Preface

“A verbal contract isn’t worth the paper it is written on.” (Samuel Goldwyn)

Last year, at a conference in Paris we decided to go out for dinner. Our little group consisted of four different people with various expectation and requirements of a good dinner. Everybody has their own experience in finding the right place for just two persons – but how difficult does it become to satisfy 4 people’s requirements! Thus, we started to collect different wishes: vegetarian food must be served, credit cards should be accepted, the restaurant must be in walking distance, and last but not least, it cannot be too expensive. Finally, we just jumped into one restaurant, which was close by. The menu outside looked quite promising in terms of food offerings (vegetarian and non-vegetarian) and stickers at the door pointed out that credit cards are accepted.

While waiting for the food – quite a long wait already hinting at more surprises in the making on that evening – we started to discuss our selection process. You can easily image that folks with computer science backgrounds try to analyse it using their domain knowledge. Thus, one of us started directly to model restaurant offerings as non-functional properties of services while the second looked into service selection based on these offerings. The third member of our group was more concerned with validating that all offerings are satisfying requirements and finally, the last member wanted to see how feedback can be used to allow for a more agile offering. While discussing all these it turned out that there was no vegetarian food available on that evening, and even settling for other dishes we found out after the meal that credit cards were not accepted after all. Having settled the bill we retired to a nearby bistro to exchange stories about broken service promises …

Why are we telling this story and how is it related to the book? A restaurant is an everyday service, services have characteristics that go beyond their pure functionality (here delivering the needed calories for survival), people have different expectations and agreements are needed and often promised characteristics of a service are not met.

So, we can say we have learned a lot during the diner experience:

1. Service modelling, non-functional properties, quality of service, and service level agreements can be applied easily in daily life and are not limited to Service-oriented Architectures.
2. There is common understanding of the existence of non-functional properties of services, but there is no concrete agreement on how to express them. This problem starts with a lack of agreed syntax extending into understanding the semantics of specific properties.

3. Non-functional properties have a very significant impact on how any service is chosen and how satisfied users are – this impact extends into Service-oriented Computing.

WHY SHOULD THERE BE HANDBOOK IN THIS AREA?

Notions of services and Service-oriented Computing have emerged and matured over the last decade, bringing with them a plethora of available services that are selected by users and developers and composed into larger applications. The general view with services is that they expose business level functionality, and hence, are at an abstraction level previously not typically addressed by IT. However, with a growing number of services available and selected by business organizations to fulfill much of their business operation concerns beyond pure functionality come to the fore.

In general, one can consider the functionality of a service as an essential requirement for selecting it, but if there is a choice of such services other concerns, called non-functional are evaluated in decision-making. As an example from daily life – besides perfect restaurant selection - we can consider car insurance: the service provided by different insurance companies is comparable: they provide insurance that covers theft and third party damage (the functional aspect). However when buying insurance we consider issues such as the reliability of the insurer, the costs of insurance, extra benefits provided, the process of dealing with the insurer, and others: these would be the non-functional properties (NFPs).

The area of non-functional properties for services is quite wide, covering aspects from describing and classifying NFPs, obtaining values for them (either as part of the design or automatically during execution), and using NFPs for discriminating between services. In addition NFPs form an essential component in quality of service (QoS) guarantees – usually referred to as service level agreements (SLAs). Here questions arise regarding describing SLAs, embedding NFPs in them, providing assurance that NFPs are indeed adhered to. As orthogonal dimensions both NFPs and SLAs must be (1) managed and (2) can be general or domain specific. These two dimensions place additional requirements on the aspects described before and have led to much work of their own.

The area of non-functional properties for services is quite diverse and has been evolving very fast, with industrial contributions and standard bodies as well as academic work providing (partial) solutions to certain aspects. Various research domains provided approaches to describe, map, and use NFPs, such as the Semantic Web researchers as well as service-oriented academics. Therefore, there is a huge variety of different approaches and methods to address NFPs. Work in this area has no unique forum in which it is presented; in contrast, publications are distributed in a large number of conferences and workshops (e.g. ECOWS, ICWS, ICSOC, NFPSLA-SOC, NFPSLAM, SCC, ServiceWave).

This handbook brings together results from the different communities and provides a clear overview of the work done and we would hope also a clear direction for future work. It should also bring the diverse communities closer together by highlighting synergies between and complementation of approaches and lead to more focused efforts of making progress rather than duplicating work in one community because of unawareness of that done (or ongoing) in another.
WHO SHOULD READ THIS BOOK?

