Service-Oriented Design: The jABC Approach

Tiziana Margaria[1], Bernhard Steffen[2], and Manfred Reitenspieß[3]

[1]Service Engineering for Distributed Systems, Universität Göttingen (Germany), margaria@cs.uni-goettingen.de

[2]Chair of Programming Systems, Universität Dortmund (Germany), steffen@cs.uni-dortmund.de

[3]Director Business Development, RTP 4 Continuous Services, Fujitsu Siemens Computers, Munich (Germany), Manfred.Reitenspiess@fujitsu-siemens.com

Service-Oriented Design has driven the development of telecommunication infrastructure and applications, in particular the so-called Intelligent Network (IN) Services, since the early 90s: IN services are customized telephone services, like e.g., Free-Phone, where the receiver of the call can be billed if some conditions are met, Virtual Private Network, enabling groups of customers to define their own private net within the public net, or Credit Card Calling, where a number of services can be billed directly on a credit card account. The realization of new IN services was quite complex, error prone, and extremely costly until

- a service-oriented, feature-based architecture,
- a corresponding standardization of basic services and applications in real standards, and
- adequate programming environments

came up: they set the market, enabled the flexibilization of services, and dramatically reduced the time to market. Today the current trend moves toward *triple-play* services, which blend voice, video, and data on broadband wireline and wireless services. It builds on this successful experience while reaching for new technological and operational challenges.

Reviewing our 10 years of experience in service engineering for telecommunication systems from the point of view of Service-Oriented Design then and now, we observe that much is common to the two communities. We aim in our current research at establishing a link to the notions used by the service-oriented programming (SO) community.

The central observation is that both communities pursue the same goals, a coarse granular approach to programming, where whole programs serve as elementary building blocks. However, they have quite a different view on what a service is and how it is organized. In the terminology of the SO-community, a service is a "nugget" of functionality (essentially a building block) that is directly executable and can be published for use in complex applications. In the telecommunication world, such elementary components are called Service-Independent Building blocks (SIBs), and the notion of service is typically used for the resulting (overall) application. In addition, in the telecommunication world

Dagstuhl Seminar Proceedings 05462
Service Oriented Computing (SOC)
http://drops.dagstuhl.de/opus/volltexte/2006/521

the notion of feature is used to denote substructures of services (applications) that impose additional functionality (like e.g. call forwarding, or blacklisting) on the generic basic telecommunication functionality. In the IN-architecture, the basic functionality was POTS (plain old telephony service), and features were typically only executable in the context of POTS.

It was always our point of view that some of these distinctions would disappear as soon as one lives in a fully hierarchical context, where services may themselves be regarded as elementary building blocks at a higher level of abstraction. In this scenario, which is supported by jABC, our service definition environment [4][1] , the notion of service captures the corresponding notions of both the SO- and the telecommunication communities. Moreover, the notion of SIB simply characterizes services which cannot be refined, and the notion of feature characterizes subservices that cannot be executed on their own. The remainder of this paper is written from this unifying perspective and it focusses on the impact of formal methods to improve the service development process. This concerns in particular the idea of incremental formalization [10, 8], which allows users to already exploit very partial knowledge about the service and its environment for verification. In turn, this enables a division of labour, which in particular enables the application expert to directly cooperate in the service definition process.

Already the traditional concept of services in an Intelligent Networks Architecture, stressed and expanded in the current telecommunication perspective, works with a feature-based paradigm very similar to the current concept of services. The INXpress Service Development Environment (SDE), the Siemens solution to Advanced Intelligent Networks that came out of our cooperation in 1995-1996, is a commercial product that shaped the state-of-the-art of IN-service definition in the late '90s [11–13]. Presented at various international fairs (e.g. CeBIT'97), it was installed at a number of early-adopter customers (e.g. Deutsche Telekom, South Africa's Vodacom, Finnland's RadioLinja), while a number of further key contracts followed, where our SDE was a key factor for the decision of changing to Siemens technology. The success of the IN services since then has clearly demonstrated the validity and adequacy of the service-oriented way of thinking.

The same approach to service definition, composition, and verification has been meanwhile successfully used in other application domains. With the ABC (Application Building Center) and the jABC (Java ABC) we have meanwhile built internet based distributed decision support systems [2], an integrated test environment for regression test of complex CTI systems [7], a management infrastructure for remote intelligent configuration of pervasive systems [1], as well as many other industrial applications in e-business, supply chain management, and production control systems. In the area of internet-based service orchestration and coordination we have developed since 1997 the Electronic Tool Integration Platform, ETI [9, 14] and its Web services based successor, jETI [4, 3]. jETI provides

[1] The METAFrame and the ABC are the predecessors of the jABC environment

– lightweight remote service integration by registration,
– distributed service libraries,
– a graphical coordination environment to build (hierarchical) service orchestrations,
– mechanisms to simulate and formally verify the global services (choreography)
– a distributed service execution environment.

