Foreword

Do you also always carry your mobile device? Do you use it everywhere, i.e. in the street, while travelling and at your workplace? Do you want to explicitly handle specific environment characteristics which are all different depending on where you are: For example, there usually is better connectivity at your workplace or at home than during travel, security aspects change accordingly, sometime you access your own, well-defined service environment, sometime you want to detect new services in unknown territory etc. But, of course, regardless of where and for what you use it, your device shall always function “right” in order to take advantage of whatever the specific environment has to offer (as, e.g., connectivity, compute power, service diversity, security, fun, etc.). What that basically means is, that your mobile device has to always “adapt” flexibly and appropriately to the specific characteristics of such respective environments.

In a bit more abstract terms: Most of today’s software applications are distributed in one way or the other, and many of them are executed in heterogeneous and often dynamically changing environments – with examples ranging from relatively small and static multi-processor systems (such as in car electronics) up to open world-wide service co-operations in e-business or (just to mention another set of applications) highly dynamic multi-player games, for example. And for all such environments we need appropriate system software support.

Such a distributed system infrastructure (i.e. middleware) has to be generic (i.e. standardized) in order to provide the basis for state-of- the-art software development for such applications. In addition, current distributed systems rely heavily on the paradigm of a Service-oriented Architecture (SOA) – with Web Services (WS) as its most prominent realization. Therefore, in the best of all worlds of open, service-oriented distributed systems, the “right” set of services is available when- and wherever needed and services can, in addition, be combined freely and flexibly – regardless of where, how and by whom they are realized, managed, and provided. So, new services can always be composed out of existing ones with as little effort and security problems as possible.

Reality, however, is still different and further challenges arise – just to name a few: appropriate services for such compositions may have to be “discovered” first and their respective interfaces as well as their behavior have to be uniquely identified; then, services which do not “fit” exactly, may have to be “adapted” and then (dynamically) combined in order to realize co-operative applications in various ways – dependent on, e.g., the underlying application pattern (e.g. by choreography or by orchestration); appropriate “agreements” on service-level characteristics have to be reached between co-operation partners; and, especially in pervasive and dynamic environments, sets of available services that change dynamically together with any context changes have to be taken care of.
So, at least service interfaces (esp. interface definitions) as well as co-operation patterns have to be agreed upon in order to make such co-operations possible in heterogeneous distributed environments. Some of the related standardization issues are addressed by corresponding WS*-standards and middleware platforms – although with standards mostly limited to a syntactical level. Some further ideas, e.g. on semantic extension, exist – but many of them still have to prove their relevance in practice.

The ultimate goal of software to support such applications is to be able to adapt to many heterogeneity and/or environment changes – ideally automatically and flexibly in ways as required by heterogeneous distributed service-oriented applications in dynamically changing environments.

Motivated by these challenges and related problems, this book addresses many issues of improving software development for such applications in a platform independent and loosely coupled manner – in order to realize modular and adaptive distributed service-oriented systems with great flexibility and easy to fulfill maintenance requirements. In this sense, advanced distributed applications should be self-contained, self-descriptive and self-adaptive as well as able to locate, access, and invoke related services and compose them freely, flexibly and context-dependently from any point of a distributed application environment. In addition, service compositions as well as service adaptations should be implemented in ways which preserve the loosely coupled nature of Web Services and which allow for implementation’s integration in the whole service lifecycle.

Rather attractive and practice-relevant partial solutions which address many of these challenges exist already and are referenced in this book. Accordingly, state-or-the-art research results are described and discussed – ranging from “context adaptation” (i.e. context-aware services) over “device adaptation” (i.e. service adaptation to specific device characteristics) and “user preference adaptation” (i.e. service adaptation to the specific needs of mobile users in the web), software-as-a-service etc. up to “adaptation for composition” (i.e. ways to make – independently developed – services ready for dynamic composition) and support for service mediation.

In summary, this book provides a compound collection of the most representative approaches in current research which tackle the different faces of Web Service adaptation for (e.g.) software engineers, from senior and young researchers at the academia to research and development groups in industry.

Winfried Lamersdorf
University of Hamburg, Germany March 2012

Winfried Lamersdorf is a full Professor in the Informatics Department of Hamburg University since 1991, former department head as well as deputy head for research, and head of the “Distributed and Information Systems” research unit with specific responsibilities in the area of “Distributed Systems,” and also a (founding) board member of the Hamburg Informatics Technology Transfer Institute HITEC e.V. After his PhD in Computer Science in 1985, he spent a year at the University of Maryland, USA, in collaboration with the National Institute of Standards and Technology (NIST). From 1983 to 1990 he was scientific staff member at the IBM Scientific and European Networking Centre(s) in Heidelberg working in the area of open distributed applications, with some 10 years of additional experiences in international standardization in that area. He has lead and conducted several research projects – e.g. several with the German national research fund (DFG), the European Commission (in FP 5, 6, and 7), the German ministry for education and research (BMBF), as well as with a number of industry partners (as, e.g., IBM, HP etc.). He has (co-) authored and edited numerous scientific papers as well as several books. In 2004 and 2008 he was scientific workshop co-chair and in 2006 program co-chair of the ICSOC conferences in Amsterdam (NL), Chicago (USA), and Sydney (AUS). Since 2008 he is also participating in the EU funded international Network of Excellence project in the area of Software Systems and Services (S-Cube).