoperability within the VE is ensured, i.e. effective capability of mutual communication of information, proposals and commitments, requests and results. This requires technical, semantical and pragmatical interoperability. Technical interoperability captures the use of shared transport protocols and signaling; semantical interoperability involves preserving the meaning of operations and information despite of different representation formats; and pragmatic interoperability is about the willingness and capability of collaborating. These levels of interoperability are reflected by the required service offer contents.

We consider business network models as combinations of business process models; some roles are defined as connecting roles for two or more process models in the network. Each business process model captures a separate functionality, such as billing or goods delivery control. From these elements, larger networks can be constructed by reusing models. Naturally, the models need to be generic enough for reusability, and include features for refinement in the breeding process. We consider allowing alternative behaviour patterns in the models, and specialization by indicating the shared strategy choice by policy values.

A business process model describes only the externally visible view of the interactions between participants. Models should capture conversation patterns, information exchange and synchronization needs between partners. The models need to abstract away from the local processing detail. This is in contrast to many current workflow and business process management solutions that concentrate on synchronizing local processes. Abstraction of local solutions has the benefit of reducing requirements on shared computing platform, and of increasing evolution support in the system.

The business process model description includes roles and interaction models between roles. Each role definition presents requirements on components fulfilling that role, both for selection and behaviour conformance. Similarly, the interaction model gives conformance requirements for the communication technology and events.

The model elements are similar to those presented in ODP enterprise language [1] the development of which we have participated. However, we emphasize the nature of role and interaction definitions as conformance requirements. The operational environment of web-Pilarcos architecture uses these models for contract monitoring, whereas other approaches may use business process models for generation of business processing elements or abstract execution of the defined processes.

As an interesting addition, the processes can be split into epochs, each of which have independent set of roles and interactions between roles. The epochs represent different life-cycle phases of the VE. Naturally, the model has to declare how existing members are transferred from a role in one epoch to another in the next epoch.

The populator agent forms federation contracts using information about the indicated business network model, available service offers from enterprises, and semantical information about service types, including their interoperability and substitutability relationships. The population process is complex as none of these metainformation elements dominates the process sufficiently. Figure 1 shows how business network model determines the federation contract structure. It also illustrates how service offers contain technical and semantical interoperability information that becomes a critical part in the federation contract and makes selection of performers for neighboring roles interdependent on multiple aspects. On technical level, also channel types supported at each enterprise become a restriction on the federation contracts.

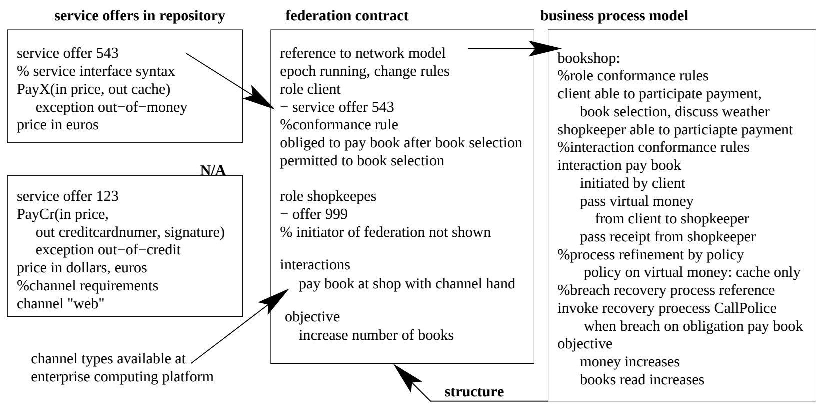


**Fig. 1.** The web-Pilarcos variation of service offers and business process model used for federation contract creation.

The federation contract is used during collaboration life-time to control interactions within the federation, and to facilitate changes to the federation itself. In addition to the explicit membership operations, monitoring the federation behaviour may cause detection of contract breaches and entering a breach recovery process. Breach recovery processes are separately specified and selected for the federation contract.

The operational time collaboration pattern involves three elements. First, the service components that provide computation for reaching the goal of the collaboration. These components form a partner grid with various addressee-counterparty patterns. The second element represents an authority that supervises the interactions between the partners of the grid. This authority has a "trusted third party" role – although the responsibilities of this federation contract may be distributed in practice. The third element captures execution of the contracted computation and communication under the supervision of the federation contract. Pragmatic decisions that deal with operational time requests on enterprise resources may cause contradictions and capturing them requires runtime monitoring and exception handling mechanisms defined as business process model elements. Figure 2 illustrates how each federated system is instrumented with interaction guards that check that conformance both towards local policy repositories and towards the federation contract. This structure support enterprise autonomy, as local policies are consulted also during federation lifetime. For example, each resource can be guarded against unwanted access even when some services using the resource are open for entering federations.

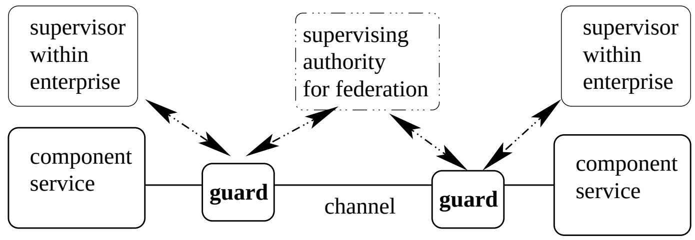


**Fig. 2.** Supervised interaction patterns.

## 3 Future work on modeling and repositories

The web-Pilarcos model discussed above is partially prototyped. Preceding work in Pilarcos project [2] resulted into a prototype on the federation populator and service offer repositories. The work in web-Pilarcos is currently in progress of implementing a prototype on the operational time control and monitoring system [3]. The Pilarcos prototype is based on CORBA component model, while the current environment uses web services technologies for interactions between the major infrastructure services.

Specific features for web-Pilarcos include the aim for hiding all local computing, and monitoring of collaboration instead of enforcing interoperability by abstract execution or such methods.

For the Pilarcos and web-Pilarcos projects, we have developed some initial aspects of the business process models and type systems. As followup we intend to further develop a business process repository and a service type repository suitable for the breeding environment needs. The challenges are twofold: verification of the business process models and the ontology of processes and related service types.

In inter-organizational collaboration environment repositories that provide publicly available business process models are fundamental tools. Published models would provide guidance to the markets of service components and guide evolution of new standard frameworks such as RosettaNet presents today. The repositories should capture not only "standard models" that can be applied in all VE systems, but in addition to common models specialize on alternative models on business domain specific problems. Such repositories should organize business process models according to suitable ontologies, thus providing help for matching of differing representations of similar business processes. An essential requirement for a business process model repository is that the models published are verified using static analysis methods. Features like completeness and simple forms of correctness should be checked.

## References

1. ISO/IEC JTC1: Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. ODP Enterprise Language. (2003) IS13235.
2. Kutvonen, L.: Automated management of interorganisational applications. In: Enterprise Distributed Object Computing (EDOC2002).
3. Kutvonen, L.: Management of dynamic business networks from autonomic enterprises. (2004) Submitted manuscript.