Building B2B middleware – Interoperability knowledge management issues

Lea Kutvonen

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

Introduction

Inter-enterprise computing between autonomous business services creates two challenges, management of the collaboration and ensuring adequate interoperability between the services used for that collaboration [2]. This paper discusses a major change in the interoperability goals themselves, and outlines a solution for agile management of interoperability knowledge.

The solution is part of the work in CINCO group (http://cinco.cs.helsinki.fi), and is described from middleware perspective. However, the solution creates next wave research directions for areas related to the inter-enterprise computing problem, such as ontology development and use in large distributed systems, multiagent systems, and dynamically evolving type disciplines.

Interoperability challenges

The interoperability goals are changing with the maturity of B2B collaboration support. Each evolution phase has its characteristic challenges and solution architectures, as illustrated in Figure 1. The issues of interest focus on the second and third wave, while the first wave completes the picture by showing the traditional integration of application silos; typical solutions included data integration, presentation of joint portals, application integration, distributed workflow management and use of middleware.

The second wave introduces generated solutions that are based on shared models. The emergence of service-oriented architectures(SOA) [4] to a wide audience has secured the use of the concepts of services, e-contracts, and metainformation for describing services. On this basis, the model-driven engineering approach (MDE) [7] provides tools for creating a unifying model for collaboration and generation of services and workflows that ensure interoperation between services provided by collaborating enterprises. Interoperability is ensured by joint design efforts and interoperation of design tools used at

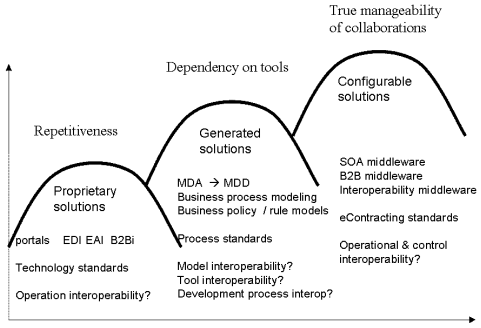

Fig. 1. From manual integration to middleware supported interoperability.

each collaborators system. The interoperability challenges focus on the production tools, their ability to exchange models and to generate logically similar implementation skeletons onto technically differing platforms.

The third wave illustrates the future enterprise computing systems that contain common, generic facilities for federated management of inter-enterprise collaborations. We call them B2B middleware [2]. The goal of this middleware layer is to provide a breeding environment for the establishment of new collaborations, and an operational time environment for controlling them. The automation of processes in these environments require though a well-formed set of knowledge about the interoperability features of services involved. Further, as the services and business processes of enterprises change, the interoperability knowledge must be dynamically increased. Thus, the interoperability knowledge is considered to be dynamically evolving, strictly typed and regulated by an evolving type discipline, heterogeneous in representation, and commonly available. Towards this goal, there is still lack of shared ontologies for e-contracts, protocols for automated management of inter-enterprise collaborations, and control of the nonfunctional aspects of these collaborations.

For the third wave solution, the B2B middleware services of each enterprise are supported by a global knowledge base organised into distributed repositories for service offers, service types, and business network models. The service offers give us information of the actual services, while the business network models define the structure of the collaboration. The service types provide a bridging concept between the service offers and roles.

Using the knowledge gathered into these repositories the B2B middleware agents can collect suggestions for new collaborations, and furthermore, check and negotiate for a multi-party contracts so that all partners a) share the intent of using the same business network model; b) conform to the role requirements given to them; c) share NFA models and communication channel types with those partners they have direct communication with; and d) conform to the generic policies (business rules for example) defined for the collaboration.

Extensible type discipline for interoperability knowledge

The main challenge for the interoperability knowledge management is to provide an extensible discipline to capture detailed enough ontology of business network models, service types, and service offers for automated use in the interoperability checking both at establishment and operational time. This discipline provides the inter-enterprise collaborations a kind of interoperability safety, analogous to strongly typed programming languages supports type safety. We have chosen to use the defined business network models as the topmost level of ontologies. Each business network model gives a root for a naming scheme within the ontology; the ontology is dynamically formed by accumulating a view to business network model, related service types, and conformant service offers. The required knowledge is based on thoroughly studied ontology and requirements for the relationships between the concepts [5].

