Self-organization and SOA for networked enterprises

Elisabetta Di Nitto[1,2] and Gianluca Ripa[2]

[1] Politecnico di Milano, DEI, Via Golgi 40, 20133 Milano, Italy
[2] Cefriel, Via Fucini, 2, 20133 Milano, Italy
dinitto@elet.polimi.it, ripa@cefriel.it

1 Introduction

The needs for flexibility and globalisation force enterprises to both decentralise their activities and continuously (re)structure their network of relationships, regarding their productive processes, their "supply chain", and their design and innovation processes.

This demands for new innovative technological solutions, which must support unprecedented levels of dynamism. In particular, the overall architecture of the information system must evolve as requirements and context change. Typical cases of context changes occur when new services become available, existing services become less competitive, or even when they are discontinued. As soon as deviations from the desired or acceptable quality of service are discovered, or new opportunities are devised, the information system should be able to reconfigure itself in an autonomic manner. SOAs naturally fit in this context as they inherently support the concept of service, which represents one of the typical items organizations exchange, and also because SOAs are suitable to be equipped with self-adaptation mechanisms that enable them to evolve on the fly.

In this context, our research work is focusing on addressing the challenges presented below.

2 Recognizing network evolutions and actuating the needed actions

Networked enterprises usually are created on the basis of some need and trust relationships between stakeholders. Of course, networks that are created in the first place are not necessarily optimal. Also their performance can evolve over time. Thus, it is important to equip the underlying SOA infrastructure with monitoring mechanisms able not only to identify issues at the technical level, but also to spot potential places for improvement and optimization of the network performance. Consider, for example, a simple network where there is a *Service Provider* and a *User*. The User needs a service and the Provider provides the service to the User for a fee. This network works quite well, but it does not provide to the User any knowledge about the reliability of the Provider, the possible mismatches between the advertised quality for that service and the actual one, and the correct pricing. Thus, it may happen that the lack of knowledge for the User with respect to the existence of competing providers and to the reputation of the known ones may result in a reduced satisfaction or in paying more for having less. In this specific situation, the introduction of a third-party broker could improve the situation and allow the User to receive the required feedback and also to extend the number of its Providers [4]. While the decision on how to reconfigure the service network is intrinsically a task for humans, still the IT infrastructure can help in identifying possible reconfigurations as a result of a continuous monitoring and analysis activity. Monitoring should be built in a reconfigurable way so that the data to be gathered are decided dynamically, depending on the situation of the network. We imagine experts in the service network domain identifying the relevant data and properly instructing the monitoring infrastructure through a simple to use, still rich language. Part of the data analysis tasks could be automated as well through the definition of proper invariants and predicates that should be checked on the basis of the data acquired by monitoring.

3 Enabling self-evolution of the underlying SOA infrastructure

Dynamic evolution of the networked enterprises clearly results in the need for evolving the SOA infrastructure as well. In the literature, various approaches have been identified, but most of them are still focusing on the specific mechanisms to use more than on the definition of a rigorous design approach that leads to a SOA system able to evolve at runtime in a controlled way. In [1–3] we propose some preliminary approaches that are currently being extended and validated. Figure 1 shows the life-cycle we have identified for adaptable SOA systems. It highlights the fact that adaptation happens at runtime and goes through various phases (see the left hand side of the figure) where activities devoted to the identification of adaptation needs and strategies are followed by the one devoted to the execution of the adaptation, which makes the system returning back in its operation phase in an evolved configuration. Even though we have contributed to the identification of these phases and are working on clarifying their purposes, still specific techniques and approaches are needed to allow designers to *design the system for adaptation and evolution.*

4 On the fly revision of SLAs

In [5] we have presented a framework that supports design-time and runtime negotiation of SLAs in a SOA context where the negotiation participants are essentially service providers and service consumers. The framework also offers the possibility to replace (part of) these participants with software agents that act on behalf of them and to design new negotiation processes, possibly including multi-parties, which are then interpreted by the runtime infrastructure. Differently from many works in literature that support specific negotiation processes,