Currently its application focus is on offering as services a number of tools for program analysis, verification and validation.

We are convinced that combined approaches, that blend the flexibility of the current SO-scenario (see e.g. [6]) with the rigour and semantic standardization culture of the telecommunication community will dramatically increase the productivity of the development of a large class of software systems [5]. Incremental formalization and automatic verification techniques may be again the key to achieving confidence and reliability for services that interact and interoperate on a large distributed scale.

References

1. M. Bajohr, T. Margaria: *MaTRICS: A Management Tool for Remote Intelligent Configuration of (Pervasive) Systems*, Proc. ICPS 2005, IEEE Int. Confer-ence on Pervasive Services, 11-14.July 2005, Santorini, Greece, pp. 457-460, IEEE Computer Society Press, 2005.
2. Tiziana Margaria: *Components, Features, and Agents in the ABC.* In *Objects, Agents, and Features*, Revised and Invited Papers from the International Seminar on Objects, Agents, and Features, Dagstuhl Castle, Germany, February 2003, LNCS 2975, pp. 154-174, Springer Verlag, 2003
3. T. Margaria: Web Services-Based Tool-Integration in the ETI Platform, SoSyM, Int. Journal on Software and System Modelling, Springer Verlag, (available in Online First, DOI: 10.1007/s10270-004-0072-z).
4. T. Margaria, R. Nagel, B. Steffen: Remote Integration and Coordination of Verification Tools in jETI Proc. ECBS 2005, 12th IEEE Int. Conf. on the Engineering of Computer Based Systems, April 2005, Greenbelt (USA), IEEE Computer Soc. Press, pp. 431-436.
5. T. Margaria, B. Steffen, M. Reitenpieß: *Service-Oriented Design: The Roots*, Proc. ICSOC 2005: 3rd ACM SIG-SOFT/SIGWEB Intern. Conf. on Service-Oriented Computing, Amsterdam (NL), 12-15 Dec. 2005, to appear in LNCS, Springer Verlag.
6. METEOR-S: see the project site at `lsdis.cs.uga.edu/projects/meteor-s/`
7. O. Niese, B. Steffen, T. Margaria, A. Hagerer, G. Brune, H.-D. Ide: *Library-Based Design and Con-sistency Checking of System-Level Industrial Test Cases*, Proc. FASE 2001, Int. Conf. on Fundamental Approaches to Software Engineering, Genoa (I), April 2001, LNCS 2029, pp. 233-248, Springer-Verlag.
8. B. Steffen, T. Margaria: *METAFrame in Practice: Intelligent Network Service Design*, In *Correct System Design – Issues, Methods and Perspectives*, LNCS 1710, Springer Verlag, 1999, pp. 390-415.

9. B. Steffen, T. Margaria, V. Braun: *The Electronic Tool Integration platform: concepts and design*, [14], pp. 9-30.
10. B. Steffen, T. Margaria, A. Claßen, V. Braun: "*Incremental Formalization: A Key to Industrial Success* ", In "SOFTWARE: Concepts and Tools", Vol. 17, No 2, pp. 78-91, Springer Verlag, July 1996.
11. B. Steffen, T. Margaria, A. Claßen, V. Braun, M. Reitenspieß: "*An Environment for the Creation of Intelligent Network Services*", invited contribution to the book "Intelligent Networks: IN/AIN Technologies, Operations, Services, and Applications – A Comprehensive Report" Int. Engineering Consortium, Chicago IL, 1996, pp. 287-300 – also invited to the *Annual Review of Communications*, IEC, 1996, pp. 919-935.
12. B. Steffen, T. Margaria, A. Claßen, V. Braun, M. Reitenspieß, H. Wendler: *Service Creation: Formal Verification and Abstract Views*, Proc. 4th Int. Conf. on Intelligent Networks (ICIN'96), Nov. 1996, Bordeaux (F).
13. B. Steffen, T. Margaria, V. Braun, N. Kalt: *Hierarchical Service Definition*, Annual Review of Communic., Int. Engineering Consortium, Chicago, 1997, pp.847-856.
14. *Special section on the Electronic Tool Integration Platform*, Int. Journal on *Software Tools for Technology Transfer*, Vol. 1, Springer Verlag, November 1997