For holding the three kinds of information, we use three kinds of metainformation repositories structured according to MOF [3][1]. The repository contents are defined as follows. The service offer repository: M0) actual service offers, M1) uses structures and names provided by service types, and M2) service offer content rules. The service type repository: M0) actual service types, NFA types, channel types, policy framework definitions, relationships and transformations between types; M1) Denotations for target concepts with specific rule for using existing policy framework names, NFA types, and channel types; M2) target concepts identified: service type, interface, identity, location ,policy, NFA type, communication channel type. The business network model repository: M0) actual business network model (BNM) specifications comprising of a set of linked community specifications, actual community specifications; M1) denotations for target concepts with specific rule for using service types for defining role requirements; M2) Target concepts identified: BNM, role, policy.

The definition of the semantics for interoperability safe collaboration spans the repositories: the service type repository provides extensions for the semantics of the ontology in both business network model repository and the service offer repository. In the service offer repository, the level M1 is not fixed but is extensible by publishing new service types in the service type repository. Similarly, new service types create possibilities for using new vocabulary in business network models.

The type discipline is defined through criteria for the relationships between information items within each repository. These relationships form the topmost metalevel for defining the target concepts for the interoperability knowledge; the rules to be defined by the publishers of the repository items will in turn define a dynamic type discipline [6].

For example, the criteria for acceptable business network model for a model repository are fairly similar to those of service types; they have to be named,

[1] MOF defines three levels for information: the actual entity level M0, the M1 level for notation rules, and the M2 level for meta-metamodel for the concepts in the storage.

can be grouped together due to asserted similarity, and must have a securely identified publisher to grant the quality of the published model. The quality of the model can be analysed and verified by future tools that give feedback on the correctness and recoverability properties, overhead cost, privacy-preservation policies and other aspects that affect the choice of the business network model. The roles of the models can only be defined by the vocabulary provided by the service types published in those repositories visible for the business network model repository provider.

Conclusion

The described knowledge base is part of the Pilarcos architecture, where the relevant repositories are pushed to the common network, to form infrastructure services. These services can be considered as societal services, or as services from trusted third parties, or even, as normal business of specific content providers. The repositories provide distribution channels to new best-practises business network models as they become defined by suitable domain consortia. Some performance measurements over a prototype implementation indicate that the overhead cost is reasonable, and the system scalable.

In comparison to ontology research, this structure is formed of a family of similarly structured dynamic ontologies, each rooted from a business network model, and used for automated interoperability enforcing.

Although QVT [1], the current state-of-the-art metainformation repository, provides much of the same functionality, our goal is more challenging because of the distribution to autonomous units and because of the dynamic extensibility of the M1 layers through the service type repository. The difference becomes motivated as we consider the three development waves and the differences between interoperability between modelling tools and interoperation of collaboration management facilities.

References

1. OMG. *Model-Driven Architecture - Vision, Standards And Emerging Technologies*, 2005.
2. Lea Kutvonen, Janne Metso, and Sini Ruohomaa. From trading to eCommunity population: Responding to social and contractual challenges. In *Proceedings of the 10th IEEE International EDOC Conference (EDOC 2006)*. Best paper award.
3. OMG. *Meta Object Facility (MOF) Specification*. April 2000.
4. M. P. Papazoglou and D. Georgakopoulos. Service oriented computing. *Communications of the ACM*, October 2003.
5. Toni Ruokolainen and Lea Kutvonen. Ontology for federated management of business networks.
6. Toni Ruokolainen and Lea Kutvonen. Service Typing in Collaborative Systems. In *I-ESA 2006*. Springer-Verlag, March 2006.
7. D.C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2):25–31.