As we have seen service, their non-functional properties and aspects related to these affect all services – so one could argue that everyone should read this book! However, when selecting chapters for the book and editing the final version we had a narrower focus in mind: we want the book to be a reference and starting point for new researchers and industrial practitioners in the area of non-functional properties and SLAs for services as well as an up-to-date reference for those already active in this field. In particular the mix of industrial and academic perspectives presented in the book which are, as expected, not necessarily married up, in that many academic solutions do not address the industrial problems is invaluable to both communities. With service computing making its way into academic curricula, a need arises for students to be exposed to more advanced topics such as the ones presented in this book, so the book or individual chapters should make suitable advanced reading for final year Undergraduate or Masters students.

SHALL THE BOOK BE READ FRONT TO BACK? OR WHERE DO I FIND CHAPTERS MATCHING MY INTEREST?

There might be books, which you can read from the first sentence up to the end. We believe such books mainly to be non-academic books. Academic books, or in this case a researchers’ handbook, are following a different path: the reader is invited to jump from one chapter to another driven by his interest and needs. In this way the book features as a reference text. If you are new to the area you might want to start reading at the beginning and work your way through it: the structure of the book is such that it is open to end-to-end reading.

In any case, getting an overview of the basic roadmap and organizational philosophy of this book will prove useful.

Section 1 (Perspectives on Non-Functional Properties) of the book covers questions around non-functional properties in general, such as how they can be modelled and how to reason about it. The reasoning can be in terms of providing guarantees on their values but also on the behaviour of the services that they characterize. This section sets the scene for many of the uses of non-functional properties discussed in the latter sections, where authors use such fundamental considerations to achieve more specific aims. This section starts with an industrial perspective and highlights how to measure the various set of non-functional properties. It provides also some insights into guarantying NFPs and how to express in a mathematically precise way using a semantic framework. Furthermore, this section investigates how NFPs influences service behaviour in a way that can be measured. Looking on ontologies for NFPs presents a more semantic viewpoint. The final chapter in this section looks at measuring performance data through hardware solutions – providing data to populate non-functional property models. A more detailed description of the chapters follows.

Chapter 1 provides a good overview from an industrial perspective as to which non-functional properties could be of interest. It highlights how to measure the diverse range of non-functional properties – that is, it proposes kinds of metrics that can be used. It also discusses why good values for some properties are harder to achieve than for others. Most crucially, the chapter provides a good insight into the variety (in terms of the properties themselves, but also the values that they can take) of non-
functional properties that can be of interest and highlights why non-functional properties are of interest to the industrial practitioner.

Chapter 2 considers the important question of how guarantees about non-functional properties can be given. The authors lift the concepts to a higher level of abstraction, based on what can be observed of a property, to then reason in a sound way about the properties in the context of not only services but also the wider system that they operate in. A semantic framework is defined to make the issues mathematically precise. A number of properties are examined in detail to show how the semantics can be defined for them, which is then exemplified with examples for each.

Chapter 3 talks about how non-functional properties influence service behaviour in a way that can be measured. It is useful for service designers to understand at an early stage what impact certain design decisions have on the behaviour of a service – often the speed with which it can perform its function in the wider architecture where it is used. This chapter explores, based on a specific modelling notation for services, how stochastic analysis of time related properties can be conducted in an analysis framework.

Chapter 4 is finally diving into ontologies. A large area of providing models that place concepts in relation is governed by ontologies. Ontologies are heavily used in the semantic web community, and by extension in the semantic web services community – usually centred on languages such as OWL. This chapter provides concrete ontological models for many non-functional properties. In some way it picks up on Chapter 1, which introduced many non-functional properties, but it breaks these into well-structured sub-elements and makes the links between these explicit. As is typical for ontological models, much of the content is conveyed through diagrams.

Chapter 5 starts with the observation that the growth in complexity of data processing needs and infrastructures requires performance quantification to determine capacity planning, service level agreements, and QoS determination. The proposed solution uses on chip registers to monitor performance in a largely non-intrusive way. The monitored data provides run-time values for non-functional aspects.

Section 2 (Service Selection) of the book is looking into how to use NFPs to support service selections which is a key use case for NFPs. In the domain of service and NFPs, there are quite a lot of different approaches around service selections in various contexts – just remember the restaurant in Paris. With the uptake of service computing, the amount of available services has been rapidly increasing, leading to a situation where a user can find many services for a specific requirement. This choice comes with a need to identify the best service for a user’s need based on non-functional properties such as cost or performance. The contributions in Section 2 deal with different approaches for selecting services. Firstly, this part starts with transactional properties for service compositions. Secondly, some work introduces a quality of service model that can be used during the selection phase by a service broker. Other chapters present an extensible model for NFPs as basis for a generic selection model and methods for selecting context-ware services for ubiquitous and pervasive systems. In Chapter 10 the topic is quality of service modelling. Here, the goal is to identify models, which share sufficient commonality to allow a mapping through meta-models. Chapter 11 is talking about quality prediction to determine QoS level that can be achieved and guaranteed. The last chapter of this part, Chapter 12, is kind of summarizing this part by looking at service selection from a user’s perspective. This chapter proposes methods to obtain user requirements using the mid-level splitting technique and to assign weights using hypothetical equivalents technique.