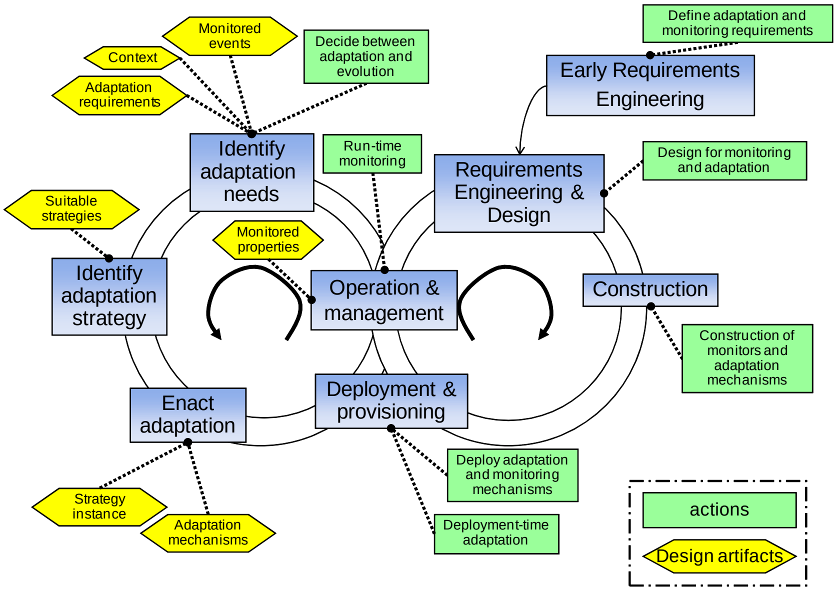
Monitored events
Context
Adaptation requirements
Decide between adaptation and evolution
Define adaptation and monitoring requirements
Early Requirements Engineering
Identify adaptation needs
Run-time monitoring
Requirements Engineering & Design
Design for monitoring and adaptation
Suitable strategies
Monitored properties
Identify adaptation strategy
Operation & management
Construction
Construction of monitors and adaptation mechanisms
Enact adaptation
Deployment & provisioning
Deploy adaptation and monitoring mechanisms
Deployment-time adaptation
Strategy instance
Adaptation mechanisms
actions
Design artifacts

Fig. 1. The S-Cube life-cycle of adaptable SOA systems.

our framework can be tailored depending on the multiplicity, workflow, protocol, and decision model that fit a specific application domain. The evolution in the networked enterprise or in the underlying IT infrastructure may determine the need for revising the SLAs established between the involved stakeholders. Consequently, approaches to identify needs for revision and to renegotiate SLAs are required.

5 Conclusions

In this position paper we shortly describe the research challenges we are currently focusing on. These concern, in particular, the need for identifying and applying potential changes in a network of enterprises, the need for supporting self-evolution as a result of changes in the network, the need for revising during the operation process the already agreed SLA. We envisage that new innovative technological solutions in these areas are necessary for addressing the service requirements of networked enterprises in the Future Internet.

Acknowledgements

Our research is partially funded by the European Community, Seventh Framework Programme FP7/2007-2013 under grant agreements 215483 (S-Cube), 216556 (SLA@SOI) and 215219 (SOA4All).

References

1. A. Bucchiarone, C. Cappiello, E. di Nitto, R. Kazhamiakin, V. Mazza, and M. Pistore. Design for adaptation of Service-Based applications: Main issues and requirements. In *International Workshop on Engineering Service-Oriented Applications*, 2009.
2. A. Bucchiarone, R. Kazhamiakin, C. Cappiello, E. Di Nitto, and V. Mazza. A context-driven adaptation process for service-based applications. In *PESOS 2010 – 2nd International Workshop on Principles of Engineering Service-Oriented Systems*, Cape Town, South Africa, 2010.
3. L. Cavallaro, E. Di Nitto, C. Furia, and M. Pradella. A Tile-based Approach for Self-assembling Service Compositions. In *In Proceedings of International Conference on Engineering of Complex Computer Systems*, 2010.
4. D. Dubois, C. Nikolaou, and M. Voskakis. A model transformation for increasing value in service networks through intangible value exchanges. In *International Conference on Service Science, ICSS2010*, 2010.
5. E. D. Nitto, M. D. Penta, A. Gambi, G. Ripa, and M. L. Villani. Negotiation of service level agreements: An architecture and a search-based approach. In B. J. Krämer, K.-J. Lin, and P. Narasimhan, editors, *ICSOC*, volume 4749 of *Lecture Notes in Computer Science*, pages 295–306. Springer, 2007.