

# What applying of the ODP viewpoints teaches us about tool-chains

Lea Kutvonen

Department of Computer Science, University of Helsinki, Finland

Email: Lea.Kutvonen@cs.Helsinki.FI

## Abstract

*For some time, we have focused our research on the generic B2B middleware services for managing inter-enterprise communities of autonomous business services. In contrast to some earlier papers where the relationships between the Pilarcos concepts and their counterparts from the ODP-RM are shown, this paper approaches the management system for inter-enterprise communities itself as the ODP system of interest. The lessons learned underline a number of research challenges we are currently working with, especially for the B2B middleware repositories, their federations, and service development tools relying on the federated repositories.*

## 1 Introduction

In terms of developing enterprise systems or business services for open markets, we are currently witnessing a relative revolution from monolithic stovepipe systems to systems with open, evolvable architecture and adaptation to differences in business processes and business rules between enterprises. The list of present buzzwords, such as SOA or SOC (service-oriented architecture and computing) [17, 23], MDA or MDD (model-driven architecture or development) [4, 22], and VE (virtual enterprises) (e.g. [1, 5, 13, 19, 24]), take us closer to the aimed global computing architecture.

In this revolution, the essential axes are as follows. First, instead of integrating monolithic enterprise computing systems, the aim must be set at interoperability of large-granule business services provided by companies on open, interoperable computing platforms. Second, instead of generating service implementations that are able to manage interoperability or inherently preserve model-level interoperability, the open computing platform should provide generic middleware services to call. Third, instead of ensuring that the service development processes are integrated and interoperable, the

products should use high-level interoperability middleware services and not get involved in interoperability or community management at all. Finally, new facilities should be introduced to capture those business-oriented aspects that refine business processes in a manner meaningful to business-strategists, e.g. needs for security, trust, exceptions on contracts, and managing business value.

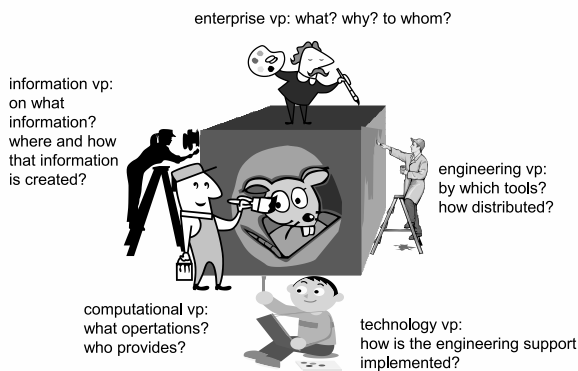
The Pilarcos middleware services [11, 12] drive towards these targets, as described below. To collect the relevant buzzwords, the approach is model-driven, reflective, and provides inter-enterprise service-oriented computing. The architecture is strongly based on autonomous agents representing each independent enterprise (regardless of whether that is a company or governmental organisation or a legally less independent department) and managing dynamic communities at operational time through eContracts. The intended tool-chains produce artefacts such as a) models for community structure and behaviour rules a basis for eContracting, b) service types for matching purposes, c) service descriptions and service realisations, and d) management information and policies.

This paper approaches the Pilarcos architecture in terms of new tool-chain requirements. The exercise is performed by outlining the main issues of the Pilarcos B2B middleware from the ODP viewpoints. This leads us to the requirements for the tool-chain for various products required for the gradual evolution of eContract-based communities. The lessons learned underline a number of research challenges we are currently working with, especially for the B2B middleware repositories, their federations, and service development tools relying on the federated repositories. We conclude by reflecting the usability of the ODP viewpoints as a design framework.

## 2 The Pilarcos middleware from four ODP viewpoints

Although we here assume familiarity with the ODP-RM (Open distributed processing reference model) and its five viewpoints [6–8, 14, 18], a few positioning sentences are needed.