Chapter 6 considers transactional properties. Transactional properties of Web services are an often-ignored non-functional property. However, many composition approaches consider how transactional properties of services compose. Considering this from a different angle; one can ask the question of how
a specific overall transactional property can be achieved by selecting the right services. This chapter presents an overview and classification of existing approaches in the area of service composition and selection based on transactional properties.

Chapter 7 introduces a quality of service model that can be used during the selection phase by a service broker. In particular it considers that providers express properties at low levels of abstractions, while users use higher levels of abstraction to express their requirements. The contribution is a mapping between the two levels of abstraction that is followed through with a service selection method.

In Chapter 8 the authors present an extensible model for non-functional properties, which provides the basis for a generic selection method. The focus is on the model and not on specific properties – these can be added as they emerge without the selection method changing. An additional aspect considered is the “composition context” which allows capturing information of the process context in which the service is executed. This context provides further non-functional information to select the most appropriate service.

In Chapter 9 the authors consider Context-aware services are emerging in the context of ubiquitous and pervasive systems. Descriptions of these services are enriched with context information linked to non-functional properties. The nature of context information is such that it is highly dynamic, unreliable, and not always available, thus posing extra challenges for service selection. This chapter presents a graph matching based algorithm to allow service selection in the face of uncertain information.

Chapter 10 provides an approach to model QoS for services, modelled using different models by different entities, as there is currently no established standard. The models share sufficient commonality to allow for mapping through meta-models. However, for matchmaking one also needs to compare actual values, and these are obtained through various metrics that are much harder to consolidate. This chapter presents algorithms to align metrics and to address numerous issues arising through the underlying mismatches.

In Chapter 11 obtaining and measuring QoS levels is a non-trivial task and can be quite intrusive on the system. Quality prediction allows providers to determine QoS levels that they can achieve and guarantee. This chapter considers modelling systems and using the Palladio Component Model for analysis and performance prediction. These techniques can be applied before the system is deployed and also apply in Event-based SOA.

Chapter 12 talks about capturing user requirements. Choosing services is non-trivial, as the plethora of research contributions in this field shows. One difficult issue is to capture the expectations of a user -- users tend not to be technical staff and hence cannot judge the effect of some decisions, which could lead to no service being available. On the other hand they understand the need to compromise and settle for “inferior” services. This chapter proposes methods to obtain user requirements using the mid-level splitting technique and to assign weights using the hypothetical equivalents technique.

Service contracts are the topic of Section 3 (Service contracts). A progressive use of services in business environments, but also a growing use of services for crucial personal affairs lead to and increased dependence on services. Therefore, service contracts between consumer and provider are becoming key issues for service usage in various environments. Thus, this dependence requires, like any business transaction in the real world, a good agreement between the expectations of all parties involved in a transaction. Service contracts attempt to provide this agreement and they are fundamentally dependent on non-functional properties: it is much easier to agree on the functionality than on the quality with which a service should be delivered or can be expected to be delivered. The first chapter provides already a good overview of approaches to express service contracts. The other chapter pick up on more specific
aspects to measure service behaviour as input for service contracts by introducing concepts such as service popularity or service goals.

Chapter 13 considers the plethora of work to describe service contracts – most of which covers similar aspects but presents these in different notations. This presents a syntactic and semantic problem when negotiating contracts. The authors provide an overview of the existing languages and present a solution to reconcile contracts presented in different languages. Their work covers the whole contract lifespan from identification, through negotiation and contract creation to monitoring contracts.

How often a service is invoked is a non-functional property reflecting interest in or need for a specific service. Chapter 14 presents a model to describe service popularity and a measure called ‘Gross interest’ to capture service popularity. This specific NFP can be used like any other one, but additionally proves useful when deciding on which services to replicate as shown by the authors’ example.

SoA has made a big impact on how software solutions are produced at a technical level, and always had the ambition to lift software to the business level. Chapter 15 considers the intentional Service-Oriented Architecture (iSOA), aimed at business people by moving from functional to intention driven service descriptions. Clearly such a move needs to consider Quality of Service aspects and the chapter introduces this aspect using quality goals and their quantitative and qualitative evaluation.

The authors of Chapter 16 are concerned with evaluating the performance of Web Services. A platform for monitoring service providers and identifying the most appropriate service is presented. The platform can be used as an analytical vehicle for testing Web Services and understanding their behaviour. The measured data can support creation of service contracts since the measurements focus on performance and a characterization of what to evaluate and how.

Chapter 17 considers the topic of Business Continuity Management, which is concerned with ensuring that potential threats are identified and resolution strategies are in place to handle IT failures. Specifically the chapter discusses major challenges in this area and presents approaches to map business requirements to service level terms and metrics in the context of SoA.

In Section 4 (SLA Governance), we are moving the SLA topic further towards SLA governance. While Section 3 is talking about service contracts and SLAs in general, Section 4 is more about the interactions with SLA and the necessity of SLAs in terms of contracts and impact on the business. As these interactions are very critical, it is necessary to arrive at good contracts (that is one might need to negotiate on details of an agreement) as well as ensuring that contracts are adhered to. The area of SLA governance is concerned with these aspects. The chapters cover topics focused on negotiation of SLAs including provisioning of resources to fulfil SLAs, which is even more relevant down the entire IT stack. However, the final chapter is covering business aspects and compliance with a focus on the end-user perspective by visualising data. SLA governance could easily fill a book by itself, so this part touches on some key aspects of the area with a high correlation to NFPs.

Chapter 18 surveys semantic modelling of security and SLAs and the state of the art in SLA-based governance. Based on this survey and observation of the essential aspects a semantic model of resource dependability is defined. This model allows describing the service commitments made to customers and resource capacity required from suppliers as non-functional properties. An approach to SLA-based system governance allowing for elastic provisioning of resources makes use of the model.

Service level agreements are usually defined through a negotiation process. Chapter 19 introduces various models, specifications, and realizations of service level agreement negotiations. This existing work shares a common core which is identified and exploited to define a coherent framework applicable to different negotiation models and protocols covering all functions of the negotiation phase. Challenges
are discussed, predominately the need to increase acceptance of service level agreement negotiation and enhancing interoperability.

Service-oriented architecture allows for systems to be built from loosely-coupled components with SLAs describing how services are used in conjunction with each other. These SLAs typically describe what a customer can expect from the provider, but there is no agreed understanding of how these high-level contracts can be mapped into operational artefacts in different layers of the IT system. Chapter 20 presents a framework to manage SLAs across the IT stack in an integrated framework.

The authors of Chapter 21 present a vision of a future where services form part of an open eco system, where services are provided by independent parties, services evolve, and collaborations are dynamically established. For this vision to work, it is essential that features of services and their dependencies can be governed effectively. The paper presents a framework and the required underlying conceptual models to enable such governance in open service eco systems.

Chapter 22 presents a compliance dashboard. The need for this is motivated by the fact that assessing whether business practices conform to laws and agreements is difficult, and hence, there is a need for appropriate tools. Developing such tools, such as addressing the right level of abstraction in descriptions, visualizing analysis results for different stakeholders, and more fundamentally, managing and visualizing large amounts of data in a time effective manner is important.

Section 5 (Future Directions) provides a perspective authors view on future directions, how to use NFPs in large data sets or how to handle real-time requirements for NFPs. This section tries to come up with some predictions on what will be required in a wider field of view. The last section contains only one chapter, but if you wish to only read one chapter, make it this one. Each individual chapter in the previous sections provides the respective authors view on future directions of their area of work. The motivation for this part is a bit more ambitious in that it tries to look into a future that is further afield and that tries to make predictions on what will be required in a wider field of view.

Chapter 23 is concerned with Web Services, the Programmable Web, and Web 2.0, which have appeared, seen much support from IT companies, and have seen much hype and also good uptake within business boundaries; however, there have been problems hindering a universal uptake. This chapter presents a concise summary of the current state of affairs and then explores persisting problems with data management and data integration as well as capturing and processing non-functional properties (especially in a time efficient manner) which have to be solved across technology boundaries, including traditional SOAP based services as well as Web APIs and RESTful services.

**WHAT HAS BEEN ACHIEVED?**

Whether you are new to the field of non-functional properties for services or a seasoned researcher or industrial practitioner, we believe that this book will be a useful resource for you. The book provides structure to existing work, sets clear future directions for the field, and brings together results from different communities.

The book presents a unique mixture of results achieved by both industry and academia, and thus bridges (or at least identifies the gaps) between these results and covers the whole span from defining NFPs through to their use and management in systems.

We hope that you will enjoy the book and that you will pick it up frequently to consider new dimensions or aspects. Most of all we hope that this book will inspire you to contribute your own solutions
and ideas into an exciting field of research and industrial practice, and that together with the existing community, you will ultimately help people to be able to enjoy the best services without disappointment, or considering that dinner in Paris, get the food they like without any disappointments and be able to enjoy every aspect of an evening out!

Stephan Reiff-Marganiec
University of Leicester, UK

Marcel Tilly
European Microsoft Innovation Center, Germany