Although the viewpoints are often considered as layers and providing a stepwise development process, our approach is, however, to keep the viewpoints strictly as projections of the same system – except that the viewpoints express the required and sufficient building instructions for the system. In the case that the available instructions contradict each other, the consistency control mechanism within the tools or the self-aware system must rise to the issues. This is illustrated in Figure 1, captured from teaching material.



**Figure 1. The ODP viewpoints.**

Effectively, the tool-chain suitable for building inter-enterprise communities is a self-aware system into which a set of separate design tools can introduce a) new building blocks, b) new compositions from those blocks, and c) new consistency requirements within the compositions.

Another point in Figure 1 deserves attention: For inter-enterprise computing we do not use the technology viewpoint in most discussions at all. Due to the goal of technology-independent solutions, the technology decisions become private, except of those that directly affect the transfer of messages between systems. Even there, we expect an open binding pattern [3] into which interceptors (i.e., transformers) are chosen depending on the communication environment and the context (with challenges on performance, flexibility, reliability, and security).

## 3 Enterprise specification

The enterprise viewpoint specification of the Pilarcos B2B middleware sets the *objective* as providing a support environment for business service developers and composers so that they are provided necessary functions for requesting community management services and interoperability ensuring facilities. This is done in two separate *epochs*, namely the breeding environment [12] and the operational time environment [11]. As the goals and functionalities of these epochs have been described earlier, we concentrate here on the roles involved, and policies.

The breeding environment roles are as follows: *Communities* are built by network management agents from business services. *Network management agents* each represent their local enterprise in negotiations and management activities of the community. *Business services* are independent configurations of enterprise applications providing a meaningful whole. *Service providers* make business services available on an open service market by advertising their services in terms that are made commonly understandable by publishing service types for common shared structure and vocabulary. *Service type designers* design these service types, and make them publicly available. *Business network modellers* are business experts and understand motivations in choosing certain business processes; supported with modelling languages and technical experts they provide models that can be automatically manipulated.

These roles describe the need of a tool-chain for building up metainformation for the community establishment. In short (and showing details from other viewpoints), the tool-chain is illustrated in Figure 2.

The above gets more interesting as we add policies, especially obligations, to the enterprise specification. To give a first example, service providers are required to provide service offers that are accountable, legally binding offers, and only offer service properties that can be met. As it is clear that there is no engineering mechanism that would provide such a property, we must assume that there is social enforcement support, for example trust management in place. Depending on how prompt and federated this kind of new management is required to be, we end up in computational models ranging from preventive [15] to post-sanctioning systems [16].

Another policy example is that the modellers and designers are required to publish only valid models that are based on best practices and follow the regulations on the business area, and to maintain those published models available as long they are used. On the other side, the users of these models may choose whether to

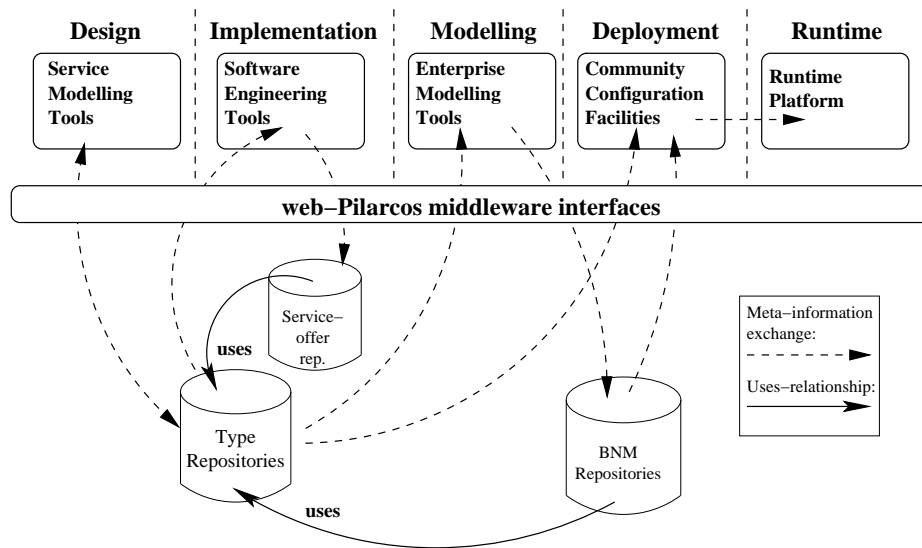


Figure 2. The tool-chain for creating necessary metainformation and business services [20].

use new or old models. Because of the need for autonomy at each site, it is not possible to externally enforce appropriate behaviour from the publishers, and therefore, we end up choosing a solution where repositories hold the advertisements, and also police the quality of published material. It is a matter of the designers' taste whether these repositories also become roles in the specification; we prefer not to do so because the inclusion would require addition of clearly computational design choices as well.

In the objectives clause above, we have already included an overall architecture requirement that in traditional software development processes would be part of the early phases: we have considered it as an essential *engineering specification principle* that the model information provided by designers and modellers is made usable both in the breeding and operational environment epochs.

This decision differentiates between the present wave of MDA tool development and the future tool development based on the eContracting middleware. The MDA tool track is involved in ensuring model-interoperability [2], while the middleware track leaves the software production tools as private issue, and is involved only with the contents of the eContracts and availability of interceptor transformations for communication. Still, the transformations may need to capture modifications from messaging techniques through semantics representation to business level process differences or resolving policy-mismatches. The architecture is further discussed after analysis of the information specification.

## 4 Information specification

The information viewpoint specification defines the relevant information repositories (i.e., logical bodies of knowledge) and items of information to be exchanged between roles. It is relevant to focus on the information semantics, not on the representation engineering. In the Pilarcos publications, we have described service offer repositories [9], service type repositories [21], and business network model repositories [21], as well as the structure for eContracts [10]. The same publications cover the *computational specification*, introducing the operations on these metainformation repositories.

The information viewpoint structuring rule requires information and information repository behaviour to be defined through *static, invariant and dynamic schemata*. The static schemata define the structure and the semantics of each item and repository.

The invariant schemata define the ontology rules and type discipline for the repository. The invariants may state for example, that service offers can only exist if there is a published service type, and the service offer conforms to all criteria defined for that service type. The invariant indeed locksteps the service types and offers to an understandable whole, without preventing new types and corresponding offer sets from emerging. The invariant schemata can also be used for defining validation and verification rules for models to be entered in the repositories. We currently have work in progress for defining these rules and applying them to the type repository implementation [21].

The dynamic schemata define the acceptable modi-

fications to the information state. Such state changes include, for example,

- change of business policy values,
- change of monitoring rules over the community behaviour against the eContract, and
- creating a substitutability relationship between two service types by introducing a transformation between them and storing it into the service type repository.

The first two examples of dynamic schemata items raise a design issue for the present set of business process description languages. We have based our business network models on the ODP enterprise language specification, as it provides the essential concepts required for collaborative behaviour description. In early contributions to the development of the ODP enterprise language, we pushed for the inclusion of assignment rules for services into roles, and furthermore, role criteria for checking conformance during the community lifetime. This opens space not only for the functional monitoring of service behaviour, but also for the non-functional aspects.

At present there are a number of emerging languages and MDD-related projects where the non-functional aspects are becoming part of the business process definition language. However, it is by far not clear that the monitoring aspects should be statically built in into the process definition.

Another idea is to provide a tool for each enterprise administrator to modify the monitoring rules, methods and reactions at operational time, depending on the perceived cost versus importance of the monitoring. Naturally, these administrative actions may have their effect on the eContract between autonomous partners in the community, thus requiring a negotiation protocol to be used between parties.

This is one of the areas where the MDD track and the B2B middleware track lead to differing sets of tools and different level of functionality for community management.

The last example above falls into a category where the middleware track ends up in far less demanding software generation challenges. Looking into the management of non-functional aspects of interoperability by MDD tools, a need emerges to either produce a very generic framework able to optimise the final configuration, or to produce a sophisticated tool environment that effectively restricts the functionality to reasonably-performing solutions. In the middleware track, the middleware itself provides a framework with the generic optimisation facilities implemented once in

an interoperable manner, and moreover, also the weaving of various aspects into the behavioural structure in a consistent way.

## 5 Computational and engineering specifications

The computational viewpoint specification covers the operations for metainformation repositories, but also for agents such as the populator [10], eContract negotiation and monitoring [11], and trust management [10, 25].

The engineering viewpoint specification is spelt out mostly together with the computational view, but still needs to be complemented with the distribution decisions for the metainformation repositories.

It is not feasible to assume all the essential repositories to be centralised, nor that each enterprise would have a private repository. The social infrastructure needs and business strategies define whether an enterprise assumes a private repository or trusts an external one. It is to be expected that service offer repositories are fairly common and overlapping in terms of offers; this allows different service type ontologies to be utilised when marketing services. Service type repositories are expected to have less need for marketing their contents, but still have a fair load to share between each other. Especially for service type repositories the availability of published information is essential. For the business network model repositories, we would expect organisations such as standards bodies, governmental agencies and market drivers on various business areas to provide model information.

A fundamental engineering challenge is securing the middleware repositories from unauthorised modifications or fraudulent publications. Furthermore, there may be needs to compare the reputation of repositories in markets for their quality. But most importantly, it is essential that the identification of the repositories and the publishers is made tractable.

## 6 Conclusion

This paper has given an example of the usability of the ODP viewpoints as a design framework for a very large scale distributed system. We have found that returning to the basics of viewpoint specification does indeed clarify the design, and helps in searching functionality missing from the architecture. When we have used the ODP viewpoints in smaller student projects, we have found that the prerequisite of being able to efficiently use the ODP framework is to have sufficient

knowledge of concrete distributed systems prior to the introduction of the ODP terminology. The thinking patterns provided by the ODP-RM are good for comparing solutions, and ensuring a consistent design, but too abstract for the basis of a learning experience.

As we have demonstrated here, even a rather informal analysis of the viewpoint specifications on a system that is not originally designed using the viewpoints systematically (or consciously) brings up new challenges for further research and provides a number of guidelines for already running studies.

This exercise indicated the following needs and possibilities. First, new lessons were learnt about the relationship of the ODP-RM, MDD, and SOA. Second, we were able to improve the formalisation of the meta-information repository semantics. Third, we were able to solve a nagging discrepancy in the approaches of the business process definition languages and eContracting in the area of non-functional aspects. Finally, we identified a number of trust and security related requirements for the middleware repositories for further work.

## Acknowledgement

This article is a product of the work performed at the Department of Computer Science at the University of Helsinki, running the Collaborative and Interoperable Computing group towards the next project, SOAMeS.

## References

- [1] R. Aggarwal, K. Verma, J. Miller, and W. Milnor. Constraining driven web service composition in meteor. In *Proceedings of the IEEE SCC*, 2004.
- [2] C. Atkinson and T. Kuhne. Model-driven development: a metamodeling foundation. *IEEE Software*, (5):36 – 41, Sept. 2003.
- [3] G. S. Blair, G. Coulson, N. Davies, P. Robin, and T. Fritzipatric. Adaptive Middleware for Mobile Multimedia Applications. In *Proceedings of the 8th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 1997.
- [4] D. S. Frankel. *Model Driven Architecture - Applying MDA to Enterprise Computing*. OMG Press, 2003.
- [5] P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig. CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises. *International Journal of Computer Systemes Sciences and Engineering*, 15(5):277 – 290, 2000.
- [6] ISO/IEC JTC1. *Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. IS10746 Reference Model of Open Distributed Processing*, 1996.
- [7] ISO/IEC JTC1. *Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. IS15414 ODP Enterprise Language*, 2003.
- [8] L. Kutvonen. Architectures for Distributed Systems: Open Distributed Processing Reference Model. In *HeCSE Workshop on Emerging Technologies in Distributed Systems*, Lammi, Finland, Jan. 1998. Helsinki University of Technology, Digital Systems Laboratory, Series A, Number 50.
- [9] L. Kutvonen. Automated management of inter-organisational applications. In *6th International Enterprise Distributed Object Computing Conference (EDOC 2002)*, 2002.
- [10] L. Kutvonen, J. Metso, and S. Ruohomaa. From trading to ecommunity population: Responding to social and contractual challenges. In *The Tenth IEEE International Enterprise Computing Conference (EDOC2006)*, 2006.
- [11] L. Kutvonen, J. Metso, and T. Ruokolainen. Inter-enterprise collaboration management in dynamic business networks. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, Agia Napa, Cyprus, Nov. 2005.
- [12] L. Kutvonen, T. Ruokolainen, and J. Metso. Interoperability middleware for federated business services in web-pilarcos. *International Journal of Enterprise Information Systems (IJEIS)*, 2006.
- [13] A. Lazcano, G. Alonso, H. Schuldt, and C. Schuler. The wise approach to electronic commerce. *International Journal of Computer Systems Science and Engineering*, 2000.
- [14] P. Linington. RM-ODP: The architecture. In K. Raymond and L. Armstrong, editors, *The 3rd International Conference on Open Distributed Processing - Experiences with distributed environments*, pages 15–33, Brisbane, Australia, 1995. Chapman & Hall.
- [15] J. Metso and L. Kutvonen. Managing Virtual Organizations with Contracts. In *Workshop on Contract Architectures and Languages (CoAla2005)*, Enschede, The Netherlands, Sept. 2005.
- [16] Z. Milosevic, P. F. Linington, S. Gibson, S. Kulkarni, and J. B. Cole. Inter-Organisational Collaborations Supported by E-Contracts. In W. Lamersdorf, V. Tschammer, and S. Amarger, editors, *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)*, IFIP Conference Proceedings, pages 413–429. Kluwer, 2004.
- [17] M. P. Papazoglou and D. Georgakopoulos. Introduction. *Communications of the ACM*, 46(10):24 – 28, 2003.

- [18] J. R. Putman. *Architecting with RM-ODP*. Prentice Hall, 2001.
- [19] R. Rabelo, L. M. Camarinha-Matos, and R. V. Vallejos. Agent-based brokerage for virtual enterprise creation in the moulds industry. In *E-business and Virtual Enterprises*, 2000. <http://gsigma-grucon.ufsc.br/massyve>.
- [20] T. Ruokolainen and L. Kutvonen. Service typing in Collaborative Systems. In *Interoperability for Enterprise Software and Applications Conference (I-ESA2006)*. Springer Verlag, 2006.
- [21] T. Ruokolainen and L. Kutvonen. Service Typing in Collaborative Systems. In *Interoperability of Enterprise Software and Applications (I-ESA 2006)*, 2006.
- [22] J. Siegel. *Developing in OMG's Model-Driven Architecture*. Object Management Group, Nov. 2001. White paper, revision 2.6.
- [23] M. P. Singh and M. N. Huhns. *Service-Oriented Computing: Semantic, Processes, Agents*. John Wiley & Sons, Ltd., 2005.
- [24] W.-J. van den Heuvel and H. Weigand. Coordinating web-service enabled business transactions with contracts. In *Proceedings of the 15th Conference on Advanced Information Systems Engineering. LNCS 2681*, pages 568–583. Springer Verlag, 2003.
- [25] L. Viljanen, S. Ruohomaa, and L. Kutvonen. Guarding enterprise collaborations with trust decisions—the TuBE approach. In *Proceedings of the First International Workshop on Interoperability Solutions to Trust, Security, Policies and QoS for Enhanced Enterprise Systems (IS-TSPQ 2006)*, Mar. 